東海大学大学院　平成29年度　博士論文

Winning Strategies for the Cops and Robbers Game on Planar Graphs

(平面的グラフにおける警察•泥棒ゲームの勝ち戦略に関する研究)

指導　　譚　学厚　　教授

東海大学大学院　総合理工学研究科

総合理工学専攻

Photchchara Pisantechakool

Dedication

To mum and dad.

Thank you for the unconditional love and supports you have been given me for all these years, through hardships, long distance, and even the most busy time of my life.

Love always,
Mon

Acknowledgments

First, I would like to express my most sincere appreciation and gratitude to Professor Xuehou Tan for his kind support and guidance all along this doctoral course. Through his passion and enthusiasm in research, I have been able to make it this far, to finally finishing my doctoral thesis. Those precious lessons and passions will always be a part of me, no matter what path or career I will take.

My deepest gratitude to the chair of thesis committee, Professor Osamu Uchida, for his patience, kindness and wiseness all along the evaluation process of my work. Also, I would like to sincerely thank the other committee members, Professor Kagayaki Kuroda, Professor Takeshi Asakawa, Professor Mitsuhiro Toratani, and Professor Toshinori Sakai for their remarks and advices that helped in the improvement of this work.

I would like to express my appreciation to Professor Harumi Murakami for supporting and advising me for so long since my first moment here in Japan. I learned my Japanese from her, then became a research student under her guidance while I looked for my thesis advisor on doctoral course, and even consulted her on Japanese abstract and presentation of this thesis.

I am grateful to the staff of our department office, and other institution inside Tokai University that supported me in all my procedures, particularly to Aoi Mano for her kindness and her help in all the procedures I have to go through as international student.

Finally, I would like to thank Tokai University for financial and mental supports to this research.

Contents

List of Figures

Main Terms

2-cop-winning vertex	the vertex at which two cops can capture the robber
capture time	the number of rounds it takes for the cops to capture the robber
cop number	the minimum number of cops required to successfully capture the robber in the given graph
dominating set	a subset of vertices such that every vertex of the given graph not in the subset is adjacent to at least one member of the subset
Euler's formula	For any planar graph $G = (V(G), E(G))$, the following is true: $\lvert V(G)\rvert - \lvert E(G)\rvert + \lvert F(G)\rvert = 2$
guarded shortest path	a shortest path controlled by a cop, such that the robber will get caught by the cop if he moves to any vertex on the path
robber territory	a subgraph which has all vertices that the robber can safely enter

Frequently Used Symbols

$d(v)$	degree of vertex v
$\delta(G)$	the smallest degree of all vertices in G
$\pi(u, v)$	shortest path between vertices u and v
$c(G)$	cop number of graph G
$capt_k(G)$	capture time of graph G using k cops
$G[S]$	subgraph of graph G induced by a set of vertices S
R_i	robber territory at stage i of the strategy
$C(G)$	set of cycles whose all vertices belong to the infinite face of graph G
$S(F(G))$	sum of the edges of all faces in $F(G)$

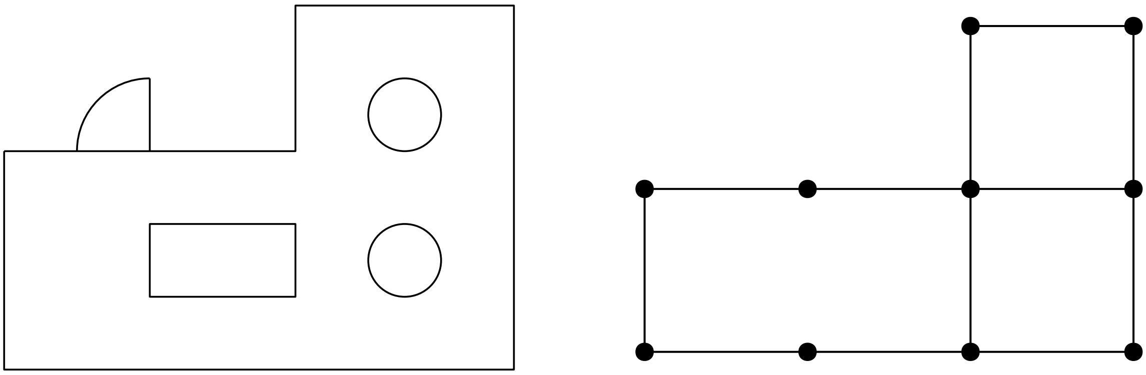
Chapter 1

Introduction

1.1 Background

In almost every culture, and perhaps even animal kingdoms, there are games that children play. The simplest form of social interaction between the young ones, as well as the practice for survival skill, is the game of chasing. It is so simple that it can be played with two players and play field. The game usually ends with the fastest runner's win if the play field is simple. However, when it becomes more complex, with impassable obstacles, limited play field, or even restricted to an equal movement speed, it becomes the battle of wits.

Such a simple game has turned into many forms and over various platforms, such as the PAC-Man video game, pursuit-evasion [25], or the Cops and Robbers game [8]. One player would play as a pursuer, and the other an evader, and the goal for pursuing player is to catch the evader, while the evader's goal is to avoid being caught indefinitely. When played on a graph [25, 26], each player can only occupy a vertex at any time, and can only move through an edge. To make the game fair, each player takes turn to move, usually the pursuer being the first to go into the play field, allowing the evader to think of the best spot to start (as far away from the cop as possible). More often, the pursuer is referred to as a female, and the evader as a male. The earliest of the video game were played on a discrete setting, see Fig. 1.1 on how the



(a) A room for chasing game. (b) A corresponding graph.

Figure 1.1: Changing the play field from room environment into a graph

play field is changed into a graph.

For graph theory [10, 32], it was first considered in 1978 by Quilliot in his doctoral thesis [30]. However, due to the language barrier (his was published in French), Nowakowski and Winkler independently considered and published their own result [23] in 1983. Although [30] predated [23], the latter was often referred to as the starting point of literature on the topic. Both works came to the same conclusion, and both only considered one cop. But the work which is most referred to, and the foundation of all Cops and Robbers research, came from the work of Aigner and Fromme [1]. They have solidified the result on one cop, introduced multiple cops, and proved that three cops are sufficient to win on a planar graph. In their words, any graph that requires a minimum k cop to win is called k-cop-win graph, although when $k = 1$ it is simply called cop-win graph. Since then, many papers on this topic have been written about the *cop number*, which is the smallest number of cops required to win.

1.2 Cops and Robbers Game on Planar Graphs

A graph is planar if and only if it can be drawn on a plane such that its edges do not intersect anywhere except at the vertices. For any planar graph, Euler has provided a theorem on the relationship between the number of vertices, edges, and faces [32]. Moreover, the cop number for a planar graph has been proved to be at most three [1]. These two factors make it simpler for the Cops and Robbers game if played on a planar graph. For the cop number problem, Maurer et al. have made a report on many different planar graphs in 2010 [21].

In many variations of the Cops and Robbers game that are focused on the application, the play field is usually set in a planar graph (as it is a simplified version of any map or layout). In the game of Helicopter Cops and Robbers, the cops can move to any locations (except on top of the robber) in one round. This simulates the real world scenario where the cops can use helicopters to spot and chase the criminals down. In the game of Firefighter, the firefighters must put down the fire and can move to any location which is not burning. In other words, it is similar to Helicopter Cops and Robbers but the opponent can multiply itself to adjacent vertices. The Firefighter variation is modeled to simplify the spread of fire, diseases, or even computer viruses. This problem is often set in grid setting (mapping a set of floor tiles into one vertex).

Moreover, the transformation from discrete setting (graph) to continuous (geometric or polygons) is easier to do so with planar graph [6]. Thus, in order to apply the problem to real-world uses, it is easier to consider the graph problem on planar graphs.

Another approach to the problem is to consider how long does it take the cop player to win in a graph, or the *capture time* of a graph. This was first introduced by Bonato et al. in 2009 [9] on any cop-win graph, and later extended on to grids, which are known to be 2-cop-win graphs, by Mehrabian [22].

In our study, the linear capture time for general planar graph is established [28]. An important fact that is required in any 2-cop-win graph is identified as a new concept of winning vertices for two cops. As a very first attack to the conjecture that dodecahedral graph is the smallest 3-cop-win planar graph, we show the existence of wining vertices in all the planar graphs of order at most 19 [29].

1.2.1 Pursuit-Evasion Game's Variations

The original papers ([23, 30, 31]) only considered one cop playing against one robber, and referred to the topic as pursuit game. The introduction of multiple cops in [1] gave the name of the topic as it is used now, the "Cops and Robbers" game, with plurals on both sides. However, the new variation on how each player can "see" one another [15] has made the name "Cops and Robbers" seem out of place, as the game becomes more of finding patrol routes for guardsmen. To accommodate the new variation, the general term "pursuit-evasion game" was adopted from the earlier works of Parsons [25, 26].

Pursuit-evasion game consists of two variations; the Cops and Robbers game, in which both players can see each other's locations, and the graph searching game, in which the pursuer does not know the location of the evader until they occupy the same space (the same vertex in a graph). It also includes all the variations in between, such as being able to "see" the robber if the cop is within certain number of edges away from him, or even allow the robber to occupy an edge instead of just a vertex. Many variations on limited visibility for the cop player are detailed in the work of Alspach [3]. Two approaches to find the cop number are often asking whether *recontamination* should be allowed or not. Some tried to prove that the cop number can be reduced if recontamination is allowed (see [7]), while other strictly focused on monotonic search approach (see [20] and [19]).

Recently, a new take on the game took it closer to real world setting. It took a different approach on the play field; instead of discrete setting of a graph, a continuous setting inside a polygon was considered [6]. Despite the differences in settings, the result of Aigner and Fromme [1] played the key part in proving that even with obstacles, or holes in polygons, three cops suffice.

Note that for continuous setting inside a polygon, the graph searching counterpart is then the visibility problem in geometry. If the capture time is limited to the start of the game, and ask how many pursuers are needed, then the problem can be reduced to the Art Gallery problem in computational geometry.

1.2.2 Applications

The direct applications of the Cops and Robbers game are in chasing or capturing objectives, such as the missile guidance system [18], but it also has many applications beyond such scope. The Firefighter game is modeled after the spreading of fire, but also has its use in stopping computer viruses, or even diseases, from spreading. Although the "helicopter" movement seems too unrealistic for actual firefighting, it has perfect application in containing the computer viruses in networks. Network infrastructures or hubs can be shut down or armed with firewall by alerting the staffs at the infrastructure through telephones in global scale problem, or disconnected manually in local scale. The Firefighter strategy can select the fewest possible stations or hubs to minimize the impact on network traffics, and stop the spread of viruses effectively.

Some other applications of the Cops and Robbers game are as follow.

1. Computer Gaming: Originally the Cops and Robbers game is similar to that of PAC-man game, but the player plays the robber instead. In modern games played on mobile phone, where players must go to certain locations in the real world to do objectives, the strategy used in Cops and Robbers can be applied. For example, in "Pokemon Go", players can work together to find the Pokemon sighted near their area, similar to cops working together to capture the robber.

In "Zombies, Run!" mobile game for fitness, a player has to run away from virtual zombies (shown on the map in mobile phone) in the real world. The difficulty of the game can be adjusted based on Cops and Robbers strategy on planar graphs. For example, in easy game, one zombie cannot catch the player unless the player runs into a dead-end. In the harder difficulty, there can be three or more zombies, but only two of them are as fast as the player and follow the capture strategy (such as going through different roads instead of just chasing the player), so the player may still have a chance to escape. In the hardest difficulty, the three zombies are as fast as the player and follow the capture strategy.

In multi-players shooting game, simulation for SWAT, or even air force training [24], the A.I opponents can be implemented to use the robber's strategy in order to escape. This forces the players in one team to work together using the cops' capture strategy in order to win.

2. Motion Planning for Swarm Robots: In a war, low-cost swarm robots can be sent into a region and set up explosions. It is important that the collective explosion radii must cover the entire region to maximize the damage, while the number of robots (or the more expensive bombs) in that region should be kept minimal [14]. This problem is similar to the game of *absolute-win* Cops and Robbers. In this game, the cops are initially placed such that the robber would occupy the same vertex as some cop or the neighbor of some cop no matter which vertex he chooses in the first round, with the smallest number of cops. This is also the same problem as the dominating set in graph, as the initial locations for the cops are actually the members of the dominating of the smallest size. If the graph is a grid, it can be mapped to a rectangular region, in which the explosion on a vertex would have the radius to cover the adjacent vertices. The dominating set of the grid is then the set of locations that the robots must detonate their bombs in order to cover the entire region.

Swarm robots do not operate by following inputs from central control unit. Instead, they follow a preprogrammed behavior, which act or react on local inputs, such as sensing environments, communications with other robots or internal timer. The preprogrammed behavior is called distributed algorithm, and handle how the robots respond to local inputs in order to achieve the predetermined goal (in this example, dominate the region with explosion). Swarm robots can be equipped with other functionalities, such as GPS or explosives. There are other applications using the same distributed algorithm, based on the functionalities equipped to the robots. They can be equipped with short-ranged motion detectors to search the region for missing persons or intruders. When armed with land mine detectors, they can be dropped from the plane onto a battlefield to mark the areas with land mines. In the area where other forms of communication are cut off, they can be employed to set up relay nodes to extend the radiophone range.

3. Ocean Rescue: In the Helicopter Cops and Robbers [8], the cops can move on helicopters and move faster than the robber. Instead of capturing criminals, it can also be applied to rescue shipwrecked survivors in the ocean. Because the ocean waves can sweep some survivors away, the objectives may not stay in one spot, which is somewhat similar to the robber behavior in Cops and Robbers game. The strategy for normal Cops and Robbers game on planar graph can also be used to reduce the "search and rescue" area. This is done by using the shoreline as natural barrier and two helicopters to set up two shortest paths (creating triangular area with shoreline) to contain the area that survivors can possibly be swept to.

1.3 Graph Notation

A graph $G = (V(G), E(G))$ is defined as a set $V(G)$ of vertices which are connected by a set $E(G)$ of edges [32]. The number of vertices is denoted by $|V(G)|$, and the number of edges is denoted by $|E(G)|$. A cycle, denoted by C, is a path whose start vertex and end vertex are the same. A graph is *regular* if every vertex has the same degree.

A graph is *planar* if and only if it can be drawn on a plane such that its edges only intersect at the endpoints (which are vertices). And such, every face of a planar graph is unique. For a planar graph G, $F(G)$ is defined as a set of faces, which are also called the cycles of

G. The number of faces is denoted by $|F(G)|$. For a face $f \in F(G)$, the number of its sides is denoted as $S(f)$, and also called the length of cycle f. Note that a cycle may consist of multiple faces, but each face has exactly one unique cycle. The graph's girth, denoted by $g(G)$, is the size of its smallest cycle. A *cut vertex* of G is a vertex such that when removed, either the number of connected components of G is strictly increased or G becomes a single vertex. For a vertex set $S \subseteq V(G)$, $G[S]$ is defined as the subgraph of G that is induced by the vertex set S, i.e., an edge $e \in E(G)$ belongs to $G[S]$ if two vertices of e belong to S.

The *distance* between two vertices v and u is the length of a shortest path between them, and is denoted by $d(v, u)$. A shortest path between u and v is denoted by $\pi(u, v)$, and $|\pi(u, v)|$ denotes the length of the shortest path between u and v (and thus the distance as well). The *diameter* of G, denoted by $diam(G)$, is the largest distance between any two vertices of G. If two shortest paths $\pi(a, u)$ and $\pi(b, v)$, of length at least two have all different vertices, excluding a, b, v, and v, then we say that $\pi(a, u)$ and $\pi(b, v)$ are *distinct.*

For a vertex v, the *degree* of v, denoted by $d(v)$, is the number of the neighbors of v. The minimum degree of a graph G, denoted by $\delta(G)$, is the smallest degree of all vertices in G. Let $N(v)$ denote the set of all neighbors of vertex v, and $\bar{N}(v) = N(v) \cup \{v\}$. For any vertex v, if there exists a vertex $u \in N(v)$ such that $\bar{N}(v) \subseteq \bar{N}(u)$, then v is called a *dominated* vertex.

1.4 Problem Statement

The game of Cops and Robbers is played by two players on a graph G; one controls the cops (cop player), another controls the robbers. By the nature of Cops and Robbers game, graph G must be connected; otherwise the cops may not be on the same connected component as the robber, and thus it may not be cop-win. We focus on the variation that plays on a simple, undirected, planar graph. The game model we use is based on k-cops model introduced in [1], in which the cop player controls multiple cops, and the robber player controls one robber. Both players know the locations of one another's pieces.

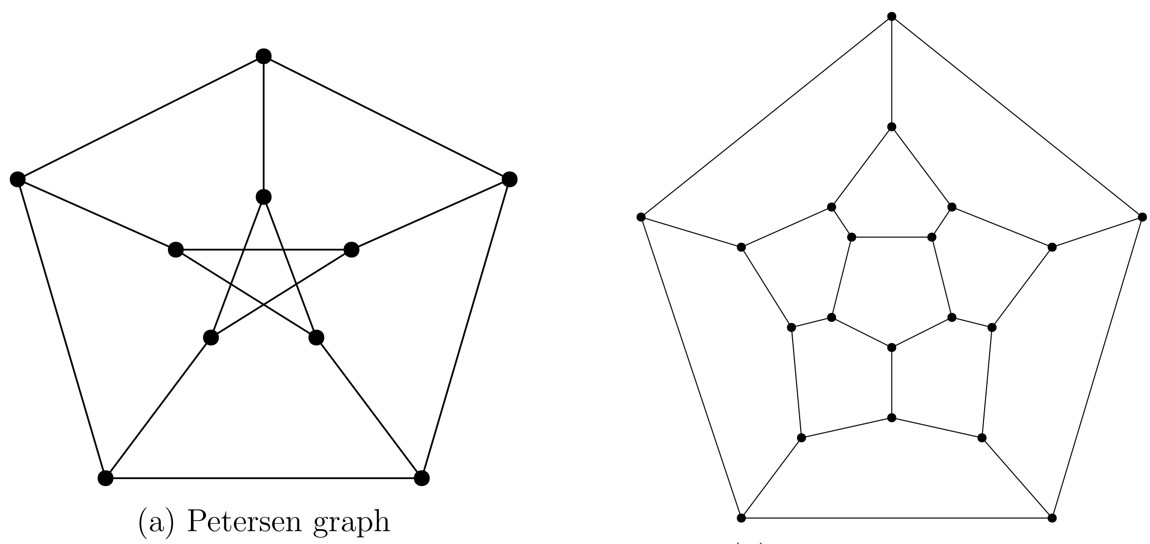
The game is played as follow. First, the cop player places her cops on some vertices of G, and then the robber places his. Two players take turns to move their pieces. A piece can only be moved to one of its adjacent vertices in one turn, and the cop player can move multiple cops simultaneously. A *round* in the Cops and Robbers game consists of a cop player's turn and a robber player's turn, and the first round in which each player places one's own pieces on G is considered as Round 0. If any cop occupies the same vertex as the robber (captures the robber in this model), then the cop player wins. If the robber can avoid being captured indefinitely, then the robber wins. The minimal number of cops required to win on a graph G, or the *cop number* of G, is denoted by $c(G)$. A graph G is said to be k*-cop-win* if and only if $c(G) = k$.

For a planar graph G, it is known that $1 \leq c(G) \leq 3$ [1]. A method to determine whether a graph is cop-win was given in [23, 30]. However, the method to determine whether a planar graph is 2- or 3-cop-win has not been fully given yet. Also, the conjecture that a dodecahedral graph (of order 20) is the smallest 3-cop-win graph has not been proved yet. It is one of the main problems studied in this thesis.

The problem of the *capture time* in the game of Cops and Robbers is to find how long it takes for the cop player to capture the robber. The unit of measurement used in this study is the number of rounds. For a planar graph, the bound has not been established yet, but there already exists a winning strategy using three cops [1]. However, the capture time for that strategy is observed to be quadratic [28]. In this thesis, we will present a new strategy of linear length for planar graphs.

1.5 Previous Results

We review some known results which are related to our works.



(a) Petersen graph

(b) Dodecahedral graph

Figure 1.2: Petersen graph and dodecahedral graph

Theorem 1.1. *[[23], Theorem 1, and independently [30]] Graph G is a 1-cop-win if and only if by successively removing dominated vertices, G can be reduced to a single vertex.*

Theorem 1.1 can be applied to any tree, since we can simply chase down the robber from some vertex to a dead end at some leaf.

Theorem 1.2. *[[1], Theorem 3] If a graph G has $g(G) \geq 5$ then $c(G) \geq \delta(G)$.*

Theorem 1.2 gave useful insight on determining the cop number for any graph without any 3- or 4-cycle. Aigner and Fromme have also given two examples along with the proof; a Petersen graph (Fig. 1.2(a)) and a dodecahedral graph (Fig. 1.2(b). Both of which were conjectured to be the smallest 3-cop-win graph and planar graphs, respectively, as they are 3-regular (every vertex is of degree three) without any 3- or 4-cycle. Only until recently, the former conjecture was proved by Baird et al. [5].

Theorem 1.3. *[[5], Theorem 1 and 2] The Petersen graph is the smallest 3-cop-win graph.*

The proof for Theorem 1.3 is rather straightforward (by showing that all the possible graphs of order at most 10, except Petersen graph, are 2-cop-win). The following observation can be made.

Observation 1.4. For any planar graph of order 10 or smaller, the cop number is at most two.

The conjecture that the dodecahedral graph is the smallest 3-cop-win planar graph has not yet been proven. The dodecahedral graph is a planar graph of order 20, whose all vertices are of degree three and all faces are 5-cycles, as shown in Fig. 4.7(a). From Theorem 1.2, the following observation can be made.

Observation 1.5. The dodecahedral graph is 3-cop-win.

Also, for a planar graph, the largest cop number was proved to be three.

Theorem 1.6. *[[1], Theorem 6] For any planar graph G, the cop number of G is at most three.*

Aigner and Fromme's proof of Theorem 1.6 provided a series of concepts and tools that can be used in actual capture strategy. One is the concept of the *guarded paths*, is devoted to diminishing the area in which the robber can move freely.

Lemma 1.7. [Guarded Shortest Path [1], Lemma 4] Let G be any graph, $u, v \in V(G)$, $u \neq v$ and $P = \pi(u, v)$. We assume that at least two cops are in the play. Then a single cop c on P can, after the movements no more than twice the diameter of G, prevent the robber r from entering P. That is, r will immediately be caught if he moves into P.

It is imperative that we provide the proof of Lemma 1.7 as well, particularly for the new claim that "a single cop c on P can, after the movements no more than twice the diameter of G, prevent r from entering P".

Modified Proof of Lemma 1.7. Suppose the cop c is on vertex $i \in P$ and the robber r is on vertex $j \in V(G)$. Assume $\forall z \in P, |\pi(z,j)| \geq |\pi(z,i)|$; denote this as (*).

Claim A. No matter what the robber does, the cop, by moving in the appropriate direction on P, can preserve condition (*). If the robber r does not move, then neither does c, and (*) holds. If r moves to a vertex k, then $\forall z \in P, |\pi(k,z)| \geq |\pi(j,z)| - 1 \geq |\pi(i,z)| - 1$. If $z_0 \in P$ exists with $|\pi(k,z_0)| \geq |\pi(i,z_0)| - 1$, then c, by moving on P toward z_0, also reduces the distance by 1 and (*) still holds. In the case that such a z_0 does not exist, there must be some vertices $x, y \in P$ such that they are on the different sides of i on the path P, and $|\pi(k,x)| = |\pi(i,x)| - 1$, $|\pi(k,y)| \leq |\pi(i,y)|$ or $|\pi(k,x)| \leq |\pi(i,x)|$, $|\pi(k,y)| = |\pi(i,y)| - 1$. This is impossible, since by the triangle inequality and minimality of P; $|\pi(x,y)| \leq |\pi(k,x)| + |\pi(k,y)| \leq |\pi(i,x)| + |\pi(i,y)| - 1 = |\pi(x,y)| - 1$, a contradiction.

Claim B. It takes a number of movements no more than twice the diameter of G for c to enforce (*). First, c moves to some $i \in P$, which takes at most $diam(G)$ movements. By the same argument as described above, $|\pi(j,z)| < |\pi(i,z)|$ only holds for z's on P on one side of i. By moving toward z on P, which takes at most $|P|$ or $diam(G)$ movements, (*) is eventually forced. □

By Lemma 1.7, the cops can limit the robber's movement to one side of the path, and thus diminishes the area the robber can safely enter. Aigner and Fromme have introduced two related concepts, which are the *stages* and the *robber territories*, in order to construct a method for three cops to capture a robber, as the proof of Theorem 1.6.

Definition 1.8. [[1], Proof of Theorem 6] The stage i, $0 \leq i \leq t$, is the assignment of a subgraph R_i which has all the vertices the robber can still safely enter. The assignment of R_i is done after the cop player has fixed her pieces at the end of stage $i-1$, and we assume $R_0 = G$. The subgraph R_i is called the robber territory.

We assume that stage 0 exactly coincides with Round 0 and thus $R_1 = G[V(G) - \{v\}]$, where v denotes the vertex initially occupied by the cops. At a stage i (≥ 1), the cop player constructs a new guarded path so that R_i is reduced to $R_{i+1} \subsetneq R_i$. Thus, stage i may consist of several rounds. The *length* of stage i is then the number of rounds it takes for the cop player to fix her pieces, and the length of stage 0 is assumed to be zero.

At a stage, an unoccupied (free) cop c will move to position herself on some vertex of a guarded path P so as to enforce the condition (*), i.e., be on the vertex such that she takes less time than the robber to move to any other vertex on P. Until (*) is enforced by c, the robber r may cross P safely. Once (*) is enforced and constantly preserved by c, r can no longer cross P without being captured. The locations satisfying (*) on P change as the robber moves, and there always exists at least one at any time [1].

Observation 1.9. If a cop c is already on P, then the number of rounds it take for c to eventually enforce (*) is bounded above by the length of P.

Observation 1.10. When a cop successfully controls (i.e., enforce (*) on) a shortest path, all vertices of that path do not belong to the robber territory.

From Lemma 1.7, capturing the robber can be done by letting the cops alternatively take the role of a free cop and move to guard a new shortest path within the robber territory. Once the free cop successfully guards the path, she becomes occupied and another cop, whose guarded path no longer interacts with the robber territory, becomes free at the next stage. At stage j ($j > i$), the path once guarded at stage i by a now-free cop is called an *obsolete path*.

Aigner and Fromme's method is to repeatedly find a new guarded path that differs from previous ones (they are not required to be distinct), until the robber territory is eventually reduced to one vertex. However, the capture time of their strategy is not examined in [1].

Remark. From Lemma 1.7, the length of each stage of Aigner and Fromme's strategy [1] is bounded by $2diam(R_i)$, or loosely by $2|V(R_i)|$, as their guarded paths are usually not distinct [28]. Suppose each stage only reduces one vertex in the worst case. Then, the capture time of Aigner and Fromme's strategy can roughly be bounded by $\sum_{i=1}^{n} 2i = n(n-1)$. Note that the capture time of Aigner and Fromme's strategy may be much faster than $O(n^2)$, but it needs a more careful calculation, so as to give a smaller bound.

1.6 The Objective of This Thesis

The capture time for planar graphs using three cops has been observed to be quadratic. This capture time is still very loose and the strategy was not constructed with capture time in mind. It is an open problem to propose a new strategy whose capture time is linear.

