Allocating patrolling resources to effectively thwart intelligent attackers

Thomas Oliver Lowbridge

Thesis submitted to The University of Nottingham for the degree of Doctor of Philosophy

March 2021

Abstract

This thesis considers the allocation of patrolling resources deployed in an effort to thwart intelligent attackers, who are committing malicious acts at unknown locations which take a specified length of time to complete. This thesis considers patrolling games which depend on three parameters; a graph, a game length and an attack length. For patrolling games, the graph models the locations and how they are connected, the game length corresponds to the time-horizon in which two players, known as the patroller and attacker, act and the attack length is the time it takes an attacker to complete their malicious act. This thesis defines patrolling games (as first seen in [16]) and explains its known properties and how such games are solved. While any patrolling game can be solved by a linear program (LP) when the number of locations or game length is small, this becomes infeasible when either of these parameters are of moderate size. Therefore, strategies are often evaluated by knowing an opponent's response and with this, patroller and attacker strategies give lower and upper bounds on the optimal value. Moreover, when tight bounds are given by strategies these are optimal strategies. This thesis states known strategies giving these bounds and classes for which patrolling games have been solved. Firstly, this thesis introduces new techniques which can be used to evaluate strategies, by reducing the strategy space for best responses from an opponent. Extensions to known strategies are developed and their respective bounds are given using known results. In addition we develop a patroller improvement program (PIP) which improves current patroller strategies by considering which locations are currently under performing. Secondly, these general techniques and strategies are applied to find solutions to a certain class of patrolling games which are not previously solved. In particular, classes of the patrolling game are solved when the graph is multipartite or is an extension of a star graph. Thirdly, this thesis conjectures that a developed patroller strategy known as the random minimal full-node cycle is optimal for a large class of patrolling games, when the graph is a tree. Intuitive reasoning behind the conjecture is given along with computational evidence, showing the conjecture holds when the number of locations in the graph is less than 9. Finally, this thesis looks at three extensions to the scenario modelled by the patrolling game. One extension models varying distances between locations rather than assuming locations are a unitary distance apart. Another extension allows the time needed for an attacker to complete their malicious act to vary depending on the vulnerability of the location. For the final extension of multiple players we look at four variants depending on how multiple attackers succeed in the extension. In each extension we find some properties of the game and show that it possible to relate extensions to the classic patrolling game in order to find the value and optimal strategies for certain classes of such games.

Acknowledgements

"If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is."-John von Neumann

"Pleasure to me is wonder, the unexplored, the unexpected, the thing that is hidden and the changeless thing that lurks behind superficial mutability."-H.P. Lovecraft

To my supervisors, Dr. David Hodge, Dr. David Sirl and Prof. Frank Ball, thank you for the generous time given to aiding my studies and going above and beyond at the last moment.

To my family, who this journey would not have been possible without, I thank you from the bottom of my heart for supporting me when I thought about giving up.

To my girlfriend Louise, who stood by me with late nights and morning coffee, I can not thank you enough and will forever be repaying your kindness.

To my friends, you might not have known what I was rambling on about, but thank you for listening and easing my mind when my studies had the better of me.

Finally to you dear reader, I thank you for taking the time to read what is the culmination of my life's ambition for the past four years.

