Cops and Robbers on Chess Graphs and Hypercubes

This is a project proposal for the 2025 REU at Northeastern University. This project will be mentored by PhD students Evan Griggs and Arturo Ortiz San Miguel, who will also consult with Professor Gabor Lippner.

Overview

"Cops and robbers" is a two-player pursuit-evasion game played on graphs. Based on the children's game of the same name, the game has been the center of a highly active research area for over 50 years, since the late 1970s. In it, one player (the robber) evades capture by the other player (some number of cops) by taking turns moving along the edges of a given graph. There are many variants of the game, each of them obtained by tuning parameters like movement speed, capture distance, the number of mobile cops per turn, invisibility of the robber, hostility of the robber, *et cetera*. In light of the many variations of the game that exist, the literature on the topic is broad and diverse. We hope this REU can contribute to that literature with a paper publishing the results of the applicants.

Rules of the game

The game is played on a *graph* $G = (V, E)$, which is a collection of *vertices* V and *edges* $E \subset V \times V$ between them. Two vertices v, w are said to be *adjacent* if (v, w) is an edge. Most variants stipulate that G is finite and connected, but this is not required in general as is the case for the chess graphs defined later in this proposal. There are two teams, the *cops* and the *robber*. There are $k \geq 1$ cops but only one robber. The rules of the standard variant of the game are as follows:

(1) The k-many cops are placed on distinct starting vertices. The robber is then placed on a starting vertex.

(2) Players alternate turns with the cops moving first.

(3) Movement is confined along an edge of G, and any number of people may choose not to move during their turn.

(4) If there is a cop adjacent to the robber at the start of the cops' turn, then the cops win. Otherwise, if the robber is able to evade capture indefinitely, then the robber wins.

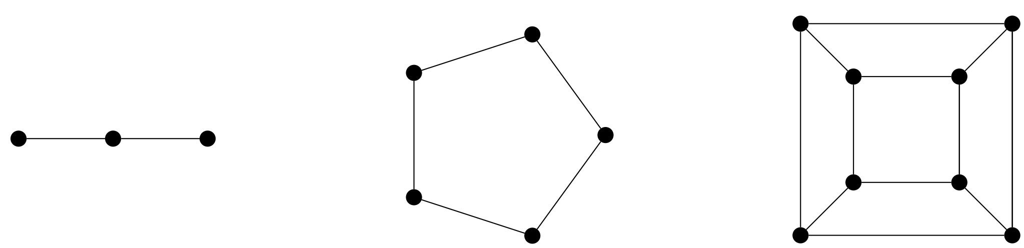
There is a specific variant of the game which makes the cops "lazy" that we intend to study.

(3L) Movement is confined to the edge set of G, and *at most* one person can move per turn.

As such, replacing rule (3) by rule (3L) obtains the rule set for "lazy cops and robbers."

The cop number of a graph and its lazy variant

If k-many cops can always capture the robber after a finite number of turns, regardless of the chosen starting configuration, we say G is *k-cop win*. The minimum number k for which G is k-cop win is denoted $c(G)$, and is called the *cop number* for G. Most of the research done in the field consists of computing the cop number of a given graph. There are known upper and lower bounds for the cop number of graphs with n vertices. However, focusing on certain classes of graphs can give us much better bounds. For



(a) $c(P_n) = 1$. (b) $c(C_n) = 2$. (c) $c(H_n) = \lceil \frac{n+1}{2} \rceil$.

Figure 1: Families of graphs with known cop numbers.

example, any path graph P_n has cop number $c(P_n) = 1$, since the cop can win by following the robber until the end of the path. Also, any cycle graph C_n has cop number $c(C_n) = 2$, since only two cops are needed to capture the robber from both sides—one simply isn't enough since the robber may evade capture indefinitely by proceeding along the cycle. A more interesting example is H_n, the n-dimensional hypercube. Within this family of graphs, the cop number is $c(H_n) = \lceil \frac{n+1}{2} \rceil$. As a final example, it is known that the Petersen graph as in Figure 2 is the smallest graph (by number of vertices) with cop number 3. It is an open question to find the smallest graph with cop number 4.

Much of the literature is about variants of cops and robbers such as "zombies and survivors," "drunk robbers," "freeze-tag," "fast robbers," and essentially any variation one can think about. We'd also like to study the lazy variant discussed above. The analogue of the cop number of a graph G is the *lazy cop number*, $c_{\mathrm{L}}(G)$, and G is similarly *k-lazy cop win* if k cops are enough to capture the robber for any starting configuration.