Also, for a planar graph, although the maximum cop number is know to be three, a method to determine whether a graph is 2- or 3-cop-win has not been given. Finding a method to determine 2-cop-win will also lead to finding another, using pigeonhole principle.

The major objectives of this study are as follow:

1. We present a new 3-cop-win strategy on a planar graph and show that its capture time is linear. This is the first result with the linear capture time for planar graphs. The strategy and the evaluation of its capture time involves many new concepts which requires deep observation. A few notable concepts are the method of outer cycles used to select end vertices for the guarded path, the types of movements that contribute to the capture time, and a life time of a path that can be mapped to some types of movements made by the cops.

2. We try to prove the conjecture that the smallest 3-cop-win planar graph is the dodecahedral graph. Although the problem is still open, we show that in any 2-cop-win planar graph, there exists a winning vertex in which two cops can capture the robber. We also prove that any planar graph of order at most 19 has such a vertex. This finding supports the conjecture. We also examined many classes for planar graphs, and found some new 3-cop-win and 2-cop-win graphs, along with the proofs. Some notable are 3-regular planar graphs whose all vertices are winning vertices for two cops, and a special graph of order 42 with a few winning vertices.

3. We give an application of the dominating set problem, which is related to the Cops and Robbers game, particularly in swarm robots. The absolute-win Cops and Robbers game requires the cop player to win on the first turn (after initial placement) using the smallest number of cops possible. This problem is exactly the same as the dominating set in graph. The distributed algorithm is required for swarm robots, since they do not have centralized control unit and rely solely on preprogrammed behavior and communications with other robots. A distributed algorithm for computing the dominating set on grid is proposed.

1.7 The Organization of This Thesis

This thesis is organized into seven chapters. A brief outline is as follows:

Chapter 1 In this chapter, brief explanations on the game of Cops and Robbers are presented. In addition, the background of the study, problem statement, previous researches, research objectives, and organization of the thesis are described.

Chapter 2 This chapter proposes the new strategy for the Cops and Robbers game using three cops on planar graphs with improved capture time. The strategy revises upon the method of Aigner and Fromme in such that all guarded paths are distinct, excluding the end vertices. The important concepts of outer cycles and its enlarged counterpart are described. The enlarged outer cycle is used to find an end vertex of a guarded path, such that the robber territory is reduced in size much more quickly than the old method. The strategy is separated into two phases; initial phase, which handles the initial placement and the first round, and recursive phase, which is case-base and repeated until the robber territory becomes a tree. At any point of time, if the robber territory becomes a tree, one cop is then employed to chase the robber until the dead end at some leaf, and the game ends as the cop player's win. The strategy is proved to be correct and complete.

Chapter 3 This chapter discusses the evaluation of the capture time on the strategy proposed in Chapter 3. Only some movements done by the cops affect the length of a stage. The types of movements which contribute to the capture time are identified and discussed. The guarded paths are treated as a state machine in order to map the movements of the cops onto the path's length. The movements of the cops which contribute to the capture time is mapped to the portions of a path. It is proved that any path is used by such movements for no more than twice of its length. The linear time of $2n$, where n is the number of vertices of a planar graph, is obtained. This greatly improved upon the previous time bound of $O(n^2)$

Chapter 4 This chapters introduces the new concept of winning vertex, and the fact that it must exist in any 2-cop-win graph. It is then used to attack the conjecture that the dodecahedral graph of order 20 is the smallest 3-cop-win planar graph, by showing that any planar graph of order at most 19 has at least a winning vertex. The contradiction proof method is employed to show the existence of cycles of length five, as well as the winning vertex of degree three and four in planar graph whose minimum degree are three and four. The existences of a winning vertex in all planar graphs, whose minimum degrees are 2, 3, 4, and 5 are showed (for a planar graph, the largest minimum degree is 5). An edge-contracted dodecahedral, which may be considered as the worst case in finding a winning strategy using two cops, and the 3-regular graph of order 16 (the largest one whose order is below 20), are showed to be 2-cop-win as the capture strategies utilizing winning vertices are provided.

Chapter 5 This chapter provides a list of 3-cop-win and 2-cop-win graphs, most of them are newly found in this thesis. Fore example, two vertex-symmetric graphs (icosidodecahedral and truncated icosidodecahedral), in which all vertices are winning vertices for two cops, are 3-cop-win. Some other graphs with winning vertices for two cops, such as Grinberg's 42 graph and also a maximal planar graph constructed from dodecahedral, are 3-cop-win. This study may help to give the characterization of 2-cop-win or 3-cop-win planar graph.

Chater 6 This chapter introduces the dominating set problem and the distributed algorithm to calculate the set in grids. The relationship of Cops and Robbers game and the dominating set problem is discussed. Previous results in centralized method is briefly reviewed and simplified. A new distributed algorithm for computing the dominating set in grids is proposed. For a given $m \times n$ grid, our distributed algorithm improved upon the number of robots required in the previous algorithm, from $\lfloor \frac{(m+2)(n+2)}{2} \rfloor + 1$ to $\lfloor \frac{(m+2)(n+2)}{2} \rfloor - 3$. This is the closest to the lower bound of $\lfloor \frac{(m+2)(n+2)}{2} \rfloor - 4$ [12].

Chapter 7 In the last chapter, the overall conclusion of the performed study is given. Some insights and recommendations for further study are also presented.

References

[1] M. Aigner and M. Fromme. "A game of Cops and Robbers". In: *Discrete Applied Mathematics* 8.1 (1984), pp. 1–12.

[3] B. Alspach. "Searching and sweeping graphs. a brief survey". In: *Le Mathematiche* 59 (2004), pp. 5–37.

[5] W. Baird et al. "On the minimum order of k-cop-win graphs". In: *Contributions to Discrete Mathematics* 9 (2014), pp. 70–84.

[6] D. Bhaduaria et al. "Capturing an evader in polygonal environments with obstacles. the full visibility case". In: *The International Journal of Robotics Research* 31.10 (2012), pp. 1176–1189.

[7] D. Bienstock and P. Seymour. "Monotonicity in graph searching". In: *Journal of Algorithm* 12 (2011), pp. 239–245.

[8] A. Bonato and R. J. Nowakowski. *The Game of Cops and Robbers on Graphs.* Student Mathematical Library 61. Providence, Rhode Island: American Mathematical Society, 2011.

[9] A. Bonato et al. "The capture time of a graph". In: *Discrete Mathematics* 309 (2009), pp. 5583–5595.

[10] A. Bondy and U. Murty. *Graph Theory.* Graduate Texts in Mathematics 244. London: Springer-Verlag, 2008.

[12] G. J. Chang. "Domination numbers of grid graphs". PhD thesis. University of South Florida, 1992.

[14] E. Fata, S. L. Smith, and S. Sundaram. "Distributed dominating sets on grids". In: *2013 American Control Conference.* Ed. by T. Gopal, G. Jaeger, and S. Steila. New York: IEEE Press, 2013, pp. 211–216.

[15] S. Gal. *Search Games.* Academic Press, 1980.

[18] R. Isaacs. *Differential games. A mathematical theory with applications to warfare and pursuit, control and optimization.* SIAM series in applied mathematics. New York: Wiley, 1965.

[19] A. Kehagias, G. Hollinger, and S. Singh. "A graph search algorithm for indoor pursuit-evasion". In: *Mathematical and Computing Modeling* 50 (2008), pp. 1305–1317.

[20] A. S. LaPaugh. "Recontamination does not help to search a graph". In: *Journal of the ACM* 40 (1993), pp. 224–245.

[21] A. Maurer, J. McCauley, and S. Valeva. *Cops and Robbers on planar graphs.* Summer 2010 Interdisciplinary Research Experience for Undergraduates. report. University of Minnesota, 2010.

[22] A. Mehrabian. "The capture time of grids". In: *Discrete Mathematics* 311 (2011), pp. 102–105.

[23] R. J. Nowakowski and P. Winkler. "Vertex-to-vertex pursuit in a graph". In: *Discrete Mathematics* 43 (1983), pp. 235–239.

[24] W. L. Othling Jr. “Applications of differential game theory to pursuit-evasion problems of two aircrafts”. PhD thesis. Air Force Institute of Technology, 1970.

[25] T. D. Parsons. “Pursuit-evasion in a graph”. In: *Theory and Applications of Graphs. Proceedings.* Ed. by Yousef Alavi and Don R. Lick. Lecture Notes in Mathematics 642. Heidelberg, Berlin: Springer, 1978, pp. 426–441.

[26] T. D. Parsons. “The search number of a connected graph”. In: *The Ninth Southeastern Conference on Combinatorics, Graph Theory, and Computing. Proceedings.* Ed. by F. Hoffman. Congressus Numerantium 21. Heidelberg, Berlin: Utilitas Mathematica Publishing Incorporated, 1978, pp. 549–554.

[28] P. Pisantechakool and X. Tan. “On the capture time of Cops and Robbers game on a planar graph”. In: *Lecture Notes in Computer Science* 10043 (2016), pp. 3–17. Springer.

[29] P. Pisantechakool and X. Tan. “On the conjecture of the smallest 3-cop-win planar graph”. In: *Lecture Notes in Computer Science* 10185 (2017), pp. 499–514. Springer.

[30] A. Quilliot. “Jeux et pointes fixes sur les graphes”. Thèse de 3ème cycle. French. PhD thesis. Université de Paris VI, 1978, pp. 131–145.

[31] A. Quilliot. “Some results about pursuit games on metric spaces obtained through graph theory techniques”. In: *European Journal of Combinatorics* 7.1 (1986), pp. 55–66.

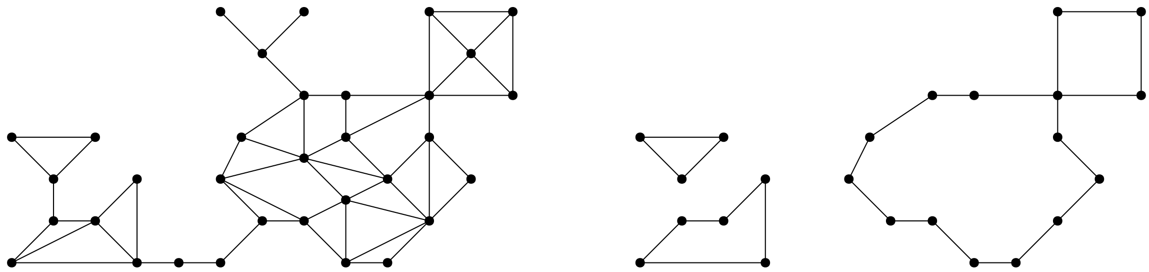
[32] D. B. West. *Introduction to Graph Theory.* 2nd ed. Prentice Hall, 2000.

Chapter 2

A New 3-Cop-Win Strategy on a Planar Graph

2.1 Introduction

In this chapter, we focus on the capture time of the Cops and Robbers game on a planar graph, which has only been studied recently [9, 22, 28]. We present a new capture strategy by refining the work of Aigner and Fromme [1] in the following two sides: (i) a new guarded path introduced at a stage shares only its end vertices with any current path, and (ii) the end vertices of a newly introduced guarded path are on or very close to the *outer cycle*, whose all vertices belong to the infinite face of the robber territory. These two refinements are involved, specially the second needs some deep observations. All guarded paths in our strategy are so chosen that they are almost distinct, excluding their end vertices. A strategy with capture time less than $2n$ can then be obtained.



(a) Robber territory R_i. (b) Outer Cycle $C(R_i)$.

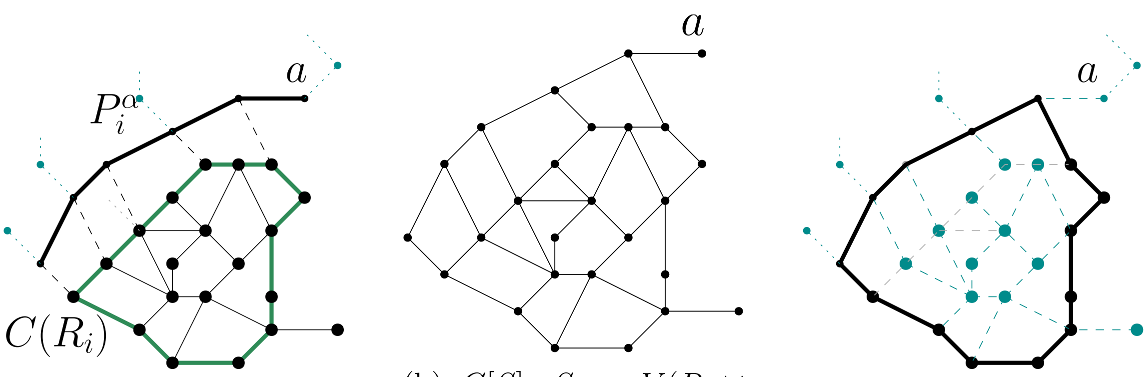
Figure 2.1: A robber territory and its outer cycles.

2.2 Outer Cycles of a Planar Graph

In order to establish a better upper bound on $capt_3(G)$, we make two refinitions over [1]. The first one is that a new path shares only its end vertices with any current path. (In [1], a new path may share more than just end vertices with a current path.) The guarded paths in our strategy are almost distinct, excluding their starting/ending vertices. Our second refinition is to choose the end vertices of the new guarded path to be on or very close to the infinite face of robber territory. To precisely describe how to choose the end vertices, we need a new concept called the *outer cycles*.

Definition 2.1. The subgraph $C(R_i)$ of R_i is defined as the set of the outer cycles, whose all the vertices and edges belong to the *infinite* (exterior) face of R_i (Fig. 2.1). In the case of a polyhedral graph, where all of its faces can be considered as interior ones, we can choose any face as the infinite face. For graphs with multiple planar embeddings, outer cycles are made from the infinite face of the planar straight line drawing.

Suppose that at the beginning of stage i, one or two current paths are guarded by the cops so as to prevent the robber from leaving R_i. These paths will be denoted by P_i^α and P_i^β. Since P_i^α and P_i^β are assigned at the end of R_{i-1}, $P_i^\alpha \cap R_i = \emptyset$ and $P_i^\beta \cap R_i = \emptyset$. We assume that P_i^α always exists, i.e., R_i ($i > 0$) can NOT be assigned without P_i^α. The newly introduced path at stage i will be denoted by P.



(a) P_i^α and $C(R_i)$. (b) $G[S]$, $S = V(R_i \cup P_i^\alpha)$. (c) $B(R_i) = C(G[S])$.

Figure 2.2: An example of $B(R_i)$ constructed from outer cycles of subgraph $S = V(R_i \cup P_i^\alpha)$.

For two end vertices of a new guarded path, one may consider to choose them from $C(R_i)$. But, it is difficult or even impossible in some cases for the new path to share only one common vertex with P_i^α or P_i^β. To overcome this difficulty, we will select the end vertices from the outer cycle of the subgraph, which is induced by the vertices of the union of R_i, P_i^α and P_i^β.

Definition 2.2. Let $S = V(R_i \cup P_i^\alpha \cup P_i^\beta)$. The graph $B(R_i)$ (of enlarged outer cycles) is defined as $C(G[S])$.

Note that $B(R_i)$ consists of only cycles, and thus not all vertices of R_i, P_i^α and P_i^β belong to $B(R_i)$. For instance, the vertex a of P_i^α (Fig. 2.2(a)) does not belong to $B(R_i)$ (Fig. 2.2(c)).

2.3 Capture Strategy on a Given Graph

This section focuses on how to construct the guarded paths at each stage of our strategy. The analysis on the capture time of the strategy will be given in Section 7. We first give two propositions, which are used throughout our capture strategy.

Proposition 1. At the end of each stage i, we have at least one free cop.

Proposition 2. During our strategy, any guarded path introduced at stage i shares only its end vertices with each of the current paths.

Before we begin, keep in mind that at some stage the robber territory may become a tree; in this case, only one cop suffices (Theorem 1.1). The main idea of our strategy is to let at most two cops guard two different paths, and then employ a free cop to guard the new path. This makes one of the current guarded paths obsolete and thus reduces the robber territory.

Our capture strategy consists of two phases.

1. Initial Phase: We first find a location to place the cops, based on the structure of the graph G. This phase also constructs the very first guarded path, and thus goes from the start to the end of stage 1. When it is over, R_1 is reduced to R_2.

2. Recursive Phase: At a stage i (≥ 2), we construct a new guarded path using a case-analysis method. At the end of stage i, R_i is reduced to R_{i+1}. We do this recursively until R_i becomes the tree case.

2.3.1 Initial Phase

The initial phase has the following two objectives: (i) find a vertex to place the cops at stage 0, and (ii) establish the first guarded path at stage 1 or in R_1.

At stage 0, if $B(R_0)$ is empty, then we know that the graph is a tree, which can be easily dealt with as stated in Lemma 1. In the case that $B(R_0)$ is not empty, we choose a vertex $v_0 \in B(R_0)$ such that v_0 is not a cut vertex (for the simplicity of assigning R_1). We place all cops c_1, c_2 and c_3 at v_0, and then wait for the robber player to place his piece r on the graph.

Suppose r is now located at some vertex of $V(R_0) - \{v_0\}$. At stage 1, we have $R_1 = G[V(R_0) - \{v_0\}]$ and $B(R_1) = B(R_0)$ (as P_1^1 is the vertex v_0 and $P_1^2 = \emptyset$). Note that $v_0 \in B(R_1)$ is on some cycle C_1 of $B(R_1)$. Let v be the vertex of C_1, which is farthest to v_0 on C_1. We find a shortest path $P = \pi(v_0, v)$ in $G[V(R_1) \cup \{v_0\}]$ and send a cop, say, c_1 to guard P. In the case that there are multiple shortest paths for the pair of end vertices, an arbitrary (shortest) path can be used here. We let c_2 guard the vertex v_0, and thus c_3 is free at v_0 until stage 1 ends.

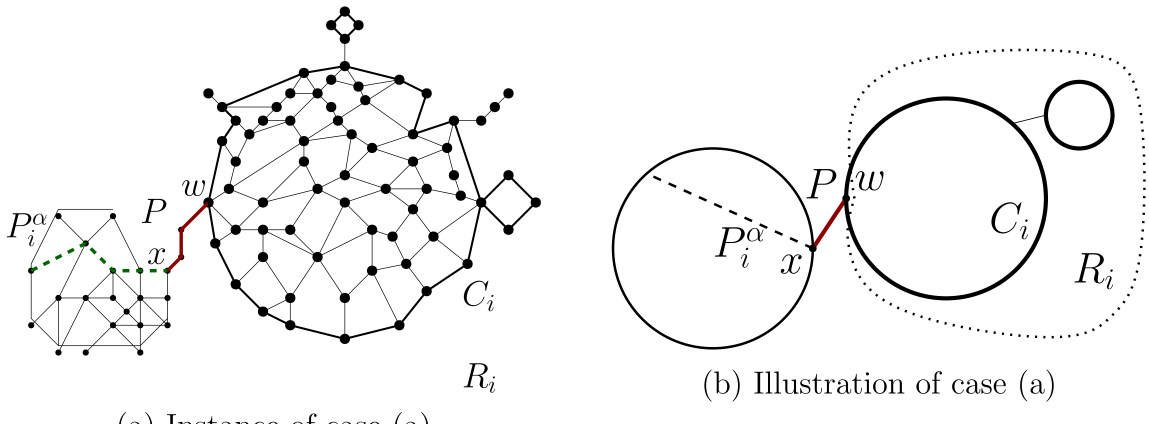
At the end of the initial phase, if P separates R_1 into two or more components, R_2 is then the connected component containing the robber r. Otherwise, $R_2 = G[V(R_1) - V(P)]$. The reduced robber territory R_2 is either a subgraph with at least one outer cycle, or a tree. If it is the tree case, the robber can simply be captured (Theorem 1.1). Otherwise, we enter Recursive Phase.

2.3.2 Recursive Phase

In this phase, we recursively reduce the robber territory R_i into R_{i+1}, until R_i becomes a tree. The reduction of R_i into R_{i+1} is done by constructing and controlling a new guarded path (Observation 1.10).

Recall that the current paths P_i^α and P_i^β were given at the end of stage $i-1$, and R_i can never be assigned without P_i^α. We distinguish the following situations.

Case (a): $B(R_i) \cap P_i^\alpha = \emptyset$ *and* $P_i^\beta = \emptyset$.



(a) Instance of case (a)

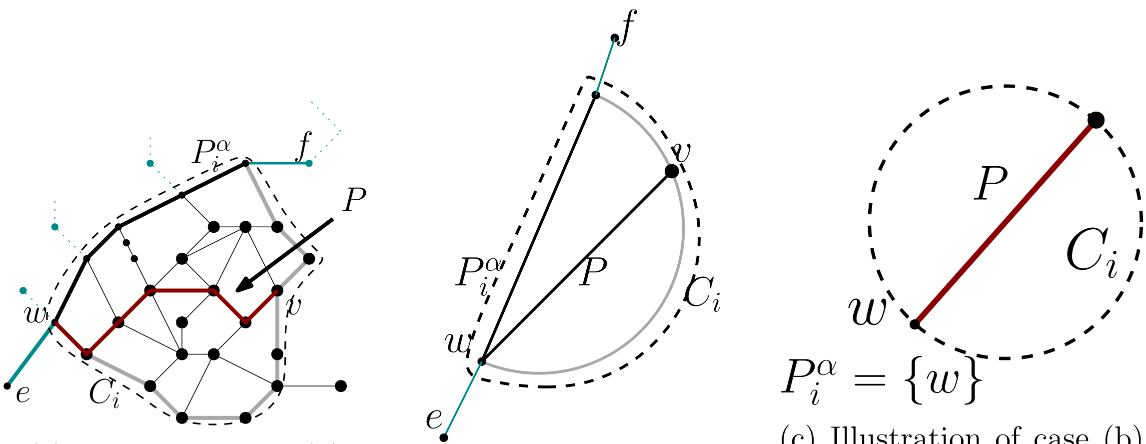
(b) Illustration of case (a)

Figure 2.3: An example of case (a).

Case (a) occurs only when P_i^α contains a cut vertex and the robber r is on the other component separated by that cut vertex (Fig. 2.3(a)). We find a vertex $x \in P_i^\alpha$ and a vertex $w \in B(R_i)$ such that $|\pi(x, w)|$

is the minimum among the shortest paths between a vertex of P_i^α and a vertex of $B(R_i)$. Since $B(R_i) \cap P_i^\alpha = \emptyset$, the pair (x, w) is unique, and all vertices of $\pi(x, w)$ are cut ones, see Fig. 2.3(a). We designate $\pi(x, w)$ as P, and move all the cops to x, then along P to w. Since P consists of only cut vertices, guarding the end vertex w suffices. When all cops occupy w, the stage ends and $R_{i+1} = G[V(R_i) - V(P)]$.

Case (b): *$B(R_i) \cap P_i^\alpha \neq \emptyset$ and $P_i^\beta = \emptyset$.*



(a) Instance of case (b)

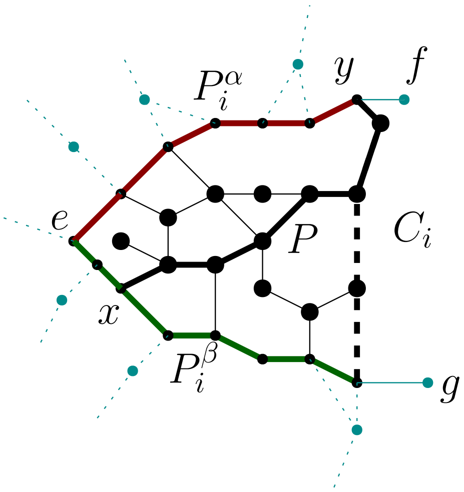
(b) Illustration of case (b) when P_i^α is not a single vertex

(c) Illustration of case (b) when P_i^α is a single vertex

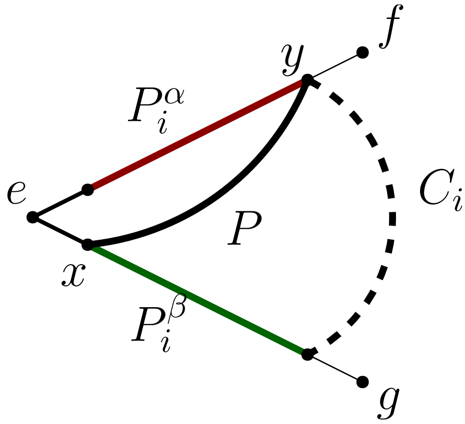
Figure 2.4: An example of case (b).

Figure 2.4(a) gives an example for this case. Let w be an end vertex of $B(R_i) \cap P_i^\alpha$, and $C_i \subseteq B(R_i)$ be the cycle containing w. Since one cop, say, c_1 can guard P_i^α (including w), two cops are free in this case. Note that w has some neighbors in R_i. Let v be the vertex of C_i, which belongs to R_i and farthest to w on C_i. We then find a shortest path $P = \pi(w, v)$ in $G[V(R_i) \cup \{w\}]$, and send a free cop, say, c_2 to guard P (another free cop is put on standby at w). In $G[V(R_i) - V(P)]$, the component containing r is then R_{i+1}. There are two possible situations in this case; when P_i^α is not a single vertex (Fig. 2.4(b)), or otherwise (Fig. 2.4(c)). Note that in the former case, one region needs to be guarded by two paths, P_i^α and P; if that region becomes R_{i+1}, then stage $i + 1$ falls into the next case.

Case (c): $B(R_i) \cap P_i^\alpha \neq \emptyset$ *and* $B(R_i) \cap P_i^\beta \neq \emptyset$.



(a) Instance of case (c)



(b) Illustration of case (c)

Figure 2.5: An example of case (c).

Note that if $P_i^\beta \neq \emptyset$, then $B(R_i) \cap P_i^\beta \neq \emptyset$ because some portions of P_i^α, P_i^β and R_i form a cycle in $B(R_i)$. In this case, two cops have to guard P_i^α and P_i^β, and thus we have only one free cop, say, c_3. It can be deduced from Proposition 2 that P_i^α and P_i^β have a common vertex, say, e. Instead of the whole paths P_i^α and P_i^β, we use only some portions of P_i^α and P_i^β starting from e. Let f (g) be the other end vertex of P_i^α (P_i^β). By the denotations, $P_i^\alpha = \pi(e, f)$ and $P_i^\beta = \pi(e, g)$. Note that f or g may not be on any outer cycle of $G[V(R_i \cup P_i^\alpha \cup P_i^\beta)]$, and thus it may not belong to $B(R_i)$.

We then find two vertices $x \in N(R_i) \cap P_i^\beta$ and $y \in B(R_i) \cap P_i^\alpha$ such that x and y are the vertices of P_i^β and P_i^α, which are closest to e and f along P_i^β and P_i^α, respectively. See Fig. 2.5 for example. Note that x has to be chosen from $N(R_i)$, instead of $B(R_i)$, because we want it to have some neighbor in R_i. It is also possible for $x = e$ if $e \in N(R_i)$, and for $y = f$ if $f \in B(R_i)$. Finally, we find a shortest path $P = \pi(x, y)$ in $G[V(R_i) \cup \{x\} \cup \{y\}]$, and send the free cop c_3 to guard P. Again, R_{i+1} is the component containing r in $G[V(R_i) - V(P)]$.

2.4 Correctness and Completeness

In this section we show that our capture strategy is correct and complete.

Theorem 2.3. *Proposition 1 is upheld for the whole of capture strategy.*

Proof. In case (b) and the initial phase, only one new path is introduced, and one cop is put on standby. Therefore, at least one cop is free at the next stage. In case (a), only one cop has to guard a vertex w of P, and thus two others are free.