Contents

List of Symbols

General symbols

Symbol	Description
$G(Q, T, m)$	Patrolling game (see pg. 18)
$G(Q, D, T, m)$	Patrolling game with edge distances (see pg. 196)
$G(Q, T, \boldsymbol{m})$	Patrolling game with node dependent attack length (see pg. 216)
$G_i(k, l, Q, T, m)$	i^{th} Multiple player patrolling game (see pgs. 228, 241, 245, 247)
$Q = (N, E)$	Graph with nodes N and edges E (see pg. 18)
T	Game length (see pg. 18)
$\mathcal{J}$	Time horizon (see pg. 18)
m	Attack length (see pg. 18)
$d(j, j')$	(Shortest) Distance between nodes j and j' (Def. 2.3.17)
$\bar{d}$	Diameter/Maximal distance between any two nodes (Def. 2.3.22)
$D(e)$	Distance associated with edge e (see pg. 196)
$d(j, j', N')$	(Shortest) Distance between nodes j and j', not using nodes in N' (see pg. 54)
W	Pure patroller strategy/Walk (see pg. 18)
$a = (j, \tau)$	Pure attacker strategy (see pg. 19)
τ	Commencement time (see pg. 19)
$\mathcal{T}$	Commencement set (see pg. 20)
$\mathcal{W}$	Set of pure patroller strategies (see pg. 19)
β_1	Bijection/Numbering for set of pure patroller strategies (see pg. 19)
$\mathcal{A}$	Set of pure attacker strategies (see pg. 19)
β_2	Bijection/Numbering for set of pure attacker strategies (see pg. 19)
$\boldsymbol{\pi}$	(Mixed) Patroller strategy (see pg. 23)
$\pi_{\beta_1(W)}$	Probability walk W is played by $\boldsymbol{\pi}$ (see pg. 23)
Π	Set of (mixed) patroller strategies (see pg. 23)
$\boldsymbol{\phi}$	(Mixed) attacker strategy (see pg. 23)
$\phi_{\beta_2(a)}$	Probability attack a is played by $\boldsymbol{\phi}$ (see pg. 23)

Symbol	Description
$\varphi_{j,\tau}$	Probability attack (j, τ) is played by $\boldsymbol{\phi}$ (see pg. 23)
$\boldsymbol{S}$	Space-time probability matrix (see pg. 23)
$\boldsymbol{S}^A$	Space-time agent matrix (see pg. 23)
Φ	Set of (mixed) attacker strategies (see pg. 23)
$P(W, a)$	Pure payoff of W against a (see pg. 20)
$P(\boldsymbol{\pi}, \boldsymbol{\phi})$	(Mixed) Payoff of $\boldsymbol{\pi}$ against $\boldsymbol{\phi}$ (see pg. 24)
$V(Q, T, m)$	Value of (classic) patrolling game (see pg. 25)
$V_{\boldsymbol{\pi},\bullet}(Q, T, m)$	Performance of $\boldsymbol{\pi}$ (see pg. 28)
$V_{\boldsymbol{\pi},\bullet,j}(Q, T < m)$	Performance of $\boldsymbol{\pi}$ at node j (Def. 3.3.27)
$V_{\bullet,\boldsymbol{\phi}}(Q, T, m)$	Performance of $\boldsymbol{\phi}$ (see pg. 28)
ω	Move-wait walk (Def. 3.2.9)
Ω	Set of move-wait walks (Def. 3.2.9)
Ω'''	Reduced set of move-wait walks (see Thm. 3.2.13)
N_A	Set of attacked nodes (see pg. 49)
$N_A(\omega, i-1)$	Set of nodes which can be visited by ω for the i^{th} node (see pg. 62)
$t_i(\omega)$	Arrival time at i^{th} node (Def. 3.2.9)
ν_i	Waiting time at i^{th} node (Def. 3.2.9)
$n_i(\omega)$	New start time for potential attack capture at i^{th} node (see pg. 56)
$l_i(\omega)$	Last time the i^{th} node was visited (see pg. 56)
$T'(\boldsymbol{\phi})$	Restricted game length under $\boldsymbol{\phi}$ (see pg. 67)
$\mathcal{Q}^{-l}$	Simplification mapping with l identifications (see pg. 80)
$\mathcal{Q}^{+k}$	Expansion mapping with k splittings (see pg. 80)
$\mathcal{C}_{Q,T,m}$	Covering number (Def. 2.3.11)
$\mathcal{I}_{Q,T,m}$	Independence number (Def. 2.3.20)