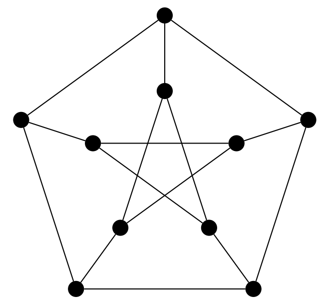


Figure 2: The Petersen graph is the smallest graph with cop number 3.

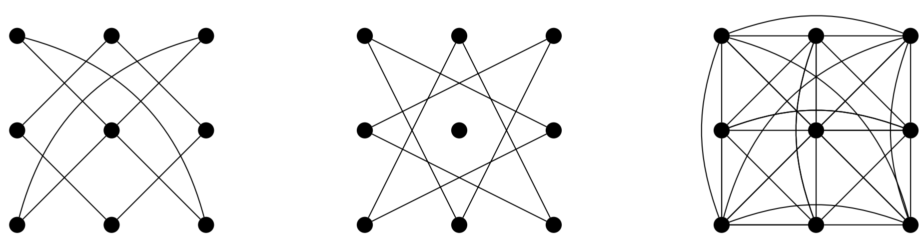
Denote by $\gamma(G)$ the *domination number* of G, that is, the smallest number of vertices needed in a set so that each edge of G has at least one vertex in the set. Note then that

$$c(G) \le c_{\mathrm{L}}(G) \le \gamma(G).$$

The first inequality holds since (3L) is a restriction of (3), while the second inequality holds since placing a cop on each vertex of a dominating set ensures a cop win in one turn.

Chess graphs and early results

A simple family of graphs are the *chess graphs* played on $n \times n$ square boards where movement is restricted to the movement types of a given chess piece. As an example, the $n \times n$ *rook graph* is the graph $\mathscr{R}_n = (\mathbb{Z}_n^2, E)$ where $((i,j),(i',j')) \in E$ if and only if $i = i'$ and $j \ne j'$, or $j = j'$ and $i \ne i'$. See Figure 4 for an illustration when $n = 3$. We similarly define the $n \times n$ *bishop*, *knight*, and *queen* graphs $\mathscr{B}_n, \mathscr{N}_n, \mathscr{Q}_n$, respectively, depicted in Figure 3.



(a) $\mathscr{B}_3$ (b) $\mathscr{N}_3$ (c) $\mathscr{Q}_3$

Figure 3: The other 3×3 chess graphs.

One can easily be convinced that $c(\mathscr{R}_n) = 2$ for all $n \ge 2$. It turns out that $\mathscr{R}_3$ is the smallest graph with lazy cop number 3. Rook graphs also serve as examples of graphs where the difference between a graph's cop number and its lazy cop number is arbitrarily large. Specifically,

$$2 = c(\mathscr{R}_n) \le c_{\mathrm{L}}(\mathscr{R}_n) = \gamma(\mathscr{R}_n) = n.$$

Bishop graphs also share this property. Similar to before, it is open to find the smallest chess graph with lazy cop number 4.

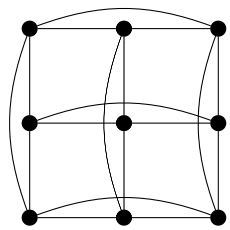


Figure 4: $\mathscr{R}_3$ is the smallest graph with lazy cop number 3.

Much less is known about queen graphs, and so they will be the initial set of graphs we will investigate. The values $c(\mathscr{Q}_n), c_{\mathrm{L}}(\mathscr{Q}_n)$ are known up to $n = 6$. It is also known that $c(\mathscr{Q}_n) = 4$ for all $n \geq 19$. It is not known if the (lazy) cop number of $\mathscr{Q}_n$ is monotonic in n, though it seems likely, which is significant since the cop number of $\mathscr{N}_n$ is decidedly not monotonic in n. We aim to prove this, and then find the smallest n for which $c(\mathscr{Q}_n) = 4$.

Barely anything is known about knight graphs, but given how approachable they are, we aim to study them promptly after the initial investigation of the queen graphs.

Proof techniques

Most methods of proof are graph theoretic and typically involve directly giving a winning strategy for either the cops or the robbers. Other proofs rely on a potential function acting as a measure of how "good" a position is for a given team, and they use this in proving that the cop number is monotonic on a given family of graphs.