In case (c), as shown in Fig. 2.5(b), the new guarded paths P may partition R_i into two components; each of them is a candidate of R_{i+1}. No matter which component becomes R_{i+1}, either P_i^α or P_i^β becomes obsolete and its cop is free at the next stage. □

Theorem 2.4. *Proposition 2 is upheld for the whole of capture strategy.*

Proof. The path P introduced in the initial phase, case (a), or case (b) always has a common vertex with the current path P_i^α, which is v_0, x, or w, respectively.

In case (c), the newly introduced path $P = \pi(x, y)$ shares y with P_i^α, and x with P_i^β and possibly P_i^α (e.g., when $x{=}e \in P_i^\alpha$). Hence, P shares only its end vertices with each of the current paths. □

Theorem 2.5. *At the end of each stage i, $R_{i+1} \subsetneq R_i$.*

Proof. As shown in Section 5, a new path introduced at a stage i consists of some vertices of R_i, excluding its end vertices. Since stage i ends when newly introduced paths are guarded by the cops, all the vertices of those paths do not belong to R_{i+1} (Observation 1.10). Hence, $R_{i+1} \subsetneq R_i$. □

The completeness of our strategy follows from Theorems 2.3 and 2.4, and the correctness follows from Theorem 2.5.

2.5 Summary

We have given the new strategy for a planar graph using three cops, and proved that it works correctly and completely. The refinitions made to the method will provide a better bound for the capture time on a planar graph, which we will discuss in the next chapter.

References

[1] M. Aigner and M. Fromme. “A game of Cops and Robbers”. In: *Discrete Applied Mathematics* 8.1 (1984), pp. 1–12.

[8] A. Bonato and R. J. Nowakowski. *The Game of Cops and Robbers on Graphs*. Student Mathematical Library 61. Providence, Rhode Island: American Mathematical Society, 2011.

[9] A. Bonato et al. “The capture time of a graph”. In: *Discrete Mathematics* 309 (2009), pp. 5583–5595.

[22] A. Mehrabian. “The capture time of grids”. In: *Discrete Mathematics* 311 (2011), pp. 102–105.

[23] R. J. Nowakowski and P. Winkler. “Vertex-to-vertex pursuit in a graph”. In: *Discrete Mathematics* 43 (1983), pp. 235–239.

[28] P. Pisantechakool and X. Tan. “On the capture time of Cops and Robbers game on a planar graph”. In: *Lecture Notes in Computer Science* 10043 (2016), pp. 3–17. Springer.

[30] A. Quilliot. “Jeux et pointes fixes sur les graphes”. Thèse de 3ème cycle. French. PhD thesis. Université de Paris VI, 1978, pp. 131–145.

Chapter 3

Linear Bound on the Capture Time for a Planar Graph

3.1 Introduction

In this chapter, we prove the linear bound of the capture time on a planar graph. To get a better understanding of the capture time, we will focus on the movements of the cops in our strategy [28]. More specifically, we evaluate the movements of the cops on every path, from its introduction to its end (that is, the path is no longer used again).

Recall that a current path Q at stage i may have some vertices, whose neighbors are not in R_i. We introduce below another concept, called the *active portion* of a current path.

Definition 3.1. [[28], Definition 4]Active Portion: Let $Q = \pi(p, q)$ be a current guarded path at stage i, x (y) the first vertex of Q from p (q) that has a neighbor in R_i. The subpath $P(x, y)$, from x to y, is called an active portion of Q.

Path Q as whole may be an active portion if both of its end vertices are in $N(R_i)$. If Q persists through many stages without becoming obsolete, its active portion may become shorter due to the change of the robber territory. This active portion can be represented as $Q \cap B(R_j)$ $(j > i)$, since Q is still a current path and thus the active portion of Q at stage j belongs to $B(R_j)$.

Lemma 3.2. Suppose that path Q is guarded by some cop c. Then it suffices for c to guard the active portion of Q.

Proof. Let $Q(x, y)$ be the active portion of Q in the current stage i. By Definition 1.8, the robber territory R_i has no vertex adjacent to any vertex on $Q - Q(x, y)$. Suppose the robber r wants to travel to some vertex $u \in Q - Q(x, y)$. Clearly, u cannot be reached in R_i without traversing through Q. So, the robber has to enter $Q(x, y)$ first, and by Lemma 1.7, he will get captured. □

We treat each path in similar fashion to a state machine; each is in a single state at any time and can change state. We discuss in details on how a single path proceeds from one state to the next.

Path's states: Assume that path U is introduced at stage i. Path U has four states in its life as follow.

1. Initializing: When U is first introduced at stage i but not fully guarded by a cop, it is in this state. Initializing state ends when the free cop successfully enforces the condition (*) on U at stage i, or in case (a) of the recursive phase, occupies the cut vertex w. When this state ends, U enters the *active* state.

2. Active: Path U is in this state as long as it has an active portion at stage k $(k \geq i+1)$. Path U enters the *deactivating* state when U does not belong to $B(R_k)$ or the active portion of U degenerates into vertex, which is also an end vertex of some active path that is introduced after U.

3. Deactivating: Path U is in this state while the cop who was guarding it is still on U. When the cop finally leaves U, this state ends and U becomes *inactive.*

4. Inactive: When path U reaches this state, it will never be traversed by the cops again, and is no longer needed in our strategy.

Initializing state lasts for only one stage. So, if a path U is introduced at stage i, it enters the active state at the beginning of stage $i+1$. Active state may last through several stages of the strategy. In some situations, active state may end and proceed to deactivating state instantly. For example, when a path U is introduced at stage i but U has a cut vertex x and the robber r is on the other component separated by that cut vertex (stage $i+1$ falls into case (a)), our strategy requires all the cops to move to some cut vertex w of $B(R_i)$, and thus puts U in the deactivating state.

Note that in earlier chapters we used *obsolete* paths to describe both deactivating and inactive states. But since we need to be more precise in evaluating the capture time, we now use two states to describe an obsolete path. The transition from deactivating to inactive state occurs in the same stage. That is, if U enters deactivating state at the beginning of stage $j > i$, then it enters inactive state at some round before the end of stage j.

3.2 Evaluation of the Capture Time

Instead of counting the number of movements taken by a free cop, we will give the bound on the number of movements taken by the free cops on each individual path. This makes it possible to not care about which cop has the largest number of movements at a stage.

3.2.1 Different types of movements on shortest paths

In order to understand which movements affect the capture time and which do not, we distinguish the following types of movements taken by the cops.

Types of Movements: the guarding action of a cop requires three types of movements; (i) moving into a guarded path, (ii) moving along the path to satisfy (*), and (iii) moving along the path while keeping to preserve (*). The length of a stage is mainly determined by the action of the free cop trying to control a new path. As soon as a free cop successfully controls the path by enforcing (*), that stage is over. Hence, movements (iii) can be ignored.

We will count movements (i) and (ii) of the free cops, whose movements determine the stages' lengths. We focus on how a path is traversed by the free cops, from its introduction until it is no longer used by any cop.

3.2.2 A life of a used shortest path

We will consider all movements on a path U from its introduction at a stage i ($i \geq 1$) until it enters inactive state at some later stage $j > i$. Generally, U is traversed in at most three stages (in its life): (1) by the cop designated to guard U at stage i, (2) by some other cop at stage $i+1$, who becomes free due to the introduction of U at stage i, and is required to move into her newly designated path at stage $i+1$, by passing through some portion of U, and (3) by the same cop at stage j as at stage i, who moves into her newly designated path. It is clear that the number of movements (ii) made by the cop at stage i is at most $|U|$. From now on, we focus on movements (i) on U, or to be exact, the movements of free cops on U at stage $i+1$ and j.

3.2.3 Summation of all movements on a shortest path

Let us first consider the special situation where $j = i+1$. This occurs when the path U is introduced at stage i which falls into case (a), see Fig. 3.1. In this situation, Path $U = \pi(x, w)$ (vertex w belongs to $B(R_i)$) consists of only cut vertices and stage i ends when all three cop occupy w. Thus, at the end of stage i, all the cops are at the end vertex w of U. Hence, movements (i) are not needed to be made on U in this situation.

For all other situations ($j \geq i+2$), let $U = \pi(p, q)$ be a path introduced at stage i ($i \geq 1$), and U has an active portion at stage $i+1$. Also, let W be the deactivating path when U becomes active at

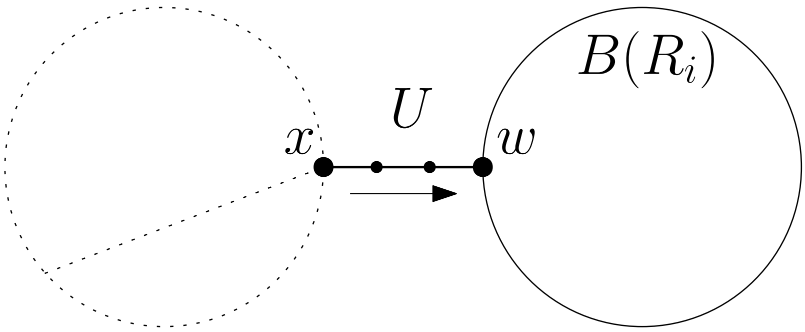


Figure 3.1: An instance in which a path U become deactivated at stage $i+1$.

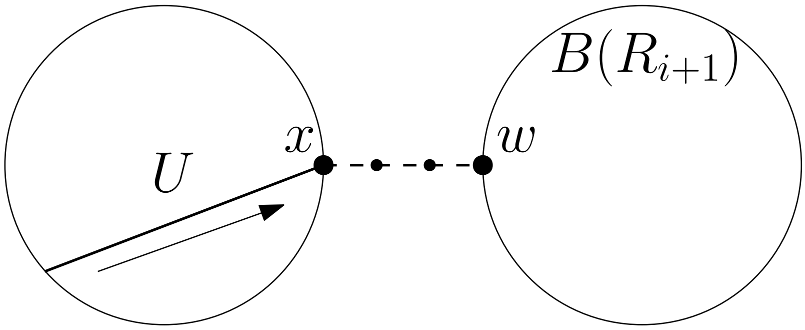
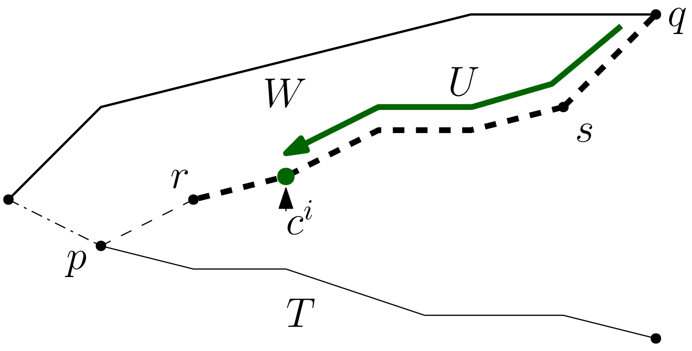


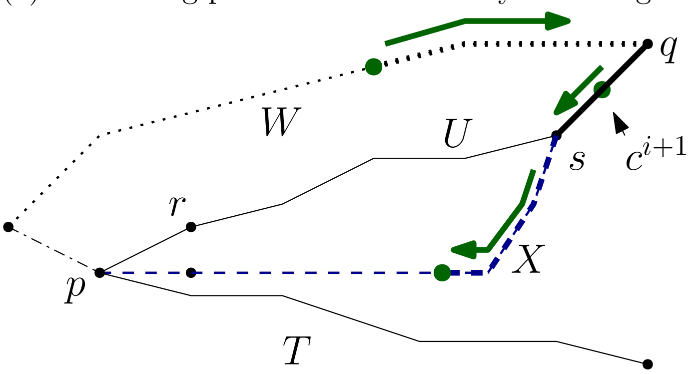
Figure 3.2: An instance in which stage $i+1$ falls into case (a).

stage $i+1$. Note that W and U share a common vertex p or q, and that the cop guarding W is free and required to guard a newly introduced path, say, X, at stage $i+1$. We discuss below how the portion of U is traversed by the free cop at stage $i+1$.

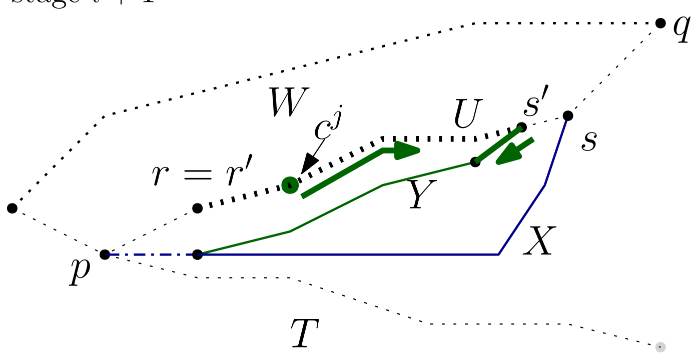
If stage $i+1$ falls into case (a), then we have vertices $x \in U$ and $w \in B(R_{i+1})$, such that $\pi(x, w)$ is minimum. See Fig. 3.2 for an example. The active portion of U at stage $i+1$ is then a cut vertex x. All the cops are required to move into w as described in Section 5.2, which is done by first moving into U, along U to x, then along $\pi(x, w)$ to w. We only need to count the number of movements of the cop who is farthest from x on U, which clearly is at most $|U|$. Hence, U is traversed by the free cops for at most $|U|$ movements (i). Note that in this situation, U directly enters inactive state at the end of stage $i+1$, and thus the evaluation of movements (i) is completed.



(a) Initializing path U is traversed by c^i at stage i



(b) Active path U is traversed by c^{i+1} ($\neq c^i$) at stage $i+1$



(c) Deactivating path U is traversed by c^j ($= c^i$) at stage j

Figure 3.3: A single path U is traversed by free cops on three separate occasions.

In the case that stage $i+1$ falls into case (b) or (c), assume that r and s are the end vertices of the active portion of U at stage $i+1$. Let w be the common vertex of X and U, and c^{i+1} the cop who guarded W at stage i and becomes free at stage $i+1$. Thus, w is either r or s. The free cop c^{i+1} is either somewhere on W or on standby at p at the end of stage i, before she moves to guard X. In the former case, we assume c^{i+1} does so by moving along W to p or q (the common vertex of W and U), then along U to w. In the latter, we assume c^{i+1} does so by moving along U to w. In either case, the number of movements (i) on U at stage $i+1$ is at most $|U(p,r)|$ or $|U(s,q)|$ (Fig. 3.3(b)).

Finally, we evaluate the number of movements (i) on U at stage j. Let Y be the path introduced at stage $j-1$, and t the common vertex of Y and U. That is, the introduction of path Y makes u enter deactivating state at stage j. Let r' and s' be the end vertices of the active portion of U at stage $j-1$, and c^j the cop who guarded $U(r',s')$ at stage $j-1$ and becomes free at stage j. Thus, t is either r' or s'. If $j = i+2$, then $Y = X$ and $U(r',s') = U(r,s)$. Otherwise $U(r',s') \subseteq U(r,s)$.

At stage j, the cop c^j is designated to guard a new path. We assume she does so by moving along $U(r',s')$ to t, then along Y to the common vertex between Y and the new path. Hence, $U(r',s')$ is traversed by c^j for at most $|U(r',s')|$ movements (i) to move out of U at stage j (Fig. 3.3(c)). Note that $|U(r',s')| + |U(p,r)| \leq |U|$ and $|U(r',s')| + |U(s,q)| \leq |U|$.

In summary, the total number of the movements (i) on U, done by the free cops, is no more than $|U|$, either. Hence, we have the following results.

Lemma 3.3. In our capture strategy, each guarded path is traversed by the free cops, whose movements determine the stages' lengths, no more than twice of its own length.

Theorem 3.4. *In our capture strategy, all the paths used in evaluating the lengths of stages are distinct, excluding their end vertices.*

Proof. Supposed the guarded paths P_i and P_{i+1} are introduced during stage i and stage $i+1$, respectively. It follows from Theorem 2.4 that, excluding their end vertices, P_i is distinct from P_{i+1}. (For completeness, in a tree case, the length of the chase on a tree is simply bounded above by the diameter of the tree.) Hence, the theorem follows. □

Theorem 3.5. *For Cops and Robbers game on a planar graph G of n vertices with three cops, $capt_3(G) \leq 2n$.*

Proof. The theorem directly follows from Theorem 3.4 and Lemma 3.3. □

3.3 Summary

We have presented a new capture strategy for Cops and Robbers game on a planar graph with three cops, and shown that the capture time of our strategy is no more than $2n$. This gives the first linear result on $capt_3(G)$.

References

[1] M. Aigner and M. Fromme. "A game of Cops and Robbers". In: *Discrete Applied Mathematics* 8.1 (1984), pp. 1–12.

[8] A. Bonato and R. J. Nowakowski. *The Game of Cops and Robbers on Graphs*. Student Mathematical Library 61. Providence, Rhode Island: American Mathematical Society, 2011.

[9] A. Bonato et al. "The capture time of a graph". In: *Discrete Mathematics* 309 (2009), pp. 5583–5595.

[22] A. Mehrabian. "The capture time of grids". In: *Discrete Mathematics* 311 (2011), pp. 102–105.

[23] R. J. Nowakowski and P. Winkler. "Vertex-to-vertex pursuit in a graph". In: *Discrete Mathematics* 43 (1983), pp. 235–239.

[28] P. Pisantechakool and X. Tan. "On the capture time of Cops and Robbers game on a planar graph". In: *Lecture Notes in Computer Science* 10043 (2016), pp. 3–17. Springer.

[30] A. Quilliot. “Jeux et pointes fixes sur les graphes”. Thèse de 3ème cycle. French. PhD thesis. Université de Paris VI, 1978, pp. 131–145.

Chapter 4

On the Smallest 3-Cop-Win Planar Graph

4.1 Introduction

In this chapter, the smallest 3-cop-win planar graph is studied. Some useful insights and one new property of 2-cop-win graphs are given. We conjecture that the dodecahedral graph of order 20 is the smallest *planar* graph, whose cop number is three. Although a proof has not yet been given, we make some progress on it.

First, we show our conjecture in the following.

Conjecture 4.1. The dodecahedral graph is the smallest 3-cop-win planar graph.

In other words, any graph of order at most 19 has a winning strategy using two cops. Such a strategy requires two important parts: (1) a goal, and (2) a method for the cop player to move each piece to its individual goal [29]. We name the goal as a *winning vertex*.

In an attempt to prove the conjecture, we provide the following evidences: (1) any planar graph of order at most 19 has a winning vertex at which the robber is captured by two cops, and (2) a special planar graph of order 19 that is constructed from the dodecahedral graph is 2-cop-win.

4.2 The 2-Cop-Winning Vertices on Planar Graphs

Let us consider the robber's final location on G before being captured by two cops. If the robber does not surrender, then the cops must *trap* him by restricting his movement just before he is captured [5]. The trapping condition using k cops against the robber r requires that $\bar{N}(r) \subseteq \bar{N}(C)$, $N(C) = \{c_1, c_2, \ldots, c_k\}$.

Definition 4.2. For a vertex v, the vertices c_1, c_2 are called the winning positions against v if $\bar{N}(v) \subseteq \bar{N}(c_1) \cup \bar{N}(c_2)$. If the winning positions c_1, c_2 against v exist, then v is called a *2-cop-winning* vertex.

A 2-cop-winning vertex and the winning positions against it are important parts of any capture strategy using two cops. That is, to successfully capture the robber, two cops must force him to be at a 2-cop-winning vertex, and at the same time, the cops occupy the winning positions against that vertex. If a 2-cop-winning vertex does not exist, then it is impossible to capture the robber with only two cops.

In this chapter, we make an important progress on the conjecture that dodecahedral is the smallest 3-cop-win planar graph. We show that any planar graph of order at most 19 has a 2-cop-winning vertex.

Lemma 4.3. [32] For any planar graph G, $\delta(G) \leq 5$.

Since $\delta(G) \leq 5$, we focus on the existences of 2-cop-winning vertices whose degrees are at most five. The 2-cop-winning vertices v and their winning positions c_1 and c_2 when $d(v) = 3$, 4, and 5 are shown in Fig. 4.1. It can be seen that when $d(v) = 3$, if v belongs to a 3- or a 4-cycle, then there exist winning positions a and b against v. In the case that $d(v) = 4$, if v is a common vertex of two 3-cycles, or a 3-cycle and a 4-cycle which do not have a common edge, then there exist winning

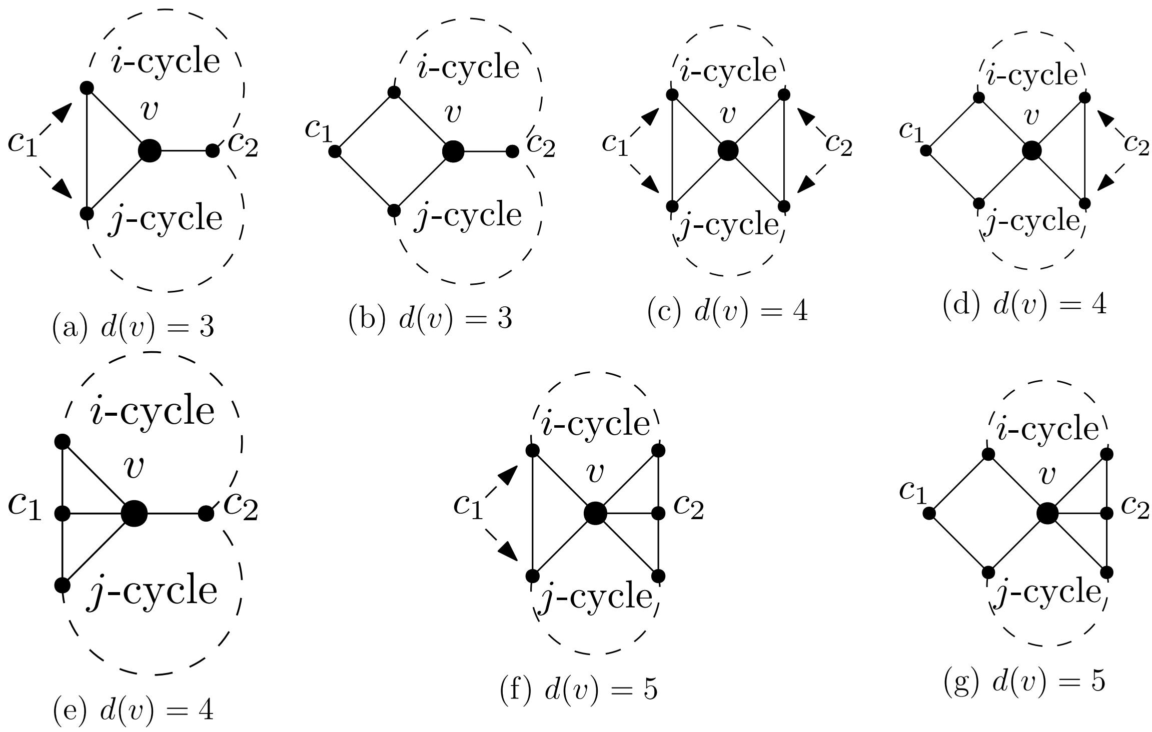


Figure 4.1: All possible instances of the 2-cop-winning vertex v ($d(v) = 3$, 4, and 5, resp.) in which $\bar{N}(v) \subseteq \bar{N}(c_1) \cup \bar{N}(c_2)$, and the cops' winning positions c_1 and c_2. Note that $i \geq 3$ or/and $j \geq 3$.

positions against v. In the case that $d(v) = 5$, if v is a common vertex of two 3-cycles that share a common edge and one 3- or 4-cycle which does not share any common edge with those two 3-cycles, then the winning positions against v exist.

In the next section, we prove that a planar graph of order at most 19 has a 2-cop-winning vertex v. We separate the graphs into different cases, based on the minimum degree of the graphs (so as to eliminate the existence of vertices of lower degrees).

4.3 Winning Vertices for Planar Graphs of Order At Most 19

Our proof consists of a series of lemmas, mainly on the relationships between $\delta(G)$ and the number of cycles of length at least four or five. For a graph G, denote by $S(F(G))$ the summation of sides of all faces in $F(G)$.

First, we show that for a planar graph G of order at most 19, there exists at least a 3- or 4-cycle.

Lemma 4.4. Suppose G is a planar graph of order at most 19. If $\delta(G) = 3$, then $g(G) \leq 4$.

Proof. It follows from Euler's Formula that $|V(G)| - |E(G)| + |F(G)| = 2$. Note also that $|V(G)| \leq 19$. Since $\delta(G) = 3$, we have this inequality:

$$|E(G)| \geq \lceil \frac{3|V(G)|}{2} \rceil \tag{4.1}$$

Since each edge is shared by two faces, we have $S(F(G)) = 2|E(G)|$. Suppose (by contradiction) that $g(G) \geq 5$. Then, we have the second inequality: $2|E(G))| \geq 5|F(G)|$. We derive from Euler's Formula that $|F(G)| = |E(G)| - |V(G)| + 2$. Substitute $|F(G)|$ with $|E(G)| - |V(G)| + 2$ in the inequality:

$$
\begin{aligned}
2|E(G)| &\geq 5|E(G)| - 5|V(G)| + 10 \\
5|V(G)| + 2|E(G)| - 10 &\geq 5|E(G)| \\
5|V(G)| - 10 &\geq 3|E(G)| \\
|E(G)| &\leq \lfloor \frac{5|V(G)| - 10}{3} \rfloor
\end{aligned}
\tag{4.2}
$$

However, these two inequalities (4.1) and (4.2) contradict each other for $|V(G)| \leq 19$. Hence, the lemma follows. □

Let us consider a planar graph G of order at most 19 when $\delta(G) = 5$. From the implication of Euler's Formula, the number of cycles whose length are at least four is smaller than when $\delta(G) = 3$. We obtain the exact numbers of such cycles in the following lemma.

Lemma 4.5. Suppose G is a planar graph of order at most 19. If $\delta(G) = 5$, then: (i) G has at most one 6-cycle and all the remaining faces are 3-cycles, (ii) G has at most one 4-cycle and one 5-cycle, or (iii) G has at most three 4-cycles.

Proof. Similar to the proof of Lemma 4.4, for G with $\delta(G) = 5$, we have this condition:

$$
|E(G)| \geq \lceil \frac{5|V(G)|}{2} \rceil \tag{4.3}
$$

For statement (i), suppose by contradiction that there is a cycle of length at least seven, or there exist a 6-cycle and a cycle of length at least four in G. Since all the remaining faces are 3-cycles, we have the following inequalities: $S(F((G)))| \geq 7+3(|F(G)|-1) = 3|F(G)|+4$ for the former and $S(F((G)))| \geq (6+4)+3(|F(G)|-2) = 3|F(G)|+4$ for the latter, both of which yield the same inequality. Thus, we simplify them into $2|E(G)| \geq 3|F(G)| + 4$. Similar to the proof of Lemma 4.4, we substitute $|F(G)|$ with $|E(G)| - |V(G)| + 2$ in the inequality:

$$
\begin{aligned}
2|E(G)| &\geq 3|E(G)| - 3|V(G)| + 6 + 4 \\
3|V(G)| + 2|E(G)| - 10 &\geq 3|E(G)| \\
3|V(G)| - 10 &\geq |E(G)| \\
|E(G)| &\leq 3|V(G)| - 10 \qquad (4.4)
\end{aligned}
$$

However, these two inequalities (4.3) and (4.4) contradict each other for any $|V(G)| \leq 19$. Hence, for any planar graph G, $|V(G)| \leq 19$, there exists at most one 6-cycle, and all the remaining faces of G are 3-cycles.

For statement (ii), suppose by contradiction that there exist two 5-cycles, or one 5-cycle and two 4-cycles, in G. Since all the remaining faces are 3-cycles, we have the following inequalities: $S(F((G)))| \geq (5 \times 2) + 3(|F(G)| - 2) = 3|F(G)| + 4$ for the former and $S(F((G)))| \geq (4 \times 2 + 5) + 3(|F(G)| - 3) = 3|F(G)| + 4$ for the latter, both of which yield the same inequality. Thus, we simplify them into $2|E(G)| \geq 3|F(G)| + 4$. Since this inequality is the same as (4.4) in the proof of statement (i), it contradicts (4.3) for any $|V(G)| \leq 19$.

Similarly, for statement (iii), suppose by contradiction that there exist four 4-cycles in G. We have the following condition: (2) $S(F((G)))| \geq (4 \times 4) + 3(|F(G)| - 4) = 3|F(G)| + 4$, which is the same as (4.4). The proof can thus be omitted. □

Lemma 4.6. Suppose that G is a planar graph of order at most 19, with $\delta(G) = 5$. Then, there exists a vertex v of degree five that does not belong to a 6-cycle, and is not common to a 4-cycle and a 5-cycle, nor to three 4-cycles.

Proof. First, recall the maximum number of edges lemma derived from Euler's formula that, for any planar graph G of order at least three, $|E(G)| \leq 3|V(G)| - 6$ [32]. From Lemma 4.5, we distinguish the following three situations.

Case 1. there exists a 6-cycle in G. So, there are six vertices $v_1, v_2, \ldots, v_6$ that belong to the 6-cycle. If there is the other vertex (than $v_1, v_2, \ldots, v_6$) of degree five, then the lemma is true (Lemma

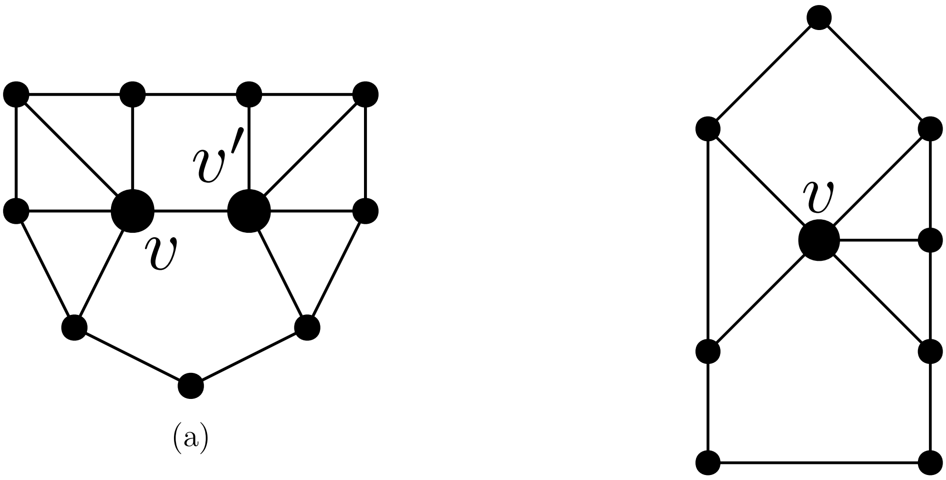


Figure 4.2: Illustration of Case (2) in Lemma 4.6

4.5 statement (i)). Otherwise, all other vertices are of degree at least six. Since $\delta(G) = 5$, at least one of the six vertices $(v_1, v_2, \ldots, v_6)$ must be of degree five. Thus, we have $\sum d(v) \geq (5 \times 6) + 6(|V(G)| - 6) = 6|V(G)| - 6$, for all $v \in V(G)$. Since $|E(G)| = \sum \frac{d(v)}{2}$, we have: $|E(G)| \geq \frac{6|V(G)|-6}{2} = 3|V(G)| - 3$, contradicting $|E(G)| \leq 3|V(G)| - 6$. Thus, there is a vertex of degree five that does not belong to the 6-cycle.

Case 2. there are at most two vertices v and v' that are common to a 4-cycle and a 5-cycle in G (see Fig. 4.2). If there is the other vertex (than v and v') of degree five, then the lemma is true (Lemma 4.5 statement (ii)). Otherwise, all other vertices are of degree at least six. Similar to the proof of Case 1, we have $\sum d(v) \geq (5 \times 2) + 6(|V(G)| - 2) = 6|V(G)| - 2$. Since $|E(G)| = \sum d(v)/2$, we have: $|E(G)| \geq \frac{6|V(G)|-2}{2} = 3|V(G)| - 1$, contradicting $|E(G)| \leq 3|V(G)| - 6$. Thus, there must be other vertex of degree five in G.

Case 3. there exists a vertex v that is common to three 4-cycles. If there is another vertex u of degree five, then the lemma is true (Lemma 4.5 statement (iii)). Otherwise, all other vertices are of degree at least six. Similar to the proof of Case 1, a contradiction occurs. Thus, there must be another vertex u of degree five in G. $\square$

Next, we focus on the planar graph of order at most 19 whose minimum degree is four. For the given graph G with $\delta(G) = 4$, we have the following result.

Lemma 4.7. Suppose that G is a planar graph of order at most 19, with $\delta(G) = 4$. Then, there exists a vertex v of degree four such that v is the common vertex of two 3-cycles.

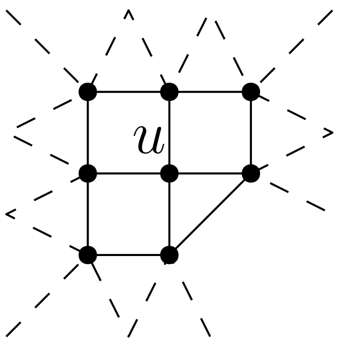
Proof. Suppose by contradiction that there exists no vertex v that belongs two 3-cycles. Since $\delta(G) = 4$, there must be at least a vertex u of degree four in G, and it does not belong to two 3-cycles.

First, we consider the situation in which there is only one vertex u in G. We prove it by first constructing a graph of the *smallest order* such that there is only one u which belongs to at most one 3-cycle, and all other vertices are of degree at least five. Let us start a graph with vertex u of degree four belonging to a 3-cycle and three 4-cycles (so that the *starting graph* is of the smallest order), see Fig. 4.3(a). We then construct a graph such that all the vertices, except for u, are of degree at least five, by adding the minimum number of vertices and edges. From the starting graph of order 8, the sum of missing degrees required to make the vertices (other than u) be of degree five is 17. This construction is shown step-by-step in Fig. 4.3(d), until our desired graph is obtained. But, its order is 21 (this number cannot be further decreased), contradicting $|V(G)| \leq 19$.

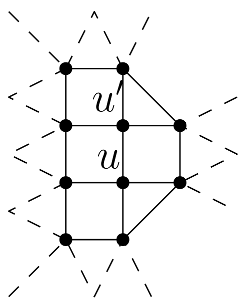
Similarly, for the situation in which there are more than one vertices of degree four, the order of the starting graph is also increased. For the starting graph be of the smallest order, vertices of degree four have to form a connected subgraph (Fig. 4.3(b)), otherwise the order of starting graph is not the smallest (at least 14 for two disjoint vertices of degree four, see Fig. 4.3(c)). In either case, the order of the resulting graph is larger than 19. The proof is complete. □

Next, for the given planar graph G of order at most 19, $\delta(G) = 3$, we show that G has the following property.

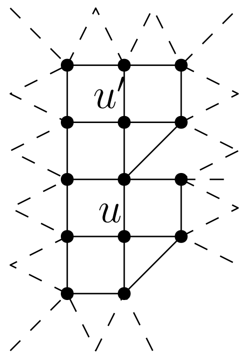
Lemma 4.8. Suppose that G is a planar graph of order at most 19, and $\delta(G) = 3$. In G, either (i) there exists a vertex v of degree three



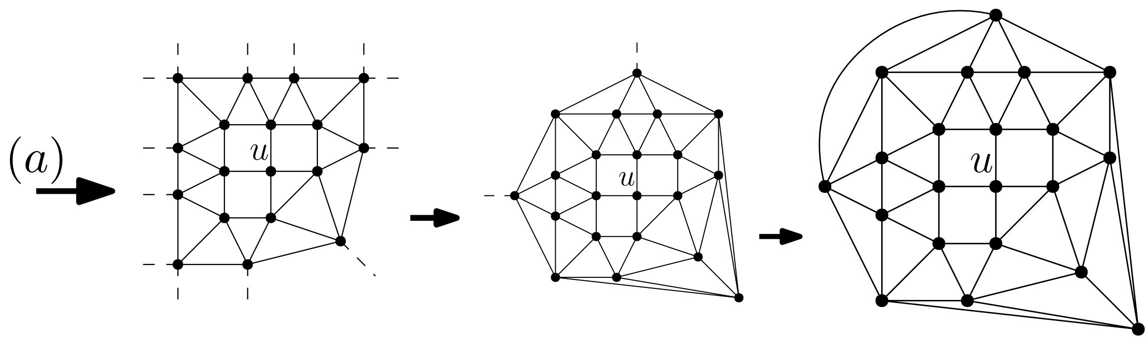
(a) Starting graph with one vertex u of degree four



(b) Starting graph with two adjacent vertices u and u' of degree four

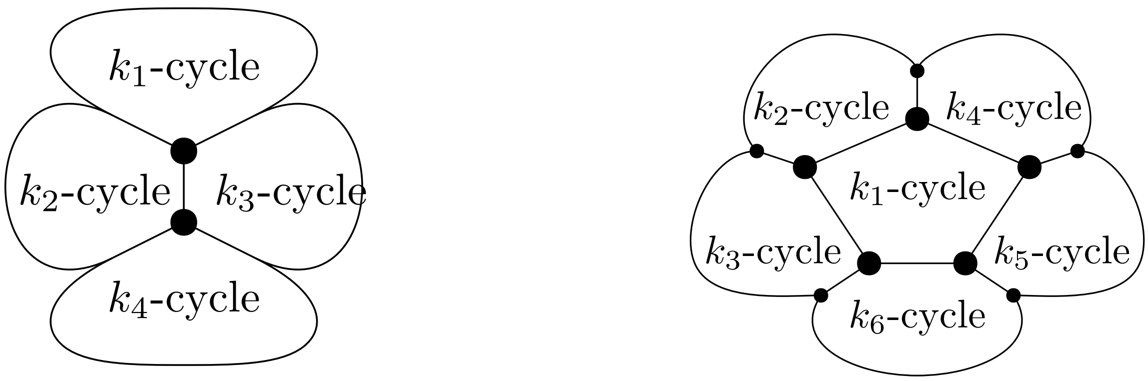


(c) Starting graph with two disjoint vertices u and u' of degree four



(d) The graph constructed from the starting graph shown in (a)

Figure 4.3: Illustration of the proof of Lemma 4.7.



(a) Case (a): the vertices of degree three cannot form a cycle.

(b) Case (b): the vertices of degree three may form a cycle.

Figure 4.4: Illustrations of Cases (a) and (b) in Lemma 4.8.

such that v belongs to a 3- or 4-cycle, or (ii) there exists a vertex v of degree four such that v belongs to two 3-cycles.

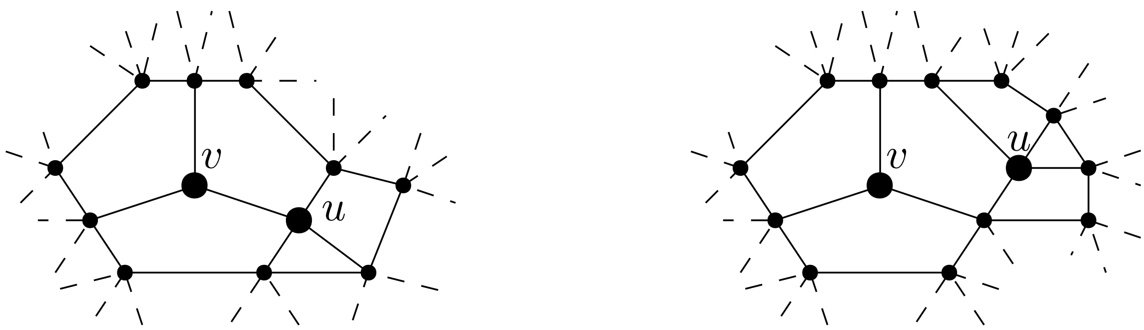
Proof. For any G of order at most 19 with $\delta(G) = 3$, at least one 3- or 4-cycle exists (Lemma 4.4). So, if some vertex of degree three belongs to a 3- or 4-cycle, (i) is true. In the following, we prove that if (i) is false, then (ii) is true.

Suppose by contradiction that (ii) is false, even when (i) is false. Let i (≥ 1) denote the number of vertices of degree three, and j (≥ 0) the number of vertices of degree four.

Consider first the situation in which $j = 0$ (so (ii) is false). Since all other vertices are of degree at least five, we have this inequality: $\sum d(v) \geq (3 \times i) + 5(|V(G)| - i)$. Since $\sum d(v) = 2|E(G)|$, we can rewrite it as follow:

$$|E(G)| \geq \lceil \frac{5|V(G)| - 2i}{2} \rceil \tag{4.5}$$

Next, we consider $S(F(G))$ for this situation. Since the vertices of degree three cannot belong to any cycle of length at most four (otherwise (i) is true), we further distinguish two following cases: (a) when $1 \leq i \leq 4$, the vertices of degree three cannot form a cycle, and thus there exist at least $i+2$ cycles of length at least five (Fig. 4.4(a)), and



(a) Starting graph with vertices v of degree three and u of degree four, u and v are adjacent

(b) Starting graph with vertices v of degree three and u of degree four, u and v are disjoint

Figure 4.5: Illustration of the proof of Lemma 4.8.

(b) when $i \geq 5$, there exist at least $i+1$ cycles of length at least five (Fig. 4.4(b)).

For case (a) ($1 \leq i \leq 4$), by considering all other faces to be cycles of length at least three, we have this inequality: $2|E(G)| \geq 5(i+2) + 3(|F(G)| - (i+2)) = 3|F(G)| + 2i + 4$. Substitute $|F(G)|$ with $|E(G)| - |V(G)| + 2$, we obtain:

$$\begin{aligned} 2|E(G)| &\geq 3|E(G)| - 3|V(G)| + 2i + 10 \\ 3|V(G)| + 2|E(G)| - 2i - 10- &\geq 3|E(G)| \\ 3|V(G)| - 2i - 10 &\geq |E(G)| \\ |E(G)| &\leq 3|V(G)| - 2i - 10 \qquad (4.6) \end{aligned}$$

However, these two inequalities (4.5) and (4.6) contradict each other for $|V(G)| \leq 19$ and $1 \leq i \leq 4$.

Similarly, for case (b) ($i \geq 5$), we have the following inequality: $2|E(G)| \geq 5(i+1)+3(|F(G)|-(i+1)) = 3|F(G)|+2i+2$. From it, we obtain $|E(G)| \leq 3|V(G)| - 2i - 8$, contradicting (4.5) for $|V(G)| \leq 19$ and $i \geq 5$ again. Hence, it is impossible to construct a planar graph G, $\delta(G) = 3$, with $i \geq 1$ vertices of degree three, which do not satisfy (i), and the remaining vertices are of degree at least five. This also implies that if (i) is false, then there exists at least one vertex of degree four.

Let us now consider the situation in which $j \geq 1$. We first construct a graph of the smallest order such that $i = 1$, $j = 1$ and both (i) and

(ii) are false. We start from vertex v of degree three, which is common to three 5-cycles. Then, we assign one neighbor of v to be a vertex u of degree four. Since (ii) is false, u can belong to at most one 3-cycle. This results in the starting graph of order 12, in which v is common to three 5-cycles and u is common to two 5-cycles, a 4-cycle and a 3-cycle, as shown in Fig. 4.5(a). If v and u are disjoint, then the starting graph is not the smallest (see the graph of order 14, shown in Fig. 4.5(b) for an example). From the starting graph of Fig. 4.5(a), we construct a graph such that all the vertices, except for v and u, are of degree at least five, by adding the minimum number of vertices. As discussed in the proof of Lemma 4.7, the resulting graph is of order larger than 19, contradicting $|V(G)| \leq 19$.

In the case that the value of i or j is increased, the order of the starting graph is also increased, and thus so is the order of the resulting graph. Hence, it is impossible to have a planar graph G of order at most 19 with $\delta(G) = 3$, in which both (i) and (ii) are false. The proof is complete. □

Theorem 4.9. *There exists at least one 2-cop-winning vertex in any planar graph of order at most 19.*

Proof. Recall first that for any planar graph G, $\delta(G) \leq 5$ (Lemma 4.3). If $\delta(G) = 1$, any vertex of degree one is clearly a 2-cop-winning vertex. Also, if $\delta(G) = 2$, any vertex v of degree two is a 2-cop-winning vertex, because two cops at two neighbors of v can trap the robber at v. In the following, we consider the three different situations for $\delta(G) = 3$, 4, 5.

For the situation where $\delta(G) = 3$, by Lemma 4.8, there exist either vertices v of degree three belonging to a 3- or 4-cycle, or vertices v of degree four which are common to two 3-cycles. In Fig. 4.6(a) and Fig. 4.6(b) (resp. Figs. 4.6(c)-4.6(d)), we show the winning positions occupied by two cops c_1 and c_2 as well as the 2-cop-winning vertex v of degree three (resp. degree four).

For the situation where $\delta(G) = 4$, by Lemma 4.7, there exists a vertex v of degree four, which is common to two 3-cycles. As shown in Figs. 4.6(c)-4.6(d), vertex v is the winning one.

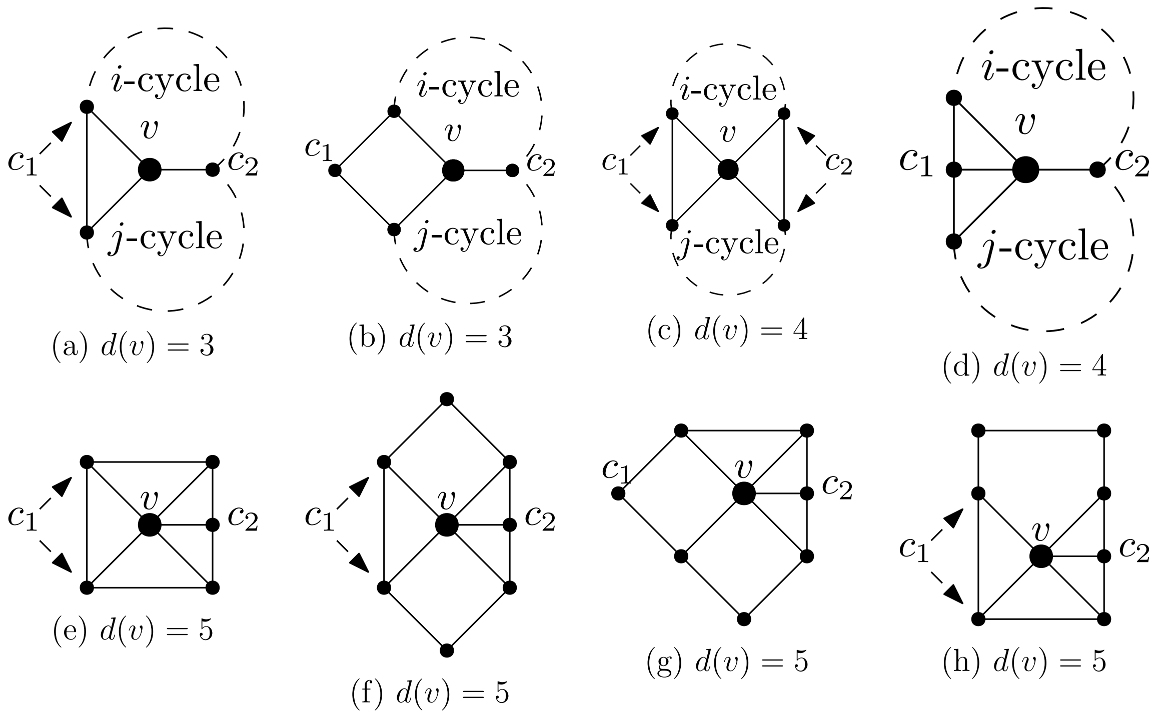


Figure 4.6: Instances of the 2-cop-winning vertex v and the cops' winning positions c_1 and c_2 on planar graph G of order at most 19. Note that for (a) to (d), $i \geq 3$ or/and $j \geq 3$.

Finally, consider the situation in which $\delta(G) = 5$. Let v be a vertex of degree five. From Lemma 4.5, v is common to five 3-cycles, or to two 4-cycles or belongs to a 5-cycle, while the remaining faces are all 3-cycles. The winning positions c_1 and c_2 against v are shown in Fig. 4.6(e), Figs. 4.6(f)-(h) and Fig. 4.6(h), respectively. Again, the 2-cop-winning vertex exists. □

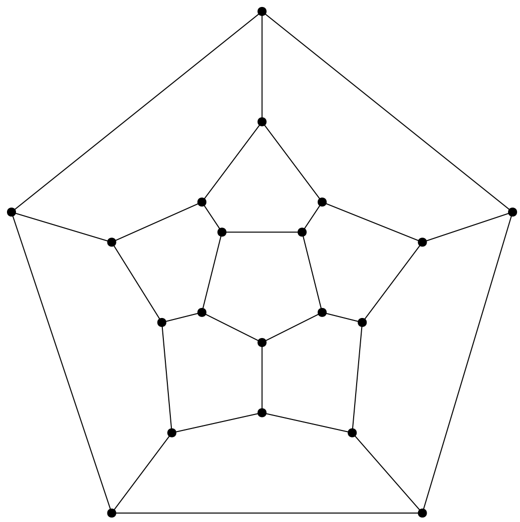
4.4 Winning Strategies for Two Cops on Some Graphs

In this section, we provide some examples of new strategies for two hard cases of planar graphs, whose order at 19 and 16. These two graphs are edge-contracted dodecahedral and 3-regular graph of order 16 (the largest 3-regular planar graph in the graphs of order at most 19). They have no known strategy so far, as it requires more complex strategy than tandem-win graphs.

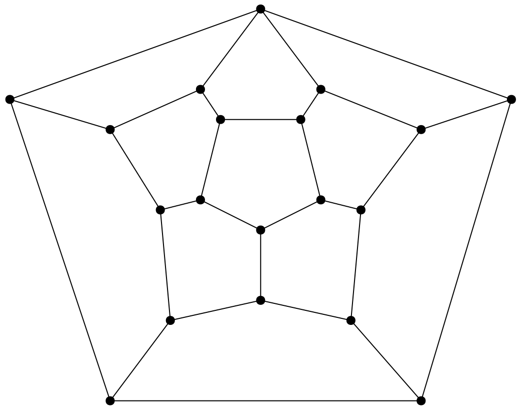
4.4.1 The edge-contracted dodecahedral graph

In a graph G, when *contraction* of an edge $e \in E(G)$ with endpoints u and v is performed on G, the edge e is replaced by a single vertex such that the edges incident to the new vertex are those, other than e, which were incident to u or v (see Fig. 4.7). The contraction of e on G results in a graph with one edge and one vertex fewer than G. This graph operation is called an *edge contraction*. We call the resulting graph the *edge-contracted* version of the original graph, e.g., the edge-contracted dodecahedral graph is the result of performing an edge contraction on the dodecahedral graph.

In this subsection, we show that the edge-contracted dodecahedral graph, which might be considered as the worst case of the planar graphs of order at most 19, has the cop number of two. Studying on the edge-contracted dodecahedral graph may give some insights on the proof of the smallest 3-cop-win planar graph and even on giving a method for determining whether a graph is 2-cop-win or 3-cop-win.



(a) Regular dodecahedral graph.



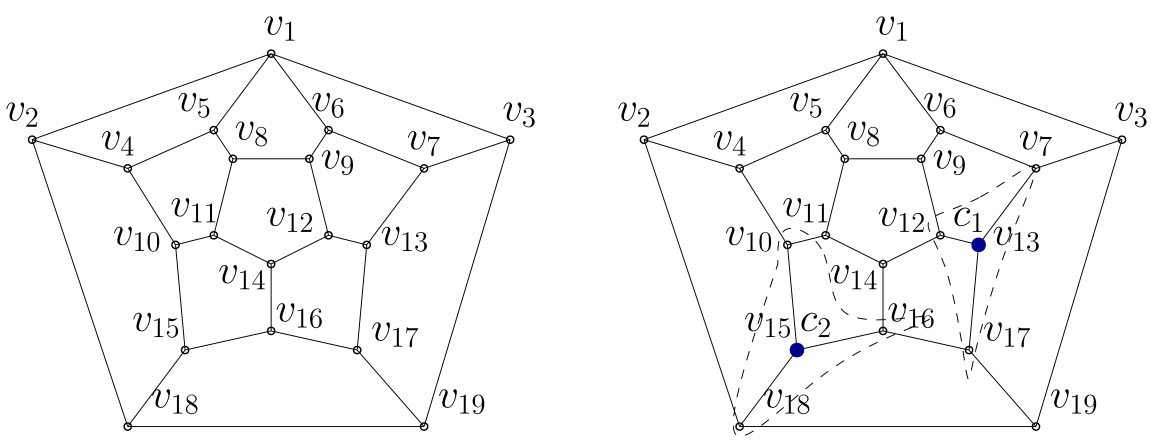
(b) Edge-contracted dodecahedral graph.

Figure 4.7: Regular dodecahedral and its edge-contracted graphs

Note that since the dodecahedral graph is vertex-symmetric, performing an edge contraction on any edge results in the same graph with different embedding. For instance, the graph in Fig. 4.9(b) can be relabeled into the graph in Fig. 4.8(a). For simplicity, we will use the embedding shown in Fig. 4.8(a) to represent the edge-contracted dodecahedral graph.

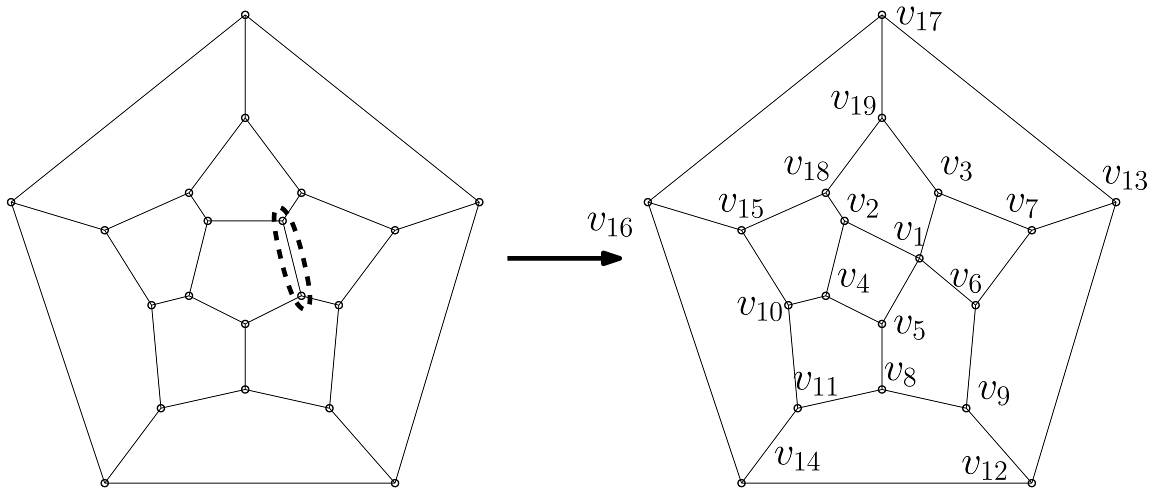
Let the triple (c_1, c_2, v) denote the two winning positions c_1 and c_2 against v. For the instance shown in Fig. 4.8(b), we have the following triples: (v_5, v_{18}, v_2), (v_6, v_{19}, v_3), (v_1, v_{10}, v_4), (v_2, v_8, v_5), (v_3, v_9, v_6), and (v_1, v_{13}, v_7). The main part of a winning strategy is how to enforce the robber into such a vertex v, and at the same time, two cops occupy the winning positions against v. Since the edge-contracted dodecahedral graph is specific, we provide below a full strategy for it.