Background

Prerequisites

Applicants should have experience with proofs and the discrete math pertaining to graphs. A light background knowledge of programming is a plus. No knowledge of chess is necessary. Applicants should be ready to spend considerable time on this to produce a paper in the end.

Texts and resources

Students will read and learn about typical arguments used to determine the (lazy) cop number of a graph, as well as the methods typical to pursuit-evasion games in general. A popular book on this topic is *The Game of Cops and Robbers on Graphs* by Anthony Bonato, and we intend to reference the text throughout the REU as its coverage of the topic and its variations is quite broad. However, most of our references and reading will come from published papers which are relatively digestible. Anthony Bonato's *Conjectures on Cops and Robbers* will serve as an expository reading which will introduce students to the biggest results and open problems in the subject. Papers more closely related to our problem include *An Introduction to Lazy Cops and Robbers on Graphs* by Sullivan, Townsend, and Werzanski as well as *Variations of Cops and Robber on the Hypercube* by Offiner and Ojakian.

We think these resources are enough to bridge any gap in knowledge should the applicant not meet the recommended prerequisites. We generally believe the arguments for most winning strategies are approachable by any first year student enrolled in Northeastern's MATH 1365.

Computations

This REU is essentially an exercise in computation. However, the crux is that the number of starting positions and subsequent moves is too large to manually verify the cop number of a graph, even for small n. However, taking advantage of the structure of the graph allows us to widely reduce the number of cases needed to check in the case that no elegant proof exists.

Students will be able to play the game and write code to check any conjectures they may have to help them come up with a winning strategy. We've already demonstrated that it is possible to write code to check whether or not a given graph is k-cop win for a given k, and can guide students in producing their own code to check their own conjectures.

Exploration

There are many families of graphs and many variations of cops and robbers. Students will be encouraged to create their own variations of the game, and to explore a family of graphs they find interesting.

We propose the following exploration as an example: consider the variant in which the robber is an aggressive king in that it can capture cops, and the cops are some collection of pieces. This variant then expands the behavior of the robber, and restricts the movements of players to particular *distinct* chess pieces. Allowing the cops to be different types of pieces, the lazy version of this variant is equivalent to a chess end game. For example, in an 8×8 chess board, it is known that a king and a rook can checkmate a king.

Timeline

Our itinerary is in accordance with the following weekly schedule. The main priority for the mentors of this REU is for the students to prove publishable results, and as such our timeline reflects that goal.

Week 1	• Offer a formal and informal introduction to the game. • Read *Conjectures on Cops and Robbers* to familiarize the students with widely-used terminology and types of questions those in the field are interested in. • Read *An Introduction to Lazy Cops and Robbers on Graphs* and introduce the students to the chess graphs, as well as the results of the paper regarding that family of graphs.
Week 2	• Find (lazy) cop numbers for $\mathscr{Q}_7, \mathscr{Q}_8$. • Think about a proof for the monotonicity of $c(\mathscr{Q}_n)$. If no elegant proof can be found, there are always exhaustive searches to be made. Monotonicity of $\gamma(\mathscr{Q}_n)$ is also open.
Week 3	• Continue the work on $\mathscr{Q}_n$ for $9 \leq n \leq 18$ from last week. • According to the insights in *An Introduction to Lazy Cops and Robbers on Graphs*, "a more careful consideration of the possible placement of the nonthreatening cops may yield a larger lower bound." Therefore we can modify the arguments made thus far to work for the lower bound of $c_{\mathrm{L}}(\mathscr{Q}_n)$.
Week 4	• Investigate $\mathscr{N}_n, \mathscr{B}_n$ for $n \geq 1$. • Find conjectures. Is $c(\mathscr{N}_n)$ bounded? Monotonic? Does $c_{\mathrm{L}}(\mathscr{N}_n)$ grow in n? Is $c_L(\mathscr{N}_n) = \gamma(\mathscr{N}_n)$?
Week 5	• Investigate the hypercube H_n for $n \geq 1$ with respect to lazy cop variant. • Find conjectures. Is $c_{\mathrm{L}}(H_n)$ bounded? Monotonic? How does $c_{\mathrm{L}}(H_n)$ grow with n? Is $c_{\mathrm{L}}(H_n) = \gamma(H_n)$?
Weeks 6–8	• Field trip to the beach for cops and robbers irl (this may be cut) • Start writing a paper. Make proofs formal. • Work on variants and families of graphs that the students and mentors find interesting. • Practice presentation.