Initial placement of two cops and the robber. In our strategy, two cops c_1 and c_2 occupy v_{13} and v_{15} in the labeling shown in Fig. 4.8(b). If the robber initially occupies a vertex in $\bar{N}(C) = \bar{N}(v_{13}) \cup \bar{N}(v_{15})$, he will be captured by the cops in the first round. So, the robber can initially occupy one of the following eleven vertices: v_1, v_2, v_3, v_4, v_5, v_6, v_8, v_9, v_{11}, v_{14}, and v_{19}.



(a) (b)

Figure 4.8: The labeling in the edge-contracted dodecahedral graph, and the initial positions of two cops c_1 and c_2 in our strategy.



(a) (b)

Figure 4.9: Another example of edge-contracted dodecahedral graph with same labeling.

In the strategy below, we provide the movements of the cops, as well as the possible positions the robber can move to. We use $c_i(v \to u)$ to denote the movement of c_i from v to u, $i = 1$ or 2, and the robber's movement after the cops' turn (in the same round) is written after a semicolon. Note that our goal is to show the existence of a capture strategy (which may not be optimal). We separate the vertices into three groups; (1) the vertices whose distances to both v_{13} and v_{15} are two, (2) the vertices whose distances to one of v_{13} and v_{15} are two and the other are at least three, and (3) the vertices whose distances to both v_{13} and v_{15} are at least three. We will give the strategy till the robber is trapped, since the robber can then be captured with one more round.

Let us first describe the scenarios in which the robber initially occupies a vertex in group (1). The vertices in group (1) are v_{14} and v_{19}.

Scenario V19: the robber initially occupies v_{19}.

Round 1 $c_1(v_{13} \to v_{13})$, $c_2(v_{15} \to v_{18})$; $r(v_{19} \to v_3)$.
Round 2 $c_1(v_{13} \to v_7)$, $c_2(v_{18} \to v_{18})$; $r(v_3 \to v_1)$.
Round 3 $c_1(v_7 \to v_7)$, $c_2(v_{18} \to v_2)$; $r(v_1 \to v_5)$.
Round 4 $c_1(v_7 \to v_6)$, $c_2(v_2 \to v_4)$; $r(v_5 \to v_8)$.
Round 5 $c_1(v_6 \to v_9)$, $c_2(v_4 \to v_{10})$; $r(v_8 \to v_5)$.
Round 6 $c_1(v_9 \to v_6)$, $c_2(v_{10} \to v_{11})$; (a) $r(v_5 \to v_5)$ or (b) $r(v_5 \to v_4)$.
Case (a): $r(v_5 \to v_5)$ at the end of Round 6.

Round 7 $c_1(v_6 \to v_1)$, $c_2(v_{11} \to v_{11})$; $r(v_5 \to v_4)$.
Round 8 $c_1(v_1 \to v_1)$, $c_2(v_{11} \to v_{10})$; the robber r is trapped. (**Round 9** one cop moves to r.)

Case (b): $r(v_5 \to v_4)$ at the end of Round 6.

Round 7 $c_1(v_6 \to v_1)$, $c_2(v_{11} \to v_{10})$; the robber r is trapped.

Scenario V19 ends within nine rounds (the longest one is Case (a)).

Scenario V14: the robber initially occupies v_{14}.

Round 1 $c_1(v_{13} \to v_{12})$, $c_2(v_{15} \to v_{15})$; $r(v_{14} \to v_{11})$.
Round 2 $c_1(v_{12} \to v_{12})$, $c_2(v_{15} \to v_{10})$; $r(v_{11} \to v_8)$.

Round 3 $c_1(v_{12} \to v_9)$, $c_2(v_{10} \to v_{10})$; $r(v_8 \to v_5)$.
Round 4 $c_1(v_9 \to v_6)$, $c_2(v_{10} \to v_{11})$; the same as the robber's turn at Round 6 in Scenario V19.

Scenario V14 ends within seven rounds (it uses the final three rounds from Scenario V19).

Next, we describe the scenarios in which the robber initially occupies a vertex in group (2). The vertices in group (2) are v_2, v_3, v_4, v_6, v_9, and v_{11}.

Scenario V2: the robber initially occupies v_2.

Round 1 $c_1(v_{13} \to v_7)$, $c_2(v_{15} \to v_{18})$; (a) $r(v_2 \to v_1)$, or (b) $r(v_2 \to v_4)$.
Case (a): $r(v_2 \to v_1)$ at the end of Round 1.

Round 2 the same as Round 3 in Scenario V19.

Case (b): $r(v_2 \to v_4)$ at the end of Round 1.

Round 2 $c_1(v_7 \to v_6)$, $c_2(v_{18} \to v_{15})$; (b.1) $r(v_4 \to v_2)$, (b.2) $r(v_4 \to v_4)$, or (b.3) $r(v_4 \to v_5)$
Case (b.1): $r(v_4 \to v_2)$ at the end of Round 2 in Case (b).
Round 3 $c_1(v_6 \to v_1)$, $c_2(v_{15} \to v_{15})$; $r(v_2 \to v_4)$.
Round 4 $c_1(v_1 \to v_1)$, $c_2(v_{15} \to v_{10})$; the robber is trapped.
Case (b.2): $r(v_4 \to v_4)$ at the end of Round 2 in Case (b).
Round 3 $c_1(v_6 \to v_1)$, $c_2(v_{15} \to v_{10})$; the robber is trapped.
Case (b.3): $r(v_4 \to v_5)$ at the end of Round 2 in Case (b).
Round 3 $c_1(v_6 \to v_1)$, $c_2(v_{15} \to v_{10})$; $r(v_5 \to v_8)$.
Round 4 $c_1(v_1 \to v_6)$, $c_2(v_{10} \to v_{11})$; $r(v_8 \to v_5)$.
Round 5 the same as Round 7 in Case (a) of Scenario V19.

In Scenario V2, the robber is captured within seven rounds (the longest one is Case (b.3), which uses final three rounds from Scenario V19).

Scenario V3: The robber initially occupies v_3.

Round 1 $c_1(v_{13} \to v_7)$, $c_2(v_{15} \to v_{18})$; $r(v_3 \to v_1)$.

Round 2 the same as Round 3 in Scenario V19.

Scenario V3 ends within eight rounds (one fewer round than Scenario V19).

Scenario V4: The robber initially occupies v_4.

Round 1 $c_1(v_{13} \to v_7)$, $c_2(v_{15} \to v_{15})$; (a) $r(v_4 \to v_2)$, (b) $r(v_4 \to v_4)$, or (c) $r(v_4 \to v_5)$.

Case (a): $r(v_4 \to v_2)$ at the end of Round 1.

Round 2 $c_1(v_7 \to v_3)$, $c_2(v_{15} \to v_{18})$; $r(v_2 \to v_4)$.
Round 3 $c_1(v_3 \to v_1)$, $c_2(v_{18} \to v_{15})$; $r(v_4 \to v_4)$.
Round 4 $c_1(v_1 \to v_1)$, $c_2(v_{15} \to v_{10})$; the robber is trapped.

Case (b): $r(v_4 \to v_4)$ at the end of Round 1.

Round 2 $c_1(v_7 \to v_6)$, $c_2(v_{15} \to v_{15})$; the same as the robber's turn at Round 2 in Case (b) of Scenario V2.

Case (c): $r(v_4 \to v_5)$ at the end of Round 1.

Round 2 $c_1(v_7 \to v_6)$, $c_2(v_{15} \to v_{10})$; (c.1) $r(v_5 \to v_5)$, or (c.2) $r(v_5 \to v_8)$.

Case (c.1): $r(v_5 \to v_5)$ at the end of Round 2 in Case (c).

Round 3 $c_1(v_6 \to v_1)$, $c_2(v_{10} \to v_{11})$; $r(v_5 \to v_4)$.
Round 4 $c_1(v_1 \to v_1)$, $c_2(v_{11} \to v_{10})$; the robber is trapped.

Case (c.2): $r(v_5 \to v_8)$ at the end of Round 2 in Case (c).

Round 3 $c_1(v_6 \to v_6)$, $c_2(v_{10} \to v_{11})$; $r(v_8 \to v_5)$.
Round 4 the same as Case (a) at the end of Round 6 in Scenario V19.

Scenario V4 ends in at most seven rounds (the longest one is Case (b)).

Scenario V6: The robber initially occupies v_6.

Round 1 $c_1(v_{13} \to v_{13})$, $c_2(v_{15} \to v_{18})$; (a) $r(v_6 \to v_1)$, (b) $r(v_6 \to v_6)$ or (c) $r(v_6 \to v_9)$.

Case (a): $r(v_6 \to v_1)$ at the end of Round 1.

Round 2 $c_1(v_{13} \to v_7)$, $c_2(v_{18} \to v_2)$; $r(v_1 \to v_5)$.

Round 3 $c_1(v_7 \to v_6)$, $c_2(v_2 \to v_4)$; $r(v_5 \to v_8)$.
Round 4 $c_1(v_6 \to v_9)$, $c_2(v_4 \to v_{10})$; $r(v_8 \to v_5)$.
Round 5 the same as Round 4 in Scenario V14.

Case (b): $r(v_6 \to v_6)$ at the end of Round 1.

Round 2 $c_1(v_{13} \to v_{12})$, $c_2(v_{18} \to v_2)$; (b.1) $r(v_6 \to v_6)$, or (b.2) $r(v_6 \to v_7)$.

Case (b.1): $r(v_6 \to v_6)$ at the end of Round 2 in Case (b).

Round 3 $c_1(v_{12} \to v_{12})$, $c_2(v_2 \to v_1)$; $r(v_6 \to v_7)$.
Round 4 $c_1(v_{12} \to v_{13})$, $c_2(v_1 \to v_1)$; the robber is trapped.

Case (b.2): $r(v_6 \to v_7)$ at the end of Round 2 in Case (b).

Round 3 $c_1(v_{12} \to v_{13})$, $c_2(v_2 \to v_1)$; the robber is trapped.

Case (c): $r(v_6 \to v_9)$ at the end of Round 1.

Round 2 $c_1(v_{13} \to v_{12})$, $c_2(v_{18} \to v_2)$; (c.1) $r(v_9 \to v_6)$, or (c.2) $r(v_9 \to v_8)$.

Case (c.1): $r(v_9 \to v_6)$ at the end of Round 2 in Case (c).

Round 3 the same as Round 3 in Case (b.1) of Scenario V6.

Case (c.2): $r(v_9 \to v_8)$ at the end of Round 2 in Case (c).

Round 3 $c_1(v_{12} \to v_9)$, $c_2(v_2 \to v_4)$; $r(v_8 \to v_{11})$.
Round 4 $c_1(v_9 \to v_{12})$, $c_2(v_4 \to v_{10})$; $r(v_{11} \to v_8)$.
Round 5 the same as Round 3 in Scenario V14.

Scenario V6 ends within eight rounds (the longest one is Case (c.2)).

Scenario V9: The robber initially occupies v_9.

Round 1 $c_1(v_{13} \to v_{12})$, $c_2(v_{15} \to v_{18})$; (a) $r(v_9 \to v_6)$ or (b) $r(v_9 \to v_8)$.

Case (a): $r(v_9 \to v_6)$ at the end of Round 1.

Round 2 $c_1(v_{12} \to v_{12})$, $c_2(v_{18} \to v_2)$; (a.1) $r(v_6 \to v_6)$, or (a.2) $r(v_6 \to v_7)$.

Case (a.1): $r(v_6 \to v_6)$ at the end of Round 2 in Case (a).

Round 3 $c_1(v_{12} \to v_{12})$, $c_2(v_2 \to v_1)$; $r(v_6 \to v_7)$.
Round 4 $c_1(v_{12} \to v_{13})$, $c_2(v_1 \to v_1)$; the robber is trapped.

Case (a.2): $r(v_6 \to v_7)$ at the end of Round 2 in Case (a).

Round 3 $c_1(v_{12} \to v_{13})$, $c_2(v_2 \to v_1)$; the robber is trapped.

Case (b): $r(v_9 \to v_8)$ at the end of Round 1.

Round 2 $c_1(v_{12} \to v_9)$, $c_2(v_{18} \to v_{15})$; (b.1) $r(v_8 \to v_5)$, or (b.2) $r(v_8 \to v_{11})$.

Case (b.1): $r(v_8 \to v_5)$ at the end of Round 2 in Case (b).

Round 3 $c_1(v_9 \to v_6)$, $c_2(v_{15} \to v_{10})$; the same as the robber's turn at Round 2 in Case (c) of Scenario V4.

Case (b.2): $r(v_8 \to v_{11})$ at the end of Round 2 in Case (b).

Round 3 $c_1(v_9 \to v_{12})$, $c_2(v_{15} \to v_{10})$; $r(v_{11} \to v_8)$.
Round 4 the same as Round 3 in Scenario V14.

Scenario V9 ends within eight rounds (the longest one is Case (b.1)).

Scenario V11: The robber initially occupies v_{11}.

Round 1 $c_1(v_{13} \to v_{12})$, $c_2(v_{15} \to v_{10})$; $r(v_{11} \to v_8)$.
Round 2 the same as Round 3 in Scenario V14.

Scenario V11 ends within six rounds.

Lastly, we describe the scenarios in group (3), which consists of v_1, v_5, and v_8.

Scenario V1: The robber initially occupies v_1.

Round 1 $c_1(v_{13} \to v_7)$, $c_2(v_{15} \to v_{18})$; $r(v_1 \to v_5)$.
Round 2 $c_1(v_7 \to v_6)$, $c_2(v_{18} \to v_2)$; (a) $r(v_5 \to v_5)$, or (b) $r(v_5 \to v_8)$.
Case (a): $r(v_5 \to v_5)$ at the end of Round 2.

Round 3 $c_1(v_6 \to v_9)$, $c_2(v_2 \to v_2)$; $r(v_5 \to v_5)$.
Round 4 $c_1(v_9 \to v_8)$, $c_2(v_2 \to v_2)$; the robber is trapped.

Case (b): $r(v_5 \to v_8)$ at the end of Round 2.

Round 3 $c_1(v_6 \to v_9)$, $c_2(v_2 \to v_4)$; $r(v_8 \to v_{11})$.
Round 4 $c_1(v_9 \to v_{12})$, $c_2(v_4 \to v_{10})$; $r(v_{11} \to v_8)$.
Round 5 the same as Round 4 in Case (b.2) of Scenario V9.

Scenario V1 ends within seven rounds (the longest one is Case (b)).

Scenario V5: The robber initially occupies v_5.

Round 1 $c_1(v_{13} \to v_{12})$, $c_2(v_{15} \to v_{10})$; (a) $r(v_5 \to v_1)$, (b) $r(v_5 \to v_5)$, or (c) $r(v_5 \to v_8)$.

Case (a): $r(v_5 \to v_1)$ at the end of Round 1.

Round 2 $c_1(v_{12} \to v_9)$, $c_2(v_{10} \to v_4)$; (a.1) $r(v_1 \to v_1)$, or (a.2) $r(v_1 \to v_3)$.

Case (a.1): $r(v_1 \to v_1)$ at the end of Round 2 in Case (a).

Round 3 $c_1(v_9 \to v_6)$, $c_2(v_4 \to v_4)$; $r(v_1 \to v_3)$.
Round 4 $c_1(v_6 \to v_7)$, $c_2(v_4 \to v_2)$; $r(v_3 \to v_{19})$.
Round 5 $c_1(v_7 \to v_{13})$, $c_2(v_2 \to v_{18})$; $r(v_{19} \to v_3)$.
Round 6 the same as Round 2 in Scenario V19.

Case (a.2): $r(v_1 \to v_3)$ at the end of Round 2 in Case (a).

Round 3 $c_1(v_9 \to v_6)$, $c_2(v_4 \to v_2)$; (a.2.1) $r(v_3 \to v_3)$, or (a.2.2) $r(v_3 \to v_{19})$.

Case (a.2.1): $r(v_3 \to v_3)$ at the end of Round 3 in Case (a.2).

Round 4 $c_1(v_6 \to v_6)$, $c_2(v_2 \to v_{18})$; $r(v_3 \to v_3)$.
Round 5 $c_1(v_6 \to v_6)$, $c_2(v_{18} \to v_{19})$; the robber is trapped.

Case (a.2.2): $r(v_3 \to v_{19})$ at the end of Round 3 in Case (a.2).

Round 4 $c_1(v_6 \to v_7)$, $c_2(v_2 \to v_{18})$; $r(v_{19} \to v_{17})$.
Round 5 $c_1(v_7 \to v_{13})$, $c_2(v_{18} \to v_{15})$; $r(v_{17} \to v_{19})$.
Round 6 the same as Round 1 in Scenario V19.

Case (b): $r(v_5 \to v_5)$ at the end of Round 1.

Round 2 $c_1(v_{12} \to v_9)$, $c_2(v_{10} \to v_4)$; $r(v_5 \to v_1)$.
Round 3 the same as Round 3 in Case (a.1).

Case (c): $r(v_5 \to v_8)$ at the end of Round 1.

Round 2 the same as Round 3 in Scenario V14.

Scenario V5 ends within fourteen rounds (the longest one is Case (a.1)).

Scenario V8: The robber initially occupies v_8.

Round 1 $c_1(v_{13} \to v_{12})$, $c_2(v_{15} \to v_{10})$; (a) $r(v_8 \to v_5)$ or (b) $r(v_8 \to v_8)$.

Case (a) : $r(v_8 \to v_5)$ at the end of Round 1.

Round 2 the same as Round 2 in Case (b) of Scenario V5.

Case (b) : $r(v_8 \to v_8)$ at the end of Round 1.

Round 2 the same as Round 3 in Scenario V14.

Scenario V8 ends within seven rounds.

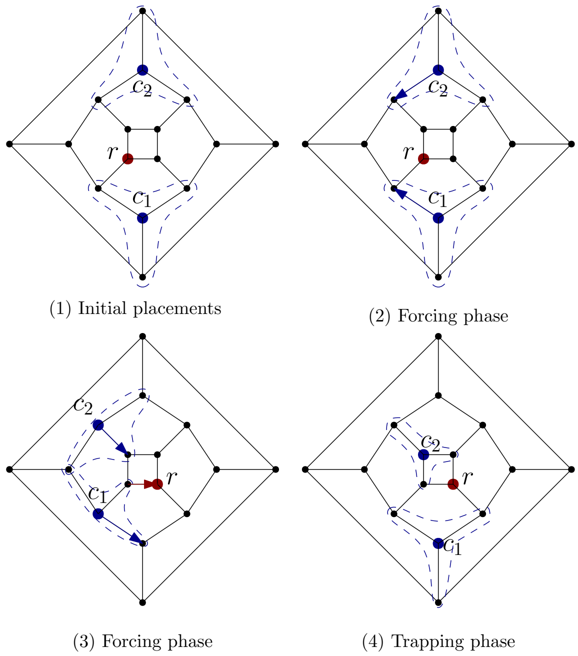
Theorem 4.10. *The edge-contracted dodecahedral graph of order 19 is 2-cop-win.*

Proof. We have given the full capture strategy using two cops for the edge-contracted dodecahedral graph. As described above, the robber can be captured within fourteen rounds. □

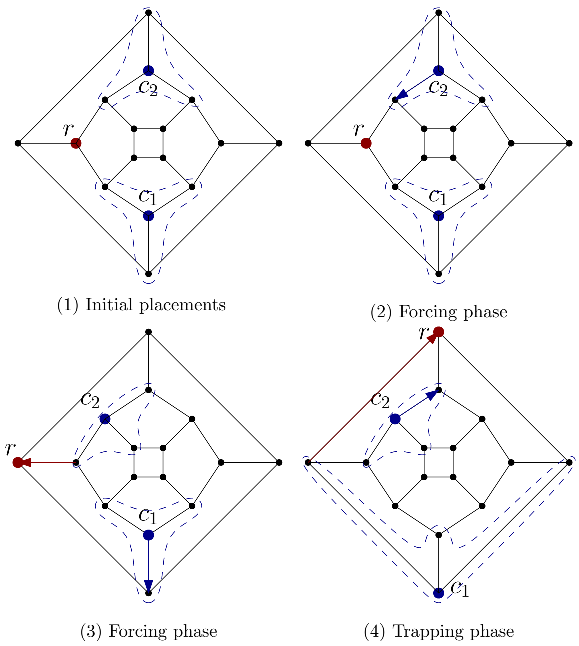
Finally, it is interesting to note that the distance between two cops is always kept to be at most three in our capture strategy. This differentiates our strategy from that of tandem-win graph, in which the distance is kept to be one [13].

4.4.2 The 3-regular graph of order 16

Among the 3-regular planar graphs, the next largest graphs that are smaller than dodecahedral graph are of order 16 (shown in Fig. 4.10). We choose one which has the largest number of 5-cycled faces possible, and provide a strategy for it. In the one for edge-contracted dodecahedral graph, the strategy can be divided into three phases: (1) forcing phase, whose goal is to control the robber's movement and force him toward a 2-cop-winning vertex, and (2) trapping phase, whose goal is to trap the robber at one of the 2-cop-winning vertices. Figure 4.10 shows the two unique scenarios in which the robber is located on two different vertices. Any other vertices that are not in $\bar{N}(C)$ will fall into one of the scenarios.



(a) Scenario in which the robber is initially located on a vertex belonging to three 5-cycles.



(b) Scenario in which the robber is initially located on a vertex belonging to a 4-cycle.

Figure 4.10: Two scenarios of the winning strategy using two cops on 3-regular planar graph of order 16.

4.5 Summary

We have proved in this chapter that a 2-cop-winning vertex, at which the robber is captured by two cops, exists in any planar graph of order at most 19. We have also shown that the planar graph resulted by performing a single edge contraction on the dodecahedral graph is 2-cop-win. Although our strategy for the edge-contracted dodecahedral graph is very elementary, we hope a clever strategy can be developed in future, not only for the edge-contracted dodecahedral graph, but also for any planar graph of order at most 19, so as to prove the conjecture that the dodecahedral graph is the smallest 3-cop-win planar graph.

References

[1] M. Aigner and M. Fromme. "A game of Cops and Robbers". In: *Discrete Applied Mathematics* 8.1 (1984), pp. 1–12.

[5] W. Baird et al. "On the minimum order of k-cop-win graphs". In: *Contributions to Discrete Mathematics* 9 (2014), pp. 70–84.

[8] A. Bonato and R. J. Nowakowski. *The Game of Cops and Robbers on Graphs.* Student Mathematical Library 61. Providence, Rhode Island: American Mathematical Society, 2011.

[10] A. Bondy and U. Murty. *Graph Theory.* Graduate Texts in Mathematics 244. London: Springer-Verlag, 2008.

[13] N. B. Clarke and R. J. Nowakowski. "Tandem-win graphs". In: *Discrete Mathematics* 299.1-3 (Aug. 2005), pp. 56–64.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* Books in the Mathematical Sciences 244. W. H. Freemn and Co., 1979.

[29] P. Pisantechakool and X. Tan. "On the conjecture of the smallest 3-cop-win planar graph". In: *Lecture Notes in Computer Science* 10185 (2017), pp. 499–514. Springer.

[32] D. B. West. *Introduction to Graph Theory.* 2nd ed. Prentice Hall, 2000.

Chapter 5

Towards a Characterization of 2-Cop-Win or 3-Cop-Win Planar Graphs

5.1 Introduction

One of our study objectives is to give a characterization of 2-cop-win or 3-cop-win planar graph. Although it is not obtained in this thesis, we provide some useful evidences. For this purpose, we will give a list of new 3-cop-win planar graphs as well as a list of 2-cop-win graphs.

We investigated whether there exist a 3-cop-win planar graph that have 2-cop-winning vertices. Three new 3-cop-win planar graphs as well as their proofs are given. Some graphs can simply be identified as 3-cop-win using the non-existence of 2-cop-winning vertex.

5.2 The 3-Cop-Win Planar Graphs That Have 2-Cop-Winning Vertices

In Chapter 4, one fact on any 2-cop-win graph is identified and used to prove that an edge-contracted dodecahedral graph is 2-cop-win. The 2-cop-winning vertices are necessary for capture strategy using two cops [29]. However, we mentioned also that another fact, which is a method to move all pieces to their respective goals, is required. And such, even if a 2-cop-winning vertex exists, a graph may not be 2-cop-win. In the following, some planar graphs that have 2-cop-winning vertices are proved to be 3-cop-win.

5.2.1 Vertex-symmetric graphs

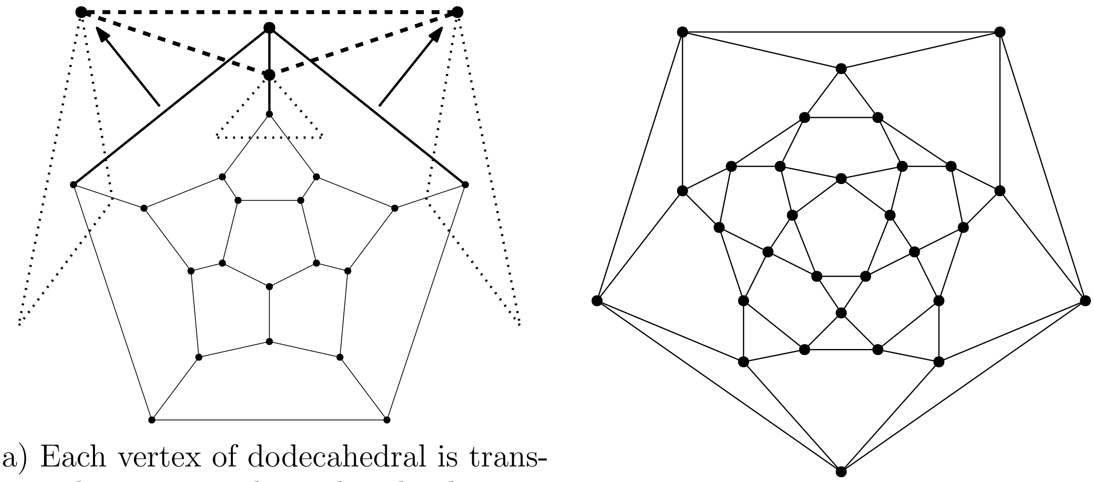
Vertex-symmetric graphs are regular graphs whose every vertex has exactly the same local adjacency and incidence. So, if a 2-cop-winning vertex exists on a vertex-symmetric graph, then all vertices are winning ones. Note that every vertex-symmetric graph is polyhedral, which can be drawn on a spherical object, and has many pairs of vertices whose distances are equal to the graph's diameter.

Icosidodecahedral graph, shown in Fig. 5.1(b), is a 4-regular planar graph of 30 vertices, 60 edges, and 32 faces, whose diameter is five. It is vertex-symmetric because every vertex is common to two 5-cycles and two 3-cycles. It is also a line graph of a dodecahedral graph (the edges of dodecahedral graph are represented by vertices of the icosidodecahedral).

In icosidodecahedral graph, every vertex is a 2-cop-winning vertex, with the winning positions c_1 and c_2 belong to two different 3-cycles whose common vertex v is the winning one (see Fig. 5.2).

Lemma 5.1. Icosidodecahedral graph is 3-cop-win.

Proof. Icosidodecahedral can be transformed into dodecahedral graph, by transforming a 3-cycle into a vertex, and the vertex shared by two 3-cycles into an edge between the transformed vertices. The



(a) Each vertex of dodecahedral is transformed into a 3-cycle, and each edge into the common vertex between two 3-cycles.

(b) An icosidodecahedral graph.

Figure 5.1: A transformation from dodecahedral to icosidodecahedral.

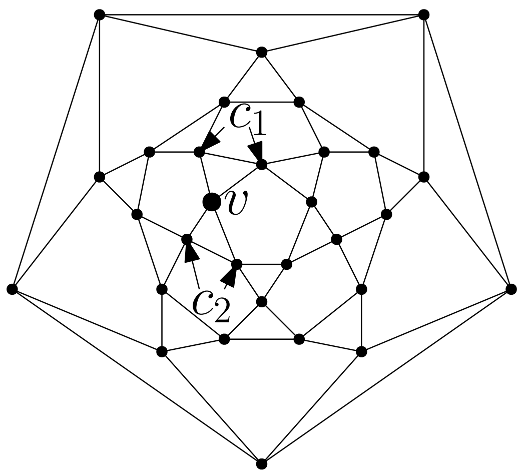
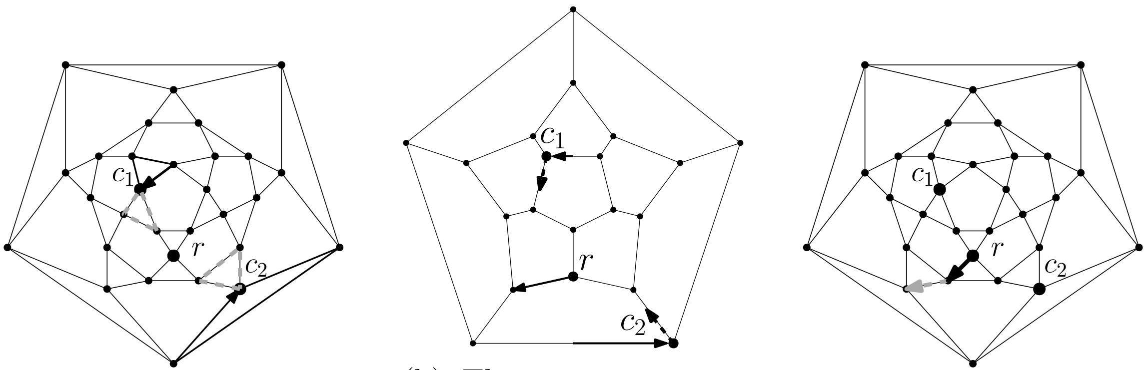


Figure 5.2: The winning positions c_1 and c_2 against a vertex v.



(a) The robber watches how the cops move on their turn on icosidodecahedral.

(b) The movements are mapped to dodecahedral. The robber then determines his next movement on it according to his strategy.

(c) The robber's movement on dodecahedral is mapped back to icosidodecahedral.

Figure 5.3: How the robber maps the winning strategy against two cops from dodecahedral to icosidodecahedral.

transformation can also be made from dodecahedral into icosidodecahedral as well, see Fig. 5.1 for an example.

The movements of the cops on icosidodecahedral graph can be mapped into the movements on dodecahedral graph. For a cop to move out of a 3-cycle in icosidodecahedral graph, two consecutive movements are required; one to determine the source 3-cycle (whose edge is traversed by the cop on this step), and another to move into a new 3-cycle (destination). This two-part sequence of movements on icosidodecahedral is mapped into one movement (moving from one vertex to another) on dodecahedral graph.

Since the robber has a winning strategy against two cops on the dodecahedral graph, he can map the strategy into icosidodecahedral graph, as follows. First, he maps the movements of the cops from icosidodecahedral graph to dodecahedral graph. Any two consecutive movements of the cops on icosidodecahedral can be mapped into one movement on dodecahedral. Note that the mapping can be done by treating odd rounds as first step and even round as second step, or vice versa. After mapping the movements of the cops to dodecahedral graph, the robber finds his next action on it (following his winning strategy), and map that action to icosidodecahedral graph. Since the cop's movement on icosidodecahedral to determine the source 3-cycle also lets the robber know the destination 3-cycle (Fig. 5.3(a)), the robber can always map the cops' movements from icosidodecahedral (by predicting the destination 3-cycles of the cops) to dodecahedral, and determine his next movement on dodecahedral according to the winning strategy (Fig. 5.3(b)). The sequence of the robber's movements on dodecahedral graph is then mapped into the two-part sequence of movements on icosidodecahedral graph (Fig. 5.3(c)). By moving on icosidodecahedral following the mapped movement sequences, the robber can avoid being captured by two cops indefinitely. Hence, icosidodecahedral is 3-cop-win. □

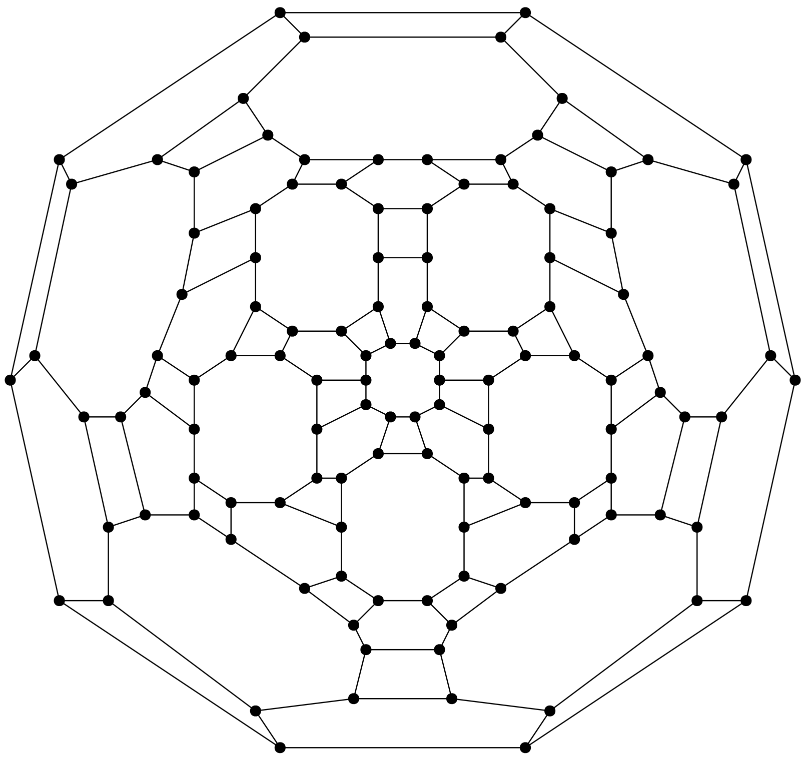
Another example of vertex-symmetric 3-cop-win planar graph with 2-cop-winning vertices is *truncated icosidodecahedral graph* (also known as *great rhombicosidodecahedral graph*), shown in Fig. 5.4(a). It is 3-regular graph, has graph diameter 15, and its faces are made of twelve 10-cycles, twenty 6-cycles, and thirty 4-cycles.

In truncated icosidodecahedral graph, every vertex v belongs to a 4-cycled face, a 6-cycle face, and a 10-cycled face. And thus, every vertex v is a 2-cop-winning vertex. See Fig. 5.4(b) for the winning positions c_1 and c_2 against a vertex v.

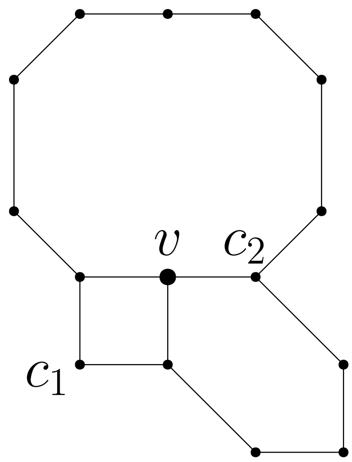
Lemma 5.2. Truncated icosidodecahedral graph is 3-cop-win.

Proof. Similar to icosidodecahedral, truncated icosidodecahedral graph can be transformed into dodecahedral graph, and vice versa. This is done by transforming 4-cycle into an edge, a 6-cycle into a vertex, and a 10-cycle into a 5-cycle of the dodecahedral graph. Again, for a cop (or the robber) to move from one 6-cycle to another, at least two steps are required (first step is to move into a 4-cycle, then another step is into the new 6-cycle).

Since the robber has winning strategy against two cops on dodecahedral graph, he can map the movements of two cops on truncated icosidodecahedral to dodecahedral, find the next action according to the strategy, then map it back to truncated icosidodecahedral graph



(a) Truncated icosidodecahedral graph



(b) The local adjacency and incidence of every vertex v of icosidodecahedral graph, and the winning positions c_1 and c_2 against it.

Figure 5.4: A truncated icosidodecahedral graph and the winning positions against each vertex.

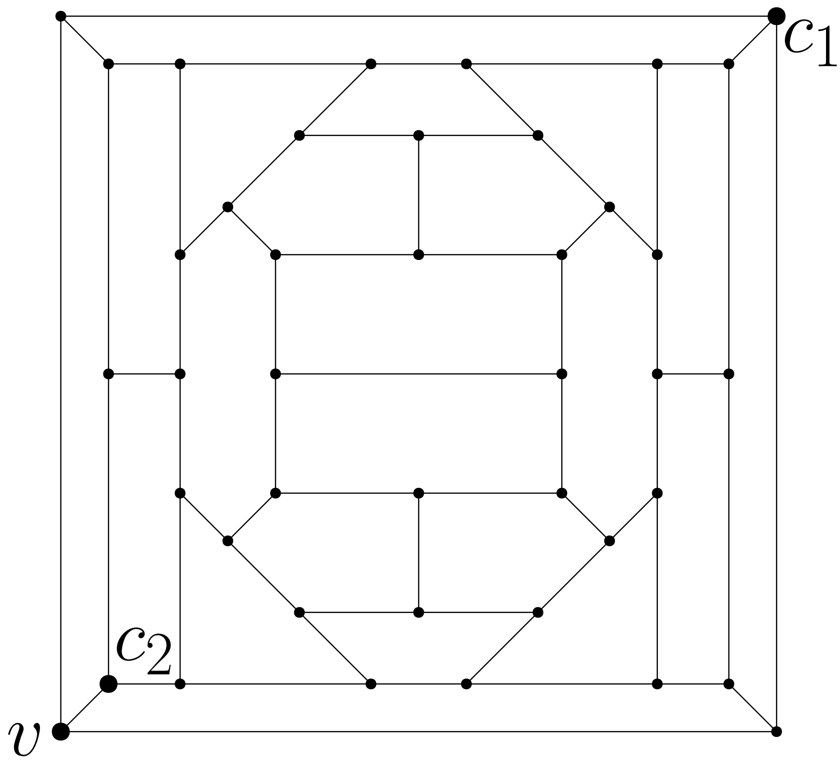


Figure 5.5: A Grinberg's 42 graph.

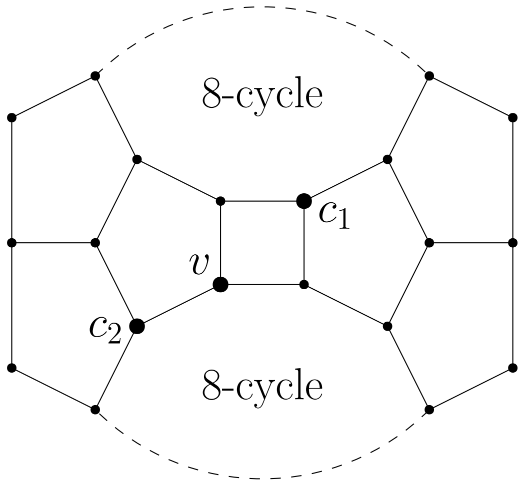
and follow the mapped sequence of movements. Hence, truncated icosidodecahedral graph is 3-cop-win. □

5.2.2 Grinberg's 42 graph

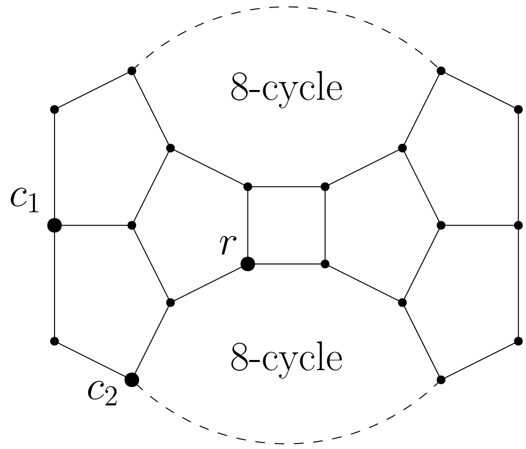
Grinberg's 42 graph (Fig. 5.5) is a 3-regular graph of order 42, with one 4-cycled face. It can be observed that 2-cop-winning vertices exist on such a face, see Fig. 5.6(a). Note that one of the winning positions is the vertex whose distance to the 2-cop-winning vertex is two on the same 4-cycle.

Lemma 5.3. Grinberg's 42 graph is 3-cop-win.

Proof. Using the strategy given in Chapter 4, we assume that two cops can successfully force the robber to move into a winning vertex. To this end, they have to move into the winning positions against it.



(a) A 2-cop-winning vertex and the winning positions against it.



(b) The locations of the cops and the robber after the cops successfully forced the robber into a 2-cop-winning vertex.

Figure 5.6: Portions of Grinberg's 42 in different embedding, where 4-cycle is located in the center.

Because they have to force the robber through a series of 5-cycled faces and vertices of degree three, two cops must prevent the robber from moving into any vertices other than the ones that lead him into 2-cop-winning vertex. So, when the robber is forced to move into it, two cops on the same side of a 4-cycled face in the graph, see Fig. 5.6(b).

At the beginning of the next round, no matter how the cops move, the robber can simply move along the edge belonging to 4- and 8-cycle once, and in the next round escape from the 2-cop-winning vertices and the cops. Since the cops cannot capture the robber anywhere else, they must force the robber to move into a 2-cop-winning vertex again, and the robber can get away using the same escape strategy. Hence, two cops cannot capture the on Grinberg's 42 graph, and thus, it is 3-cop-win. □

The existence of choke-point-like structure contributes to the 3-cop-win for Grinberg's 42 graph. However, we could not get a correct definition yet, and 3-cop-win graphs with such a structure, along with proofs, are needed to classify this type of 3-cop-win planar graphs.

While Grinberg's 42 are 3-regular, it is not vertex-symmetric, since only four vertices belong to 4-cycle (and thus they are different from the rest).

5.2.3 The maximal planar graph of order 92

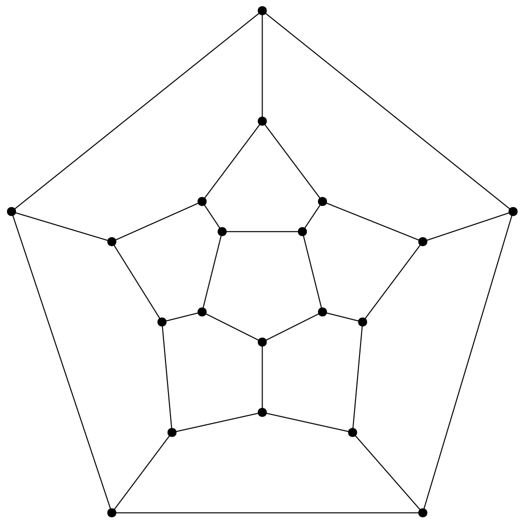
A simple graph is called *maximal planar graph* if it is planar but adding any edge would destroy that property. A maximal planar graph thus has all the faces as 3-cycles. It is also known by other names such as *triangular graph* or *triangulated graph*, but they are often used to refer to the line graph of a complete graph (dual graph in which edges are transformed into vertices) and to the chordal graph, respectively. We use the name maximal planar graph to avoid ambiguity.

One may conjecture that a maximal planar graph are either cop-win (if dominated vertices can be successively removed until a graph becomes single vertex) or 2-cop-win (otherwise). The minimum degree for any planar graph is at most five, and thus 2-cop-winning vertex of degree five, similar to that of Fig. 4.6(e), must also exist. However, there exists a 3-cop-win maximal planar graph.

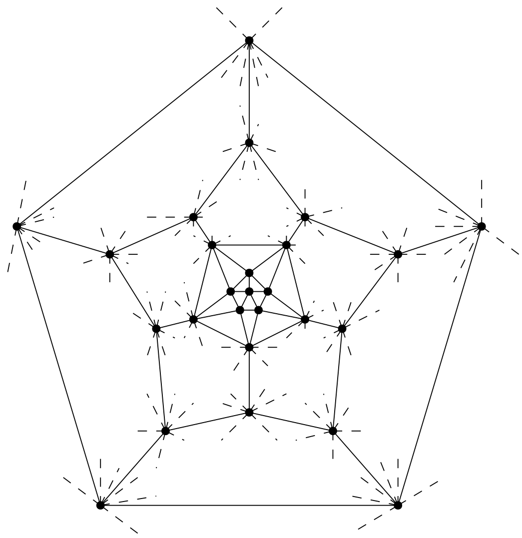
Theorem 5.4. *[[21], Theorem 12.4] There exist maximal planar graphs with cop number three.*

An example graph was constructed from the dodecahedral graph, but instead of directly connecting edges between a pair of vertices in the original graph, they added more vertices such that the paths on the original graph are still the shortest paths for any pair of vertices belonging to the original graph. The construction method can be seen in Fig. 5.7, which adds six extra vertices to each face. Note that the inner pattern shown inside the center face of Fig. 5.7(b), excluding the 5-cycle itself, is maximal. Note also that all the six extra vertices of the inner pattern are 2-cop-winning vertices. By adding this pattern to all twelve faces, the final graph is made maximal, and has 72 more vertices, thus it is of order 92.

The robber can win against two cops in this graph, by moving only on the vertices of the original graph (dodecahedral). It is inefficient to



(a) A dodecahedral graph.



(b) One face of dodecahedral is made maximal.

Figure 5.7: The construction of 3-cop-win maximal planar graph from dodecahedral.

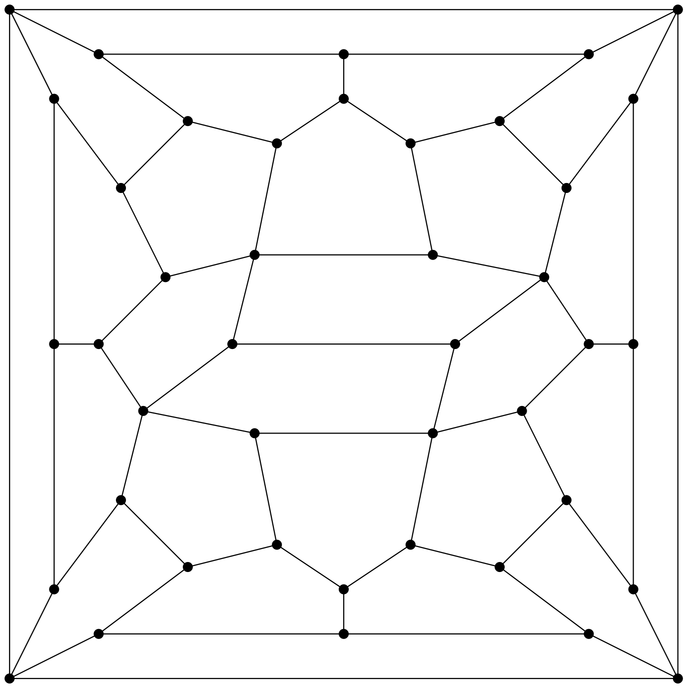
move through the added vertices, so two cops are forced to move along the original graph as well. And since the original graph is 3-cop-win, so is this graph. It should be noted that the vertices in the original graphs are of degree nine, while the added vertices are of degree five. This makes the graph neither regular, nor vertex-symmetric

5.3 Other 3-Cop-Win Planar Graphs

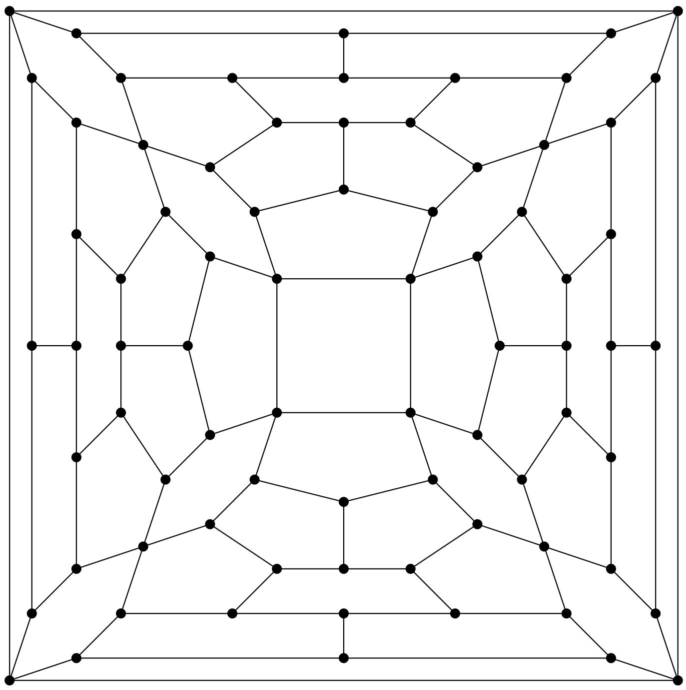
Following the new concept of 2-cop-winning vertices, its non-existence can be used to identify more 3-cop-win planar graphs, or even construct a new one. First, we state the following:

Corollary 5.5. Any planar graph with no 2-cop-winning vertex is 3-cop-win.

It is clear that any planar graph with no 2-cop-winning vertex is 3-cop-win. The first example we found in our study is the Wiener-



(a) Wiener-Araya graph



(b) Hemispheres graph

Figure 5.8: Examples of graphs with no 2-cop-winning vertex.

Araya graph [4]. The graph was intentionally constructed for a different purpose, but it also happens to be 3-cop-win. Figure 5.8(a) shows Wiener-Araya graph. Note that it has one 4-cycle, whose all vertices are of degree four (and thus the graph is neither regular nor vertex-symmetric).

With Corollary 5.5, we purposely constructed the hemispheres graph so that it has two 4-cycles but no 2-cop-winning vertex, similar to that of Wiener-Araya graph. The construction of it starts with a 4-cycle whose all vertices are degree four. Then, we added more 5- and 6-cycles to make the remaining vertices be of degree three, and consider the possibility of repeating the pattern on the other half, hence its name. Figure 5.8(b) shows the complete hemispheres graph. It should be noted also that this graph is neither regular nor vertex-symmetry.

5.4 The known 2-Cop-Win Planar Graphs

As more of 3-cop-win planar graphs are discovered, a list of known classes of 2-cop-win should also be made. These examples and their property may be useful in the characterization of 2-cop-win planar graphs. The followings are known classes of 2-cop-win planar graphs.

1. **Grid graphs:** Grids are known to be 2-cop-win and the 2-cop-winning vertices (the final vertices for the robber prior to his capture) are four vertices at the corners of grids.

2. **Tandem-win graphs:** Clarke and Nowakowski have classified some 2-cop-win graphs in which two cops move together and maintain the distance between them at one [13]. The method of determining a tandem-win graph is by using domination elimination ordering to retract the graph into single vertex. For tandem-win planar graphs, they do not have any cycle of length at least five.

3. **4-regular graphs of order at most 29:** The smallest 4-regular, 3-cop-win planar graph known so far is the icosidodec-

ahedral, which was proved in this thesis. All 4-regular planar graphs, whose order are at most 29, are known to be 2-cop-win.

It should be noted that our strategy provided in Chapter 4 for edge-contracted dodecahedral graph is not tandem-win strategy. And such, there are more 2-cop-win graphs without known method of determination yet.

5.5 Summary

We have given the new 3-cop-win planar graphs that are the results of in our study. Interestingly, some graphs with 2-cop-winning vertices are proved to be 3-cop-win. We have also provided a list of known 2-cop-win planar graphs. The icosidodecahedral graph and its truncated version are vertex-symmetric. Some known 3-cop-win planar graphs, particularly the fullerenes (3-regular graph whose twelve faces are 5-cycles and all other faces are 6-cycles), are also vertex-symmetric. It should be noted that not all 3-cop-win graph are vertex-symmetric, as can be seen in Grinberg's graphs, Wiener-Araya graph, and hemi-spheres graph. We are working to find and classify more graphs that are 2-cop-win or 3-cop-win.

References

[1] M. Aigner and M. Fromme. "A game of Cops and Robbers". In: *Discrete Applied Mathematics* 8.1 (1984), pp. 1–12.

[4] M. Araya and G. Wiener. "On cubic planar hypohamiltonian and hypotraceable graphs". In: *The Electronics Journal of Combinatorics* 18.1 (2011). URL: `http://www.combinatorics.org/ojs/index.php/eljc/article/view/v18i1p85`.

[5] W. Baird et al. "On the minimum order of k-cop-win graphs". In: *Contributions to Discrete Mathematics* 9 (2014), pp. 70–84.

[8] A. Bonato and R. J. Nowakowski. *The Game of Cops and Robbers on Graphs.* Student Mathematical Library 61. Providence, Rhode Island: American Mathematical Society, 2011.

[10] A. Bondy and U. Murty. *Graph Theory.* Graduate Texts in Mathematics 244. London: Springer-Verlag, 2008.

[13] N. B. Clarke and R. J. Nowakowski. "Tandem-win graphs". In: *Discrete Mathematics* 299.1-3 (Aug. 2005), pp. 56–64.

[21] A. Maurer, J. McCauley, and S. Valeva. *Cops and Robbers on planar graphs.* Summer 2010 Interdisciplinary Research Experience for Undergraduates. report. University of Minnesota, 2010.

[29] P. Pisantechakool and X. Tan. "On the conjecture of the smallest 3-cop-win planar graph". In: *Lecture Notes in Computer Science* 10185 (2017), pp. 499–514. Springer.

[32] D. B. West. *Introduction to Graph Theory.* 2nd ed. Prentice Hall, 2000.

Chapter 6

An Algorithm for Computing a Dominating Set for Grids

6.1 Introduction

In this chapter, a real world application of the Cops and Robbers game's variation is studied, and new algorithm is given. The problem of finding a dominating set for a graph is a well-studied problem in graph theory, and has many potential applications in sensor networks and swarm robots, as well as routing problems in mobile networks. The dominating set [16] is a graph problem where every vertex of a given graph $G = (V(G), E(G))$ must be either in a dominating set $U \subseteq V(G)$ or adjacent to a member of the dominating set, and the goal is to find a smallest set U in the graph G. For path graphs and trees, a linear-time algorithm to find a dominating set has been given[12].

Finding a domination number (i.e., the size of a smallest dominating set) of an arbitrary graph is NP-hard [16], and planar graph is also proven to be NP-hard. Grid graphs, which lie in a class of planar graph, have a special structure that allows their domination number to be determined optimally. For $m \times n$ grid, the size of the optimal dominating set was unknown until recently, but the upper bound of $\lfloor \frac{(m+2)(n+2)}{5} \rfloor - 4$ was shown in [12]. It has also been shown that the lower bound of domination number is equal to the upper bound for $m, n \geq 16$, thus characterizing the domination number of grids [17].

Previous efforts were focused on the problem of computing the dominating numbers for grids [2, 12, 16, 17]. Two previous works for computing a dominating set were Chang's doctoral thesis [12] and Fata et al.'s conference paper [14]. The domination number and the upper bound was made through observation of brute-force computational techniques [2]. With that in mind, we aim to develop the algorithms for both centralized and distributed systems. Chang's method is constructive, and one can simply derive from his method to give a centralized algorithm so as to find a dominating set of optimal size $\lfloor \frac{(m+2)(n+2)}{5} \rfloor - 4$. A distributed algorithm was given in [14], which computes a dominating set of size $\lceil \frac{(m+2)(n+2)}{5} \rceil$.

It should be pointed out that the algorithm of Fata et al. [14] is incomplete in its termination stage. A set of agents is initially located at vertices of the grid. The number of agents may be larger than $\lceil \frac{(m+2)(n+2)}{5} \rceil$, and some agents may even be at the same grid vertex. The agents have three modes: (a) sleep, (b) active, and (c) settled. All the agents, in the sleep mode at the beginning, will activate in a randomized or previously scheduled manner. The very first agent becomes settled just at its original vertex. Each active agent can communicate with the settled agents so as to find the place (vertex) where it becomes settled. As soon as settled agents no longer have to communicate, each settled agent goes back to sleep mode. The remaining non-activated agents are required to leave the grid afterwards, but the final operation is NOT a distributed one (see page 5 of [14]). Following their algorithm, the agent being active after the dominating set is found will simply restart the algorithm again, due to the fact that she cannot know whether the dominating set has already been found.

The goal of this chapter is to give a new algorithm to compute a dominating set on grids in a distributed manner [27] (that can terminate correctly). We first define a distributed system model. In particular, each agent is equipped with an answering machine that can record a broadcast message at a time, which is the most updated message. This makes it possible to let the remaining non-activated agents leave the grid. Next, we explore the techniques of Chang's corner handling so that they can work in the distributed system. Our distributed algorithm can produce a dominating set of size $\lfloor\frac{(m+2)(n+2)}{5}\rfloor - 3$, which improves upon the previous result [14] by 4. This is the best result to our knowledge.

Distributed grid domination algorithms can be adopted by distributed systems for many applications. For example, swarm robots equipped with short-ranged land mine detection devices can be deployed from an airplane into a designated area (considered as a grid). These robots can move to align themselves in optimized formation to maximize the coverage on their own without having to manually control them or using the centralized system.

6.2 Dominating Set Computation and the Cops and Robbers Game

A dominating set with minimum cardinality is called an optimal dominating set of a graph G; its cardinality is called the domination number of G and is denoted by $\gamma(G)$. Note that although the domination number of a graph, $\gamma(G)$, is unique, there may be different optimal dominating sets [2]. In this study, we focus the dominating set problem on a special class of graphs called grid graphs. An $m \times n$ grid graph $G = (V(G), E(G))$ is defined as a graph with vertex set $V(G) = \{v_{i,j} | 1 \leq i \leq m, 1 \leq j \leq n\}$ and edge set $E(G) = \{(v_{i,j}, v_{i,j'}) || j - j'| = 1\} \bigcup \{(v_{i,j}, v_{i',j}) || i - i'| = 1\}$ [10]. For ease of presentation, we will fix an orientation and labeling of the vertices, so that vertex $v_{0,0}$ is the lower-left vertex and vertex $v_{m-1,n-1}$ is the upper-right vertex of the grid. In this paper we will include super-grid in grid indices. We denote the domination number of an $m \times n$ grid G by $\gamma_{m,n} = \gamma(G)$.

For a subset $U \subseteq V(G)$, we define $N(U)$ as $\bigcup_{u \in U} N(u)$. For a subset $U \subseteq V(G)$, we say the vertices in $N(U)$ are *dominated* by the vertices in U. For graph G, a set of vertices $U \subseteq V(G)$ is a dominating set if each vertex $v \in V(G)$ is either in U or is dominated by U.

The Cops and Robbers game can be reduced to dominating set problem with the introduction of the capture time. If the cop player is put under pressure to capture the robber in the very first round with the fewest cops as possible, then the cop number is exactly the domination number of the given graph, and the initial positions for the cops are the vertices of the dominating set.

In the followings, we introduce some important results and terms used in domination problem, especially on grids.

Theorem 6.1. *[Gonçalves et al., [17]]. For an $m \times n$ grid with $16 \leq m \leq n, \gamma_{m,n} = \lfloor \frac{(m+2)(n+2)}{5} \rfloor - 4$*

Definition 6.2. (Grid Boundary) For an $m \times n$ grid $G = (V(G), E(G))$, we define the boundary of G, denoted by $B(G)$, as the set of vertices with less than 4 neighbors

Definition 6.3. (Sub-Grids and Super-Grids) An $m \times n$ grid $G = (V(G), E(G))$ is called a sub-grid of an $m' \times n'$ grid $G' = (V(G'), E(G'))$ if G is induced by vertices $v'_{i,j} \in V(G')$, where $1 \leq i \leq m' - 2$ and $1 \leq j \leq n' - 2$. If G is a sub-grid of G', G' is called the super-grid of G (see Fig. 6.1).

Definition 6.4. (Optimal Grid Pattern) A subset $U \subseteq V(G)$ constitutes an optimal grid pattern on grid $G = (V(G), E(G))$ if there exists a fixed $r \in 0, 1, 2, 3, 4$ such that for any vertex $v_{x,y} \in U$ we have $x - 2y \equiv r (\text{mod } 5)$.

One can also define an optimal grid pattern as a set of vertices whose (x, y) coordinates satisfy $y - 2x \equiv r (\text{mod } 5)$, for some fixed r. This corresponds to swapping the x and y axes. For the proofs we only analyze the case mentioned above; the other case can be treated similarly.

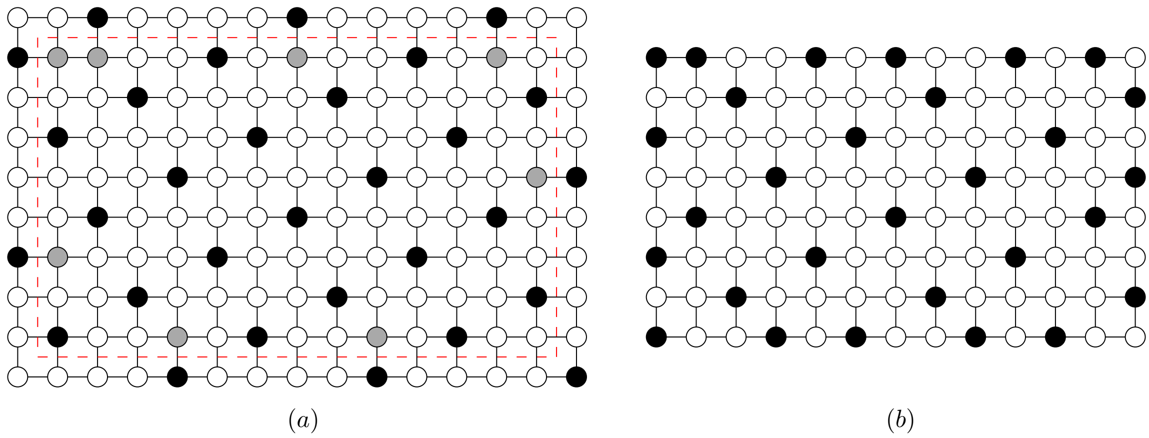


Figure 6.1: A grid G' is demonstrated and its sub-grid G is highlighted in dashed square. Vertices in $U' \setminus V(G)$ are projected onto their neighbors, which are orphans, in G

Definition 6.5. (Grid Optimization) A subset $U \subseteq V(G)$ optimizes grid $G = (V(G), E(G))$ if it constitutes an optimal grid pattern and there exists no vertex $v \in V(G) \setminus U$ that can be added to U so that U remains an optimal grid pattern. See Fig. 6.1(a).

Definition 6.6. (Orphans) Let $U \subseteq V(G)$ be a set of vertices that optimizes grid $G = (V(G), E(G))$. A vertex $v \in V(G)$ that has no neighbor in U is called an orphan (see Fig. 6.1a).

Definition 6.7. (Projection) Consider a grid $G = (V(G), E(G))$ and its super-grid $G' = (V(G'), E(G'))$. For a set $U' \subseteq V(G')$, its projection is defined as the set $U'' = (N(U' \setminus V(G)) \cup U') \cap V(G)$. Similarly, we say a vertex $v \in U' \setminus V(G)$ is projected if it is mapped to its neighbor in $V(G)$. See Fig. 6.1.

Definition 6.8. (Slot) Given a $m \times n$ grid $G = (V(G), E(G))$, let $v_{i,j} \in V(G)$ be a vertex occupied by a settled agent. The four vertices $\{v_{i+2,j+1}, v_{i-1,j+2}, v_{i-2,j-1}, v_{i+1,j-2}\}$ from $v_{i,j}$ within the boundary of G are called slots of the settled agent occupying $v_{i,j}$.

Definition 6.9. (Pseudo-Slot) When an agent settles at corner points, the agent may assign a vertex as pseudo-slot. Pseudo-slots have the same priority as slots when an active agent seeks a location to occupy, but the agent who settles there will not calculate its slots and instead go into sleep mode. See Fig. 6.6.

6.3 Chang's Centralized Constructive Method Revisited

In this section we revisit Chang's centralized constructive method that can produce a dominating set of size $\lfloor\frac{(m+2)(n+2)}{5}\rfloor - 4$ for an $m \times n$ grid, $m, n \geq 8$. Chang's constructive method consists of the following three main ideas:

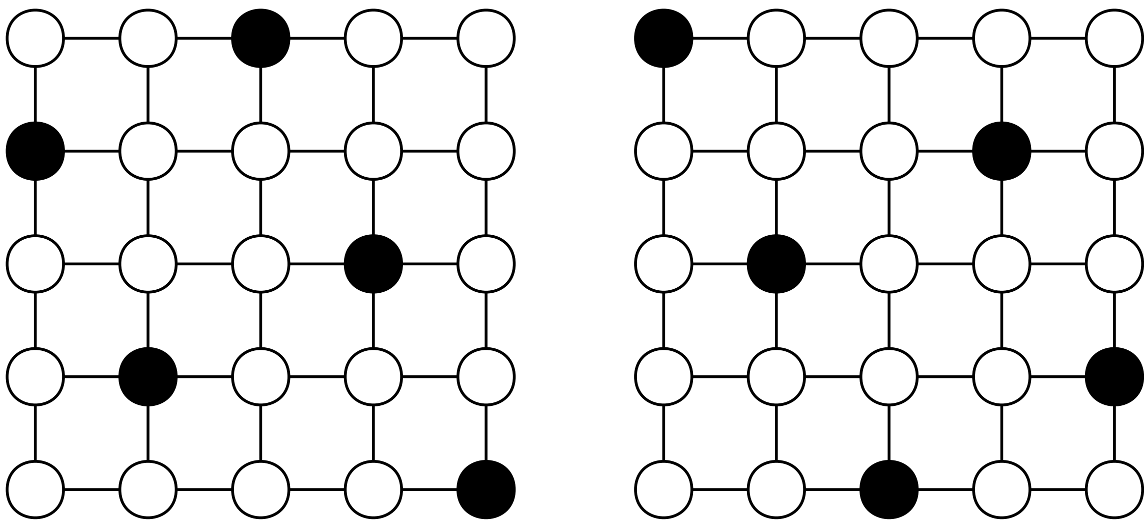
(i) Initialization: At this step, a subset $U' \subseteq V(G')$ that optimizes the super-grid G' is provided. Basically, it can select the smaller between two permutations (of super-grid) of the optimal grid pattern. See Fig. 6.1(a).

(ii) Corner Handling: Each corner (i.e., a 5×5 portion) of the super-grid has one vertex removed from U', and the vertices around four corners of the super-grid are moved into the original grid. See Fig. 6.3.

(iii) Projection: Using a process called projection, the vertices in $U' \setminus V(G)$ except for four corners are characterized and put into the original grid G.

Initialization: As stated in Definition 3 and proven in [12], for a given $m \times n$ grid graph, there exist some r in $x - 2y \equiv r (\text{mod } 5)$ such that $|S| \leq \lfloor\frac{(m+2)(n+2)}{5}\rfloor$. Careful observation showed that the optimal grid pattern repeats itself every 5×5 block. It is known that when the pattern is shifted around in one super-grid, it produces different number of the dominating vertices. There are five disjoint permutations of the pattern, based on $r \in \{0, 1, 2, 3, 4\}$ in Definition 3. For one grid size, some permutations produce smaller number of dominating vertices, but when the size changed, others may produce smaller number.

For a known size of grid, picking a suitable permutation can further



(a) Permutation α (b) Permutation β

Figure 6.2: Two permutations

reduce the number of elements. We can simply use two permutations whose maximum sizes of produced sets in all possible grid size do not overlap.[1] That is, one can always choose smaller number of dominating vertices between the two and get the minimum size. We refer to these two permutations as Permutation α and β (shown in Fig. 6.2). The derived algorithm uses these two disjoint permutations in Initialization step.

Corner Handling and **Projection**: One can further reduce the number of dominating vertices around the corners of a grid. Recall that using Projection (Definition 6.7) from super-grid, we can dominate the orphans (as stated in Definition 6.6). For grid with large size, the vertices on the boundary that are not a part of a corner must be dominated by projected vertices. However, at each corner, some elements overlap and the placement is not ideal.

By performing Corner Handling step before Projection step, we can reduce one vertex at each corner before moving all the vertices at boundary of super-grid and while doing so, move vertices around

[1] Note also that Chang's constructive method has to choose from five different permutations based on input grid size.

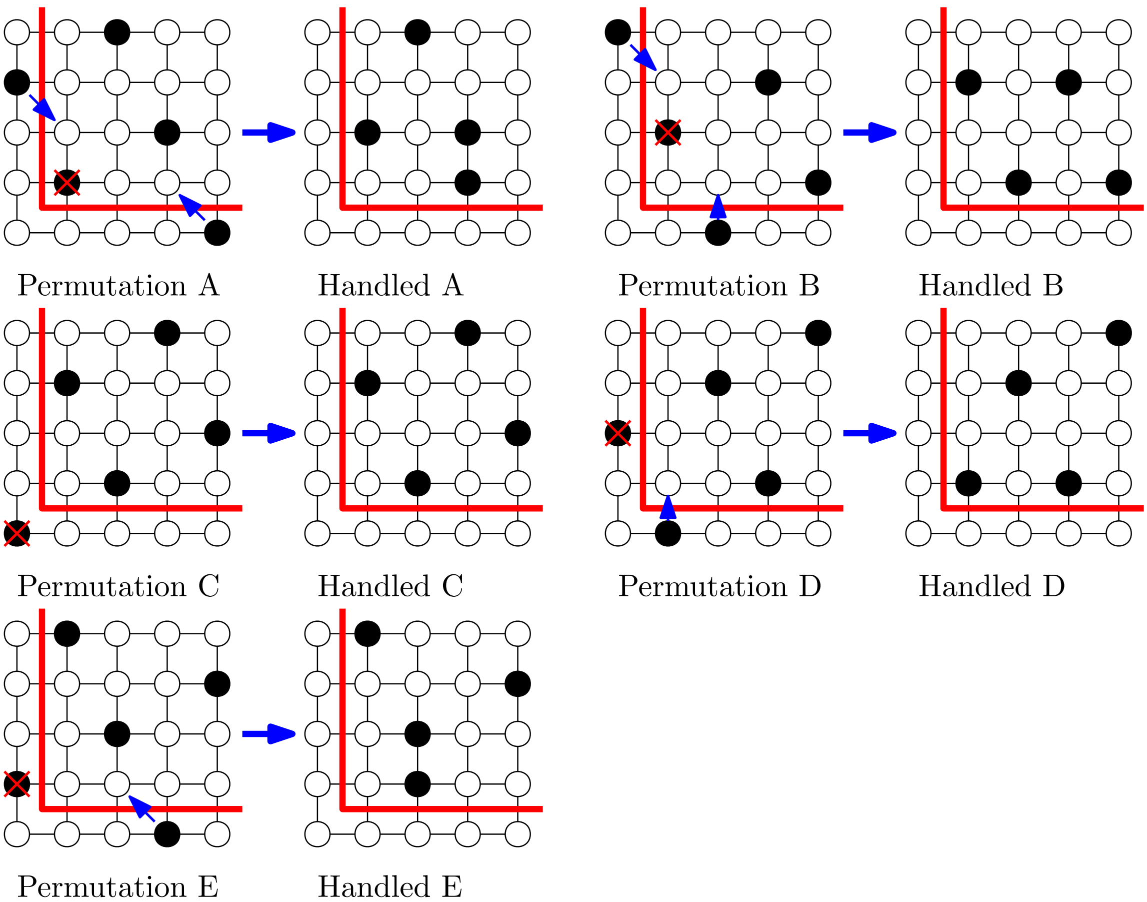


Figure 6.3: Handling each corner's permutation

the corner to the original grid. Chang's case-based handling method considers each corner as a 5×5 block and handles each permutation differently, as shown in Fig. 6.3.

Lastly, as previously stated in Definition 6.7, we use projection process to move the remaining vertices on the boundary of super-grid to its sub-grid (the original grid), and dominating the remaining orphans.

6.4 Our Distributed Algorithm for Computing a Dominating Set

In this section, we first introduce the distributed system model. (A prototype model can be found in [14].) Next, we show that Chang's centralized constructive method, especially the Corner Handling Step, can be extended so as to compute a dominating set of size $\lfloor \frac{(m+2)(n+2)}{5} \rfloor - 3$ for a given $m \times n$ grid in a distributed manner, $m, n \geq 8$.

Assume that the environment is an $m \times n$ grid $G = (V(G), E(G))$ with $m, n \in \mathbb{N}$. The goal is to dominate the grid environment in a distributed fashion using several robots (or agents) without any knowledge of environment size. At the start, there exist k agents in the environment, where k can be smaller or greater than the number of agents needed to dominate the grid. The following assumptions are made for the grid and agents.

Assumption A: Agents can be located only on the vertices of the grid, and can move between the grid vertices only on the edges of the grid. More than one agent can be at the same vertex at any given time. We refer to the vertices using standard Cartesian coordinates defined in Section 6.2[2].

Assumption B: The agents, denoted by $a_1, \ldots, a_k$ are initially located at arbitrary vertices on the grid. The agents have three modes; (a) sleep, (b) active, and (c) settled. The sleep mode in this algorithm means that the agent will not contribute to the distributed algorithm,

[2]We include vertices of super-grid in labeling for the ease of presentation.

but can still perform other unrelated functions, such as detecting intruders or land mines.

Assumption C: Each agent is equipped with an answering machine that can record a broadcast message at a time when the agent is in sleep mode. At the beginning of the procedure, all agents are in sleep mode. During each epoch, or time interval with specific length, one agent goes into active mode. The activation sequence of agents is arbitrary; it can be scheduled in advance, or randomized, but each agent will activate at fix length of time after the previous one.

Assumption D: Agents in active mode and settled mode can communicate. The active agent can communicate with the settled agents to perform the distributed dominating set algorithm. Once an agent activates and performs its parts, it goes into settled mode. After the settled agents form a dominating set, all of them go back to sleep mode and will not activate again.

Assumption E: Each agent is equipped with suitable bearing sensor (incoming direction) and range sensor to help computing the location of the sender of signal it receives from, in its own local coordinates with itself as origin and an arbitrary orientation.

Additionally, agents are equipped with short-ranged proximity sensors to sense the environment boundary. Agents are able to sense the boundary only if they are on a vertex v whose neighbor is a boundary vertex of the grid.

The idea of our algorithm is to implement the optimal grid pattern used in the centralized algorithm in a distributed manner, using communications among active and settled agents.

Agents will keep track of surrounding four locations (vertices) that are correctly aligned in optimal grid pattern. These locations are called slots (Definition 6.8), and any new agents becoming active later will communicate and attempt to occupy these slots to contribute to the optimal grid pattern. An agent at $v_{i,j}$ uses itself as origin and keep tracking of $\{v_{i+2,j+1}, v_{i-1,j+2}, v_{i-2,j-1}, v_{i+1,j-2}\}$ locations. If there exists a slot outside the boundary, an agent will keep a location of neighbor vertex whose position is on the boundary instead. These locations are called orphans as defined in Definition 6.6, and each agent outside the corner area will have at most one orphan tracked at a time. An orphan of a vertex v, denoted as orphan(v), is a vertex $u \in B(G)$ such that $u \in N(v)$. Orphans are occupied by agents after all slots are occupied, which allow us to move around some agents in similar manner to that of centralized grid domination algorithm.

Corollary 6.10. (Orphan Number) Given a $m \times n$ grid G, for any vertices outside 4 corner points (Fig. 6.5) of each corner, an agent will have at most one orphan when aligned with optimal grid pattern.

When an agent activates, it checks the most recent message to see whether there exists a message or not. Initially, there is no message in any agents' short-memory storage in message receptor, the agent then concludes that itself is the first agent to activate. After, the first agent checks for boundary to see whether this initial location is around the corner or not before settle. If an agent finds itself at the corner, it moves to ideal location around the corner instead. If there exists a message but not termination message, the agent then sends a broadcast signal to find the settled agents on the grid who have slots available, and waits for some specified time for response. The new active agent then contributes to optimal grid pattern using information received from settled agents.

We break down the distributed algorithm into three main steps for the ease of presentation.

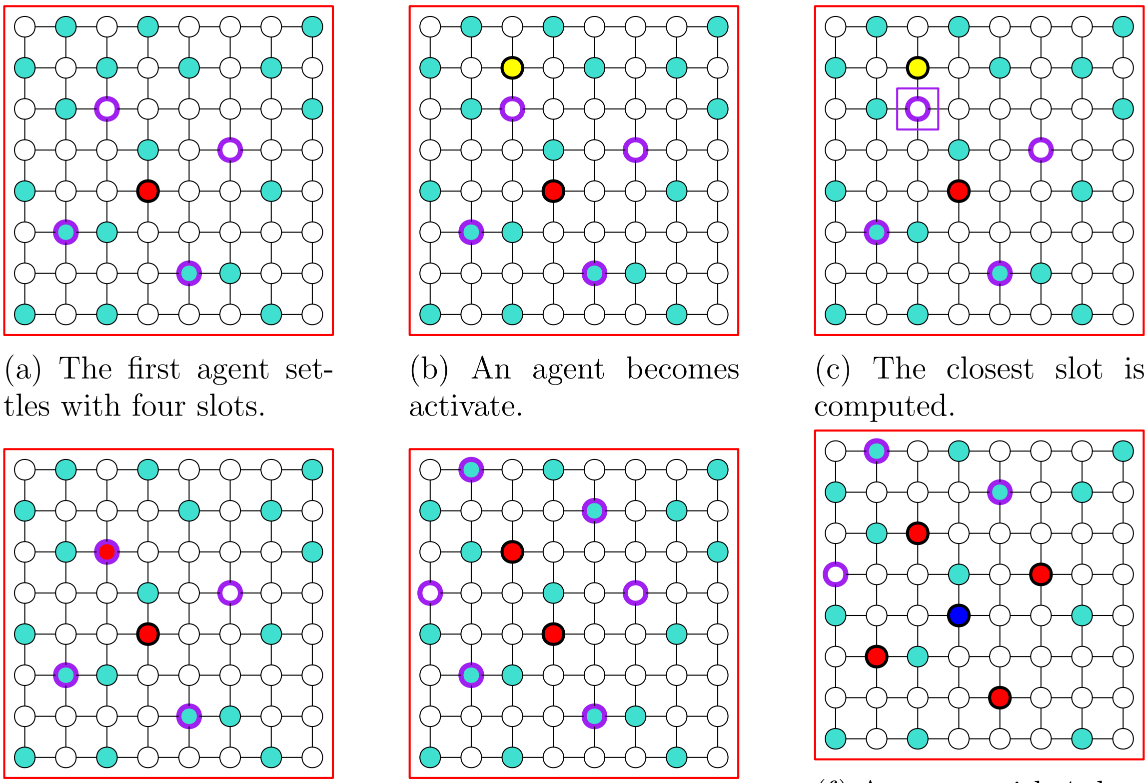
(i) Initialization: How the very first activated agent works.

(ii) Settlement: How active and settled agents communicate, and how an active agent gets settled.

(iii) Termination: How the algorithm finishes. Particularly, how the termination condition is verified in the case that the number of agents is larger than necessary to dominate the grid.

6.4.1 Initialization

As stated in *Assumption C*, the algorithm starts with all agents being in sleep mode. Each agent will activate at certain time apart from one another.



(a) The first agent settles with four slots.

(b) An agent becomes activate.

(c) The closest slot is computed.

(d) The active agent occupies that slot.

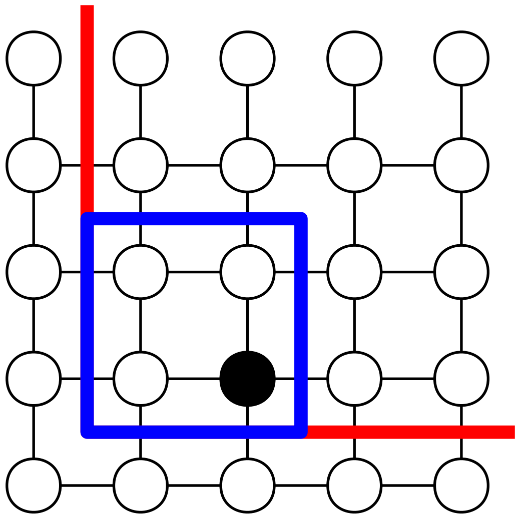
(e) The lists of slots are updated.

(f) An agent with 4 slots occupied goes into sleep mode.

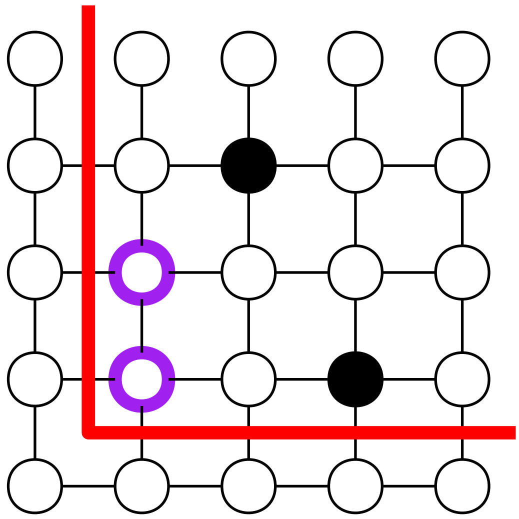
Figure 6.4: Examples of the algorithm.

Each agent acts similarly when it first activates. The first thing each agent does is checking most recently stored message to see if there exists a termination message. We will describe about termination message later in Subsection 6.4.3. Since at the beginning, no message has been broadcast yet, the agent will not see a single saved message and then conclude that it is the *First Agent* to activate in the system.

The first agent has special action to take before entering settled mode. First, the agent must check whether it is at one of the four corner points or not, using functions described in *Assumption E*. This can be done by utilizing short-range proximity sensor to sense environmental boundary. If it finds itself in one of the four corner points, as illustrated in Fig. 6.5(a), it will move to the specified location of that corner. By



(a) Permutation C: if the first agent activates at corner points (black square), it will move to designated location (black circle).



(b) Permutation D: a termination condition with two orphans.

Figure 6.5: Special instances for first agent (a) and last agent (b).

doing so, we can assume that for any corners, the agents are forming optimal grid pattern from any locations outside the four corner points, or starting a Permutation C corner.

If the first agent activates outside corner points, it will enter settled mode normally. Before an agent enters settled mode, it will make a list of unoccupied locations around itself that aligned in optimal grid pattern (Fig. 6.4(a)). After an agent enters settled mode, the list of unoccupied slots and orphan is updated when another active agent occupies any of them.

6.4.2 Settlement

For other agents activating after the first one, they take the following actions; check the recorded message, broadcast for slots, compute for closest location to occupy, then move to occupy and settle at computed location.

Since there is at least one settled agent after the first, the active agents will receive response signal from settled agents. Active agents can contribute to the algorithm by either settle in unoccupied slots, or settle in pseudo-slots or orphans. An active agent will compute for closest settled agent among those that responded and send request for slot list. Chosen settled agent sends out its whole list to active agent, who then computes for closest location of eligible slots. After determining the location to settle using $L^1 norm$ distance, active agent sends out notification that the location will be taken to all settled agents, and travels to the location. Settled agents whose lists hold such location in slot list then remove the location from their owns list. Once a list is empty and no orphan in the surrounding, a settled agent will go into sleep mode (Fig. 6.4(f)).

Once the active agent reaches the chosen location, it will first check whether it is at corner points or not, in similar fashion to that of the first agent. If an active agent finds itself in one of the corner point, it will take special action, called corner settlement, as shown in Fig. 6.6. This is the step similar to corner handling in centralized algorithm, but performed in a distributed fashion. For example, if an active agent chooses a corner point $v_{1,1}$, shown as Permutation A in Fig. 6.6, it will move to new location $v_{1,2}$ and create a list with only one location, called pseudo-slot, shown as $v_{3,1}$ in Fig. 6.6. A pseudo-slot is considered as a slot by settled agents when responding to request signal, thus both slots and pseudo-slots will be occupied before orphans. Note that Permutation D does not have any agent at corner point, so it is omitted.

If the chosen location for an active agent is not corner point, the active agent will create a list of slots and orphan normally, similar to that of first agent.

When an active agent receives a location marked as pseudo-slot and chooses it to settle, it will notify settled agents of its choice then move to the location. However, it will not create a list of slots and orphan, and instead go directly into sleep mode.

Eventually all slots and pseudo-slots are occupied, and active agents will not receive any response signal when requesting for slots. Active agents then sends out request signal for orphans, and settled agents

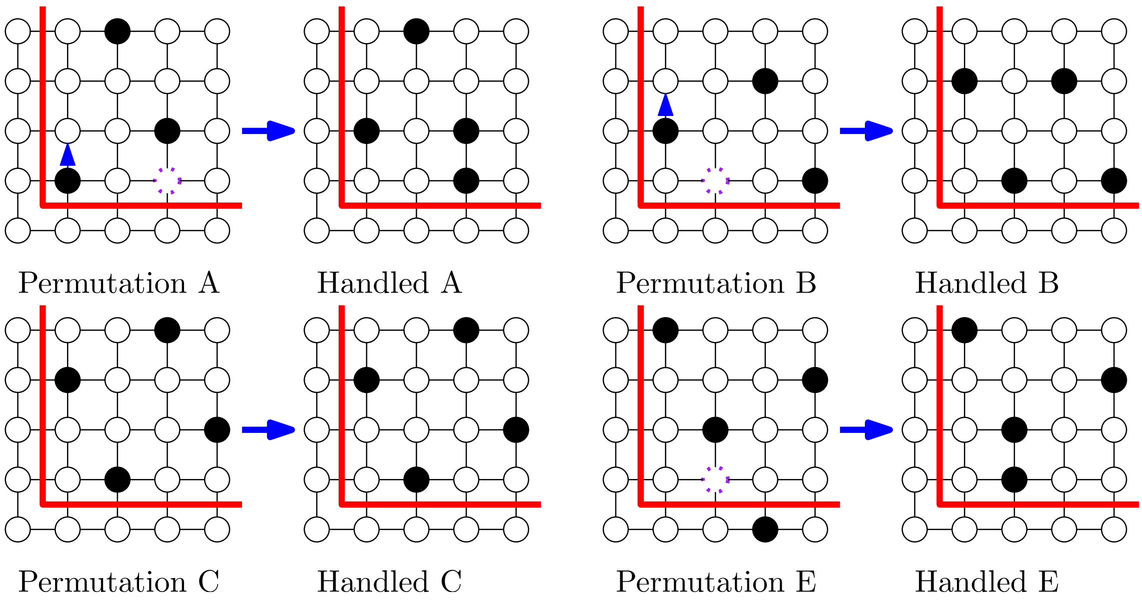


Figure 6.6: Distributed algorithm's case-based method on how to handle each corner point

with orphans in their lists will respond by sending orphan locations. An active agent then computes for closest location then notifies all settled agents of its choice. Like pseudo-slot, active agents settling at orphans will go directly into sleep mode.

6.4.3 Termination

Termination of the distributed algorithm normally happens in the following cases.

Case 1: The number of agents is not enough to dominate the grid. After settled agents receive no broadcast signal for a fixed amount of time (longer than an interval of activation sequence), all the settled agents will enter sleep mode, making themselves a subset of dominating set of the grid.

Case 2: The number of agents in the grid is more than enough to dominate the grid. The distributed algorithm will produce a complete dominating set for the grid. In this case, algorithm has three different conditions for last active agent to check.

(a) Only one settled agent responds to an active agent with only one orphan.

(b) Two settled agents respond to an active agent, but both have the same orphan.

(c) Two settled agents respond to an active agent, but both orphans are in the corner (Fig. 6.6b).

If one of the conditions is satisfied, the last active agent will send out the termination message after notifying settled agents that it will occupy the location, and then go directly into sleep mode. The termination message will be the last broadcast message, and stored in answering machine's memory of every agent, including agents that have yet to activate.

Any agent activates after the broadcast will see termination message as the most updated message, and leave the grid without contributing to the distributed algorithm.

6.4.4 The Algorithm

We now provide a complete algorithm and prove that it is correct and creates dominating set for grid correctly according to initialization step in centralized algorithm.

Algorithm: DistributedGridDomination

Initialization

1. First agent activates.
2. First agent concludes that it is the first agent because there is no stored message.
3. First agent checks for corner points as described in Subsection 6.4.1, then moves to designated location as necessary.
4. First agent settles at its current location.

Settlement

1. Other agents activate one by one in uniformly distributed interval. Activated agent checks stored message, making sure that it is not a termination message.

2. Active agents communicate with settled agents to find the closest slot to settle.
   (a) Active agent sends out request signal for slots. Settled agent responds if it has unoccupied slots.
   (b) Active agent computes the closest settled agent, then sends request for a list of all slot locations. Settled agent sends list of unoccupied slots.
   (c) Active agent computes the closest slot then notifies other settled agents. Settled agents remove to-be-occupied location.
   (d) Active agent moves to closest slot, then checks for corner points and proceeds as described in Subsection 6.4.2.
   (e) **If** active agent is at pseudo-slot location, **then** it goes directly into sleep mode.
   (f) Active agent settles at current location and becomes settled agent, then updates its list of slots.
3. During step 2.(a), **If** active agent receives no response, **then** it sends out request signal for orphans and occupy orphan as stated in Subsection 6.4.2.

Termination

1. During Settlement step 3., **If** active agent detects any of the following conditions; (i) only one settled agent responds, or (ii) two settled agents respond with the same orphan or two different orphans which are at corner points, **then** it sends out termination message.
2. During Settlement step 2.(a), **If** settled agents receives no request signal for a fixed period of time, **then** the algorithm terminates.
3. During Settlement step 1., **If** active agent sees termination message, **then** it leaves the grid.

Lemma 6.11. During the algorithm, the agents occupying non-pseudo, non-orphan slots contribute to the optimal grid pattern correctly.

Proof. An active agent always settles in slots computed by other settled agents. Since the slots computed by an settled agent are aligned with the settled agent in optimal grid pattern according to Definition

6.4, as long as the new active agent settles at non-pseudo, non-orphan slots of previously settled agents, it will contribute to the optimal grid pattern correctly. □

Theorem 6.12. *The number of agent used to dominate the grid in our algorithm is bounded to* $\lfloor\frac{(m+2)(n+2)}{5}\rfloor - 3$ *for any* $m \times n$ *grid such that* $m, n \geq 8$

Proof. Our distributed grid domination algorithm computes a dominating set correctly with the exception that the smaller of two permutations cannot be chosen in the distributed manner. So, the size of the computed dominating agent set is upper-bounded to $\lfloor\frac{(m+2)(n+2)}{5}\rfloor -$ 3. □

Note that agents may perform their tasks such as traversing the grid in at most $m+n$ steps, and number of agents required to dominate the grid is bounded to $\lfloor\frac{(m+2)(n+2)}{5}\rfloor - 3$, the running time of algorithm can be upper-bounded polynomial time of $O(mn(m+n))$ steps to construct a dominating set algorithm on an $m \times n$ grid.

6.5 Summary

We presented an algorithm to the problem of finding dominating sets on an $m \times n$ grid where $m, n \geq 8$ in the distributed manner. First, we briefly revisited Chang's centralized algorithm that obtains a dominating set of size $\lfloor\frac{(m+2)(n+2)}{5}\rfloor - 4$. We then presented our distributed algorithm that computes a dominating set of size $\lfloor\frac{(m+2)(n+2)}{5}\rfloor - 3$ using similar methods under the restrictions of the distributed system models.

References

[2] S. Alanko et al. "Computing the Domination Number of Grid Graphs". In: *The Electronics Journal of Combinatorics* 18.1 (July

2011). URL: `http://www.combinatorics.org/ojs/index.php/eljc/article/view/v18i1p141`.

[10] A. Bondy and U. Murty. *Graph Theory*. Graduate Texts in Mathematics 244. London: Springer-Verlag, 2008.

[12] G. J. Chang. "Domination numbers of grid graphs". PhD thesis. University of South Florida, 1992.

[14] E. Fata, S. L. Smith, and S. Sundaram. "Distributed dominating sets on grids". In: *2013 American Control Conference*. Ed. by T. Gopal, G. Jaeger, and S. Steila. New York: IEEE Press, 2013, pp. 211–216.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Books in the Mathematical Sciences 244. W. H. Freemn and Co., 1979.

[17] D. Gonçalves et al. "The Domination Number of Grids". In: *SIAM Journal of Discrete Mathematics* 25.3 (2011), pp. 1443–1453. URL: `http://epubs.siam.org/doi/abs/10.1137/11082574`.

[27] P. Pisantechakool and X. Tan. "A new distributed algorithm for computing a dominating set on grids". In: *Lecture Notes in Computer Science* 9130 (2015), pp. 217–228. Springer.

Chapter 7

Conclusions

The Cops and Robbers game in graph theory has been studied in many variations. However, in the most basic of them, there are still many subjects and properties that have not been known or explored yet. We are particularly interested in the applications of the Cops and Robbers game, and most of them are set to play on planar graphs. It is known that a planar graph has a cop number of at most three, but the capture time had not been established. Moreover, the characterization of 2-cop-win or 3-cop-win planar graph has not been found yet. In this thesis, we studied the Cops and Robbers game on planar graph on two topics and contributed to one of the applications.

On the topic of capture time, it had only been studied on classes of cop-win graphs (using one cop) and grids (using two cops) so far, and was proved to be linear in both. For planar graphs, although it had not been studied extensively, Aigner and Fromme have given a winning strategy using three cops [1]. However, its capture time is observed to be quadratic ($O(n^2)$). In Chapter 2, we obtained a new winning strategy for planar graphs using three cops. Its capture time was proved to be linear ($2n$) in Chapter 3. This established that the capture time of the Cops and Robbers game on planar graphs using three cops is linear.

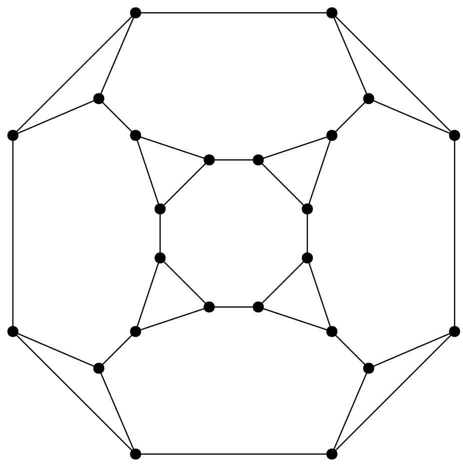
On the topic of cop number, for a given graph, the method to determine whether it is cop-win graph was given [23, 30]. However, a definite method to determine whether a planar graph is 2-cop-win or

3-cop-win has not been found yet, and we are interested in finding one through characterization of 2-cop-win or 3-cop-win planar graphs. To this end, the smallest 3-cop-win planar graph should be established first, so that any planar graphs of smaller order would be classified as 2-cop-win. We conjectured that the dodecahedral graph is the smallest 3-cop-win planar graph. Although we could not prove it yet, we have shown in Chapter 4 that for any planar graph of order at most 19, at least one 2-cop-winning vertex exists. We also gave some winning strategies using two cops for special planar graphs of order 16 and 19. We have also found new classes of 3-cop-win graphs (discussed in Chapter 5), which may give useful insight in finding the characterization of 2-cop-win or 3-cop-win planar graphs.

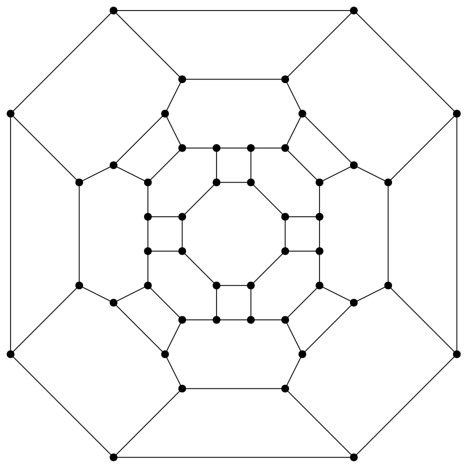
The Cops and Robbers game is related to the dominating set problem in graph as well. In this thesis, we studied an application of Cops and Robbers game in the field of motion planning for swarm robots. Swarm robots can be used to solve a sensor-coverage problem by transforming the given area into a grid, and make the robots move into positions such that the grid is dominated. Fata et al. [14] have given a distributed algorithm which can dominate any $m \times n$ grid using at least $\lfloor \frac{(m+2)(n+2)}{5} \rfloor + 1$ robots. In Chapter 6, we improved on minimum number of robots required to dominate a grid, and gave our distributed algorithm which can dominated any $m \times n$ grid using at least $\lfloor \frac{(m+2)(n+2)}{5} \rfloor - 3$ robots. This is the closest result to the domination number of a $m \times n$ grid (the smallest size of the dominating set), which is $\lfloor \frac{(m+2)(n+2)}{5} \rfloor - 4$.

In the following, we give some topics for further work.

(i) We are working to prove the conjecture that dodecahedral graph is the smallest 3-cop-win planar graph. One of the directions we may choose is to turn the strategies for special graphs into one that can be used for a particular class of planar graphs. We made an observation that $\bar{N}(c_1)$ and $\bar{N}(c_2)$ always form a connected subgraph, and the distance between two cops is at most three. This differentiates our strategy from tandem-win, in which the distance between two cops is at most one. If we choose this direction, we need to give a proper



(a) Truncated cubical graph of order 24

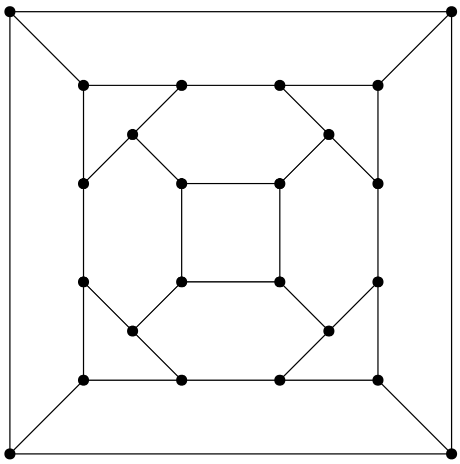


(b) Truncated cuboctahedral graph of order 48

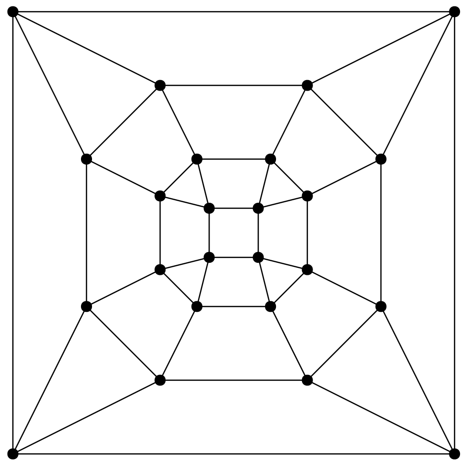
Figure 7.1: Some other examples of vertex-symmetric planar graphs which are also 3-cop-win.

definition to the class of planar graphs that this strategy is valid for. However, this may not be sufficient, because there may exist some planar graphs of order at most 19 which do not fall into tandem-win or our new class. Another direction is to use brute-force method, by enumerating all the possible planar graphs of order at most 19 and give winning strategy using two cops for all of them. This direction particularly takes time and resource, even after eliminating known 2-cop-win planar graphs.

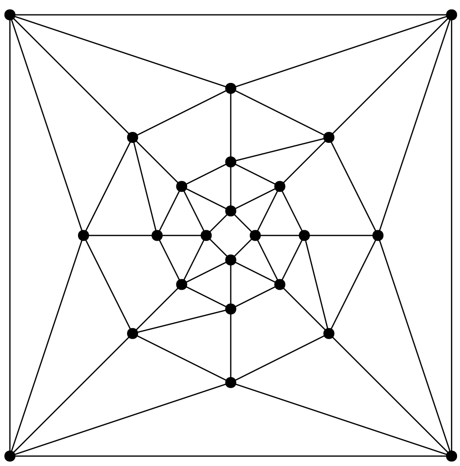
(ii) Suppose that G is a vertex-symmetric planar graph, $\delta(G) \geq 3$ and $|V(G)| \geq 20$. We believe that such a graph G without any two adjacent k-cycle faces, $k = 3$, 4 or 6, is 3-cop-win. Dodecahedral graph is known, and we provided the proofs that icosidodecahedral and its truncated version are 3-cop-win. We also show two other vertex-symmetric graphs in Figs. 7.1, which have 2-cop-winning vertices but are 3-cop-win (without proof in this thesis). These graphs do not have any two adjacent k-cycles, $k = 3$, 4 or 6. On the other hands, vertex-symmetric graphs which are known (or were proved) to be 2-cop-win so far have two adjacent k-cycle faces, $k = 3$, 4 or 6, see Fig. 7.2 for examples. We are working to prove or disprove this hypothesis.



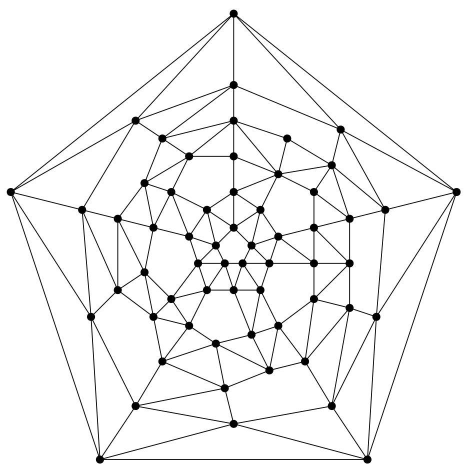
(a) Truncated octohedral graph of order 24



(b) Rhombicuboctohedral graph of order 24



(c) Snub cubical graph of order 24



(d) Snub dodecahedral graph of order 60

Figure 7.2: Examples of vertex-symmetric planar graphs which are 2-cop-win.

Note that if $\delta(G) = 2$, then G is simply a cycle of size $|V(G)|$ and thus it is 2-cop-win.

(iii) Finally. we are going to classify more planar graphs into 2-cop-win or 3-cop-win and study their properties and structures. Among the new 3-cop-win graphs we found in our study, Grinberg's 42 has unique structure. Its choke-like structure could be used to construct some new class of 3-cop-win graphs as well. The method of constructing 3-cop-win maximal planar graph, e.g., from dodecahedral graph, has a special property that makes it 3-cop-win. That is, the shortest paths between any pair of vertices of the original graph are through the vertices and edges of the original graph only. This made it possible to construct different 3-cop-win graphs (which are not maximal), as long as it has such a property. It is also possible to use this property to reduce the given planar graph (without changing its cop number) and determine the cop number from the reduced graph instead. By classifying more planar graphs and studying their properties, we may eventually find the characterization of 2-cop-win or 3-cop-win, and thus obtain the method to determine the cop number of any planar graph.

Contribution List of This Thesis

1. Photchchara PISANTECHAKOOL and Xuehou TAN, "On the capture time of Cops and Robbers game on a planar graph". in: *Lecture Notes in Computer Science* **Vol. 10043**, pp. 3-17, 2016. DOI:10.1007/978-3-319-48749-6_1
   The content of this publication is included in chapters 2 and 3.

2. Photchchara PISANTECHAKOOL and Xuehou TAN, "On the conjecture of the smallest 3-cop-win planar graph", in: *Lecture Notes in Computer Science* **Vol. 10185**, pp. 499-514, 2017. DOI:10.1007/978-3-319-55911-7_36
   The content of this publication is included in chapter 4.

3. Photchchara PISANTECHAKOOL and Xuehou TAN, "A new distributed algorithm for computing a dominating set on grids". in: *Lecture Notes in Computer Science* **Vol. 9130**, pp. 217-228, 2015. DOI:10.1007/978-3-319-19647-3_21
   The content of this publication is included in chapter 6.

Bibliography

[1] M. Aigner and M. Fromme. “A game of Cops and Robbers”. In: *Discrete Applied Mathematics* 8.1 (1984), pp. 1–12.

[2] S. Alanko et al. “Computing the Domination Number of Grid Graphs”. In: *The Electronics Journal of Combinatorics* 18.1 (July 2011). URL: `http://www.combinatorics.org/ojs/index.php/eljc/article/view/v18i1p141`.

[3] B. Alspach. “Searching and sweeping graphs. a brief survey”. In: *Le Mathematiche* 59 (2004), pp. 5–37.

[4] M. Araya and G. Wiener. “On cubic planar hypohamiltonian and hypotraceable graphs”. In: *The Electronics Journal of Combinatorics* 18.1 (2011). URL: `http://www.combinatorics.org/ojs/index.php/eljc/article/view/v18i1p85`.

[5] W. Baird et al. “On the minimum order of k-cop-win graphs”. In: *Contributions to Discrete Mathematics* 9 (2014), pp. 70–84.

[6] D. Bhaduaria et al. “Capturing an evader in polygonal environments with obstacles. the full visibility case”. In: *The International Journal of Robotics Research* 31.10 (2012), pp. 1176–1189.

[7] D. Bienstock and P. Seymour. “Monotonicity in graph searching”. In: *Journal of Algorithm* 12 (2011), pp. 239–245.

[8] A. Bonato and R. J. Nowakowski. *The Game of Cops and Robbers on Graphs.* Student Mathematical Library 61. Providence, Rhode Island: American Mathematical Society, 2011.

[9] A. Bonato et al. "The capture time of a graph". In: *Discrete Mathematics* 309 (2009), pp. 5583–5595.

[10] A. Bondy and U. Murty. *Graph Theory*. Graduate Texts in Mathematics 244. London: Springer-Verlag, 2008.

[11] G. J. Chang. "Algorithmic aspects of domination in graphs". In: *Handbook of Combinatorial Optimization*. Ed. by P. M. Pardalos, D.-Z. Du, and R. L. Graham. 2nd ed. New York: Springer-Verlag, 2013, pp. 221–282.

[12] G. J. Chang. "Domination numbers of grid graphs". PhD thesis. University of South Florida, 1992.

[13] N. B. Clarke and R. J. Nowakowski. "Tandem-win graphs". In: *Discrete Mathematics* 299.1-3 (Aug. 2005), pp. 56–64.

[14] E. Fata, S. L. Smith, and S. Sundaram. "Distributed dominating sets on grids". In: *2013 American Control Conference*. Ed. by T. Gopal, G. Jaeger, and S. Steila. New York: IEEE Press, 2013, pp. 211–216.

[15] S. Gal. *Search Games*. Academic Press, 1980.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Books in the Mathematical Sciences 244. W. H. Freemn and Co., 1979.

[17] D. Gonçalves et al. "The Domination Number of Grids". In: *SIAM Journal of Discrete Mathematics* 25.3 (2011), pp. 1443–1453. URL: `http://epubs.siam.org/doi/abs/10.1137/11082574`.

[18] R. Isaacs. *Differential games. A mathematical theory with applications to warfare and pursuit, control and optimization*. SIAM series in applied mathematics. New York: Wiley, 1965.

[19] A. Kehagias, G. Hollinger, and S. Singh. "A graph search algorithm for indoor pursuit-evasion". In: *Mathematical and Computing Modeling* 50 (2008), pp. 1305–1317.

[20] A. S. LaPaugh. “Recontamination does not help to search a graph”. In: *Journal of the ACM* 40 (1993), pp. 224–245.

[21] A. Maurer, J. McCauley, and S. Valeva. *Cops and Robbers on planar graphs*. Summer 2010 Interdisciplinary Research Experience for Undergraduates. report. University of Minnesota, 2010.

[22] A. Mehrabian. “The capture time of grids”. In: *Discrete Mathematics* 311 (2011), pp. 102–105.

[23] R. J. Nowakowski and P. Winkler. “Vertex-to-vertex pursuit in a graph”. In: *Discrete Mathematics* 43 (1983), pp. 235–239.

[24] W. L. Othling Jr. “Applications of differential game theory to pursuit-evasion problems of two aircrafts”. PhD thesis. Air Force Institute of Technology, 1970.

[25] T. D. Parsons. “Pursuit-evasion in a graph”. In: *Theory and Applications of Graphs. Proceedings*. Ed. by Yousef Alavi and Don R. Lick. Lecture Notes in Mathematics 642. Heidelberg, Berlin: Springer, 1978, pp. 426–441.

[26] T. D. Parsons. “The search number of a connected graph”. In: *The Ninth Southeastern Conference on Combinatorics, Graph Theory, and Computing. Proceedings*. Ed. by F. Hoffman. Congressus Numerantium 21. Heidelberg, Berlin: Utilitas Mathematica Publishing Incorporated, 1978, pp. 549–554.

[27] P. Pisantechakool and X. Tan. “A new distributed algorithm for computing a dominating set on grids”. In: *Lecture Notes in Computer Science* 9130 (2015), pp. 217–228. Springer.

[28] P. Pisantechakool and X. Tan. “On the capture time of Cops and Robbers game on a planar graph”. In: *Lecture Notes in Computer Science* 10043 (2016), pp. 3–17. Springer.

[29] P. Pisantechakool and X. Tan. “On the conjecture of the smallest 3-cop-win planar graph”. In: *Lecture Notes in Computer Science* 10185 (2017), pp. 499–514. Springer.

[30] A. Quilliot. “Jeux et pointes fixes sur les graphes”. Thèse de 3ème cycle. French. PhD thesis. Université de Paris VI, 1978, pp. 131–145.

[31] A. Quilliot. “Some results about pursuit games on metric spaces obtained through graph theory techniques”. In: *European Journal of Combinatorics* 7.1 (1986), pp. 55–66.

[32] D. B. West. *Introduction to Graph Theory*. 2nd ed. Prentice Hall, 2000.