Graphs

Symbol	Description
C_n	Cyclic graph with n nodes (see pg. 39)
K_n	Complete graph with n nodes (see pg. 39)
$\mathcal{H}$	Set of all Hamiltonian graphs (Def. 2.3.5)
$Gr(n_1, n_2)$	n_1 by n_2 grid graph (see pg. 105)
$\mathcal{P}_2$	Set of all bipartite graphs (see pg. 39)
$K_{a,b}$	Complete bipartite graphs with sets of size a and b (see pg. 39)
S_n	Star graphs with n leaf nodes (see pg. 40)
$\mathcal{KP}_2$	Set of all complete bipartite graphs (see pg. 39)
$\mathcal{P}_k$	Set of all k-partite graphs (Def. 3.5.4)
$K_{a_1,\dots,a_k}$	Complete k-partite graph with sets of size $a_1, \dots, a_k$ (Def. 3.5.4)
$\mathcal{KP}_k$	Set of all complete k-partite graphs (Def. 3.5.4)
S_n^k	Elongated star graph with n leaf nodes with a k branch elongation (Def. 4.2.1)
$\mathcal{SE}$	Set of all elongated star graphs (Def. 4.2.1)
$S_n^{\boldsymbol{k}}$	Generalised star graph with n leaf nodes with $\boldsymbol{k}$ branch elongations (Def. 4.3.1)
$\mathcal{SG}$	Set of all generalised star graphs (Def. 4.3.1)
(S_{n_1}, S_{n_2})	Dual star with n_1 and n_2 leaf nodes (Def. 4.4.1)
$\mathcal{SD}$	Set of all dual star graphs (Def. 4.4.1)
$(S_{n_1}^{\boldsymbol{k}_1}, \dots, S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)$	p-linked general star graph with centre link Q_c (Def. 4.4.5)
$\mathcal{SL}$	Set of linked general star graphs (Def. 4.4.5)
$(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}})$	Distant general star and distance map (Def. 6.1.12)

Strategies

Symbol	Strategy
$\boldsymbol{\pi}_{\mathrm{cw}}$	Choose-wait patroller strategy (Def. 2.3.3)
$\boldsymbol{\pi}_{\mathrm{rH}}$	Random Hamiltonian patroller strategy (Def. 2.3.6)
$\boldsymbol{\pi}_{\mathrm{Cov}}$	Covering patroller strategy (Def. 2.3.10)
$\boldsymbol{\pi}_{\mathrm{Dec}}$	Decomposition patroller strategy (Def. 2.3.13)
$\boldsymbol{\pi}_{\mathrm{ADec}}$	Arbitrary decomposition patroller strategy (Def. 3.3.1)
W^{Q}_{MFNC}	Minimal full-node cycle for graph Q (Def. 3.3.24)
$\boldsymbol{\pi}^{Q}_{\mathrm{RMFNC}}$	Random minimal full-node cycle strategy for graph Q (Def. 3.3.24)
$\boldsymbol{\pi}_{\mathrm{SimpHyb}}$	Simple hybrid strategy (see pg. 132)
$\boldsymbol{\pi}_{\mathrm{CombHyb}}$	Combinatorial hybrid strategy (see pg. 139)
$\boldsymbol{\pi}_{\mathrm{AdjCombHyb}}$	Adjusted Combinatorial hybrid strategy (see pg. 143)
$\boldsymbol{s}_{\mathrm{irH}}$	Improved random Hamiltonian (Def. 6.2.8)
$\boldsymbol{s}_{\mathrm{rsH}}$	Random spread Hamiltonian (Def. 6.3.5)
$\boldsymbol{s}^{Q}_{\mathrm{RSMFNC}}$	Random spread minimal full-node cycle for graph Q (Def. 6.3.9)
$\boldsymbol{\phi}_{\mathrm{pu}}$	Position-uniform attacker strategy (Def. 2.3.15)
$\boldsymbol{\phi}_{\mathrm{Ind}}$	Independent attacker strategy (Def. 2.3.19)
$\boldsymbol{\phi}_{\mathrm{di}}$	Diametric attacker strategy (Def. 2.3.23)
$\boldsymbol{\phi}_{\mathrm{tdi}}$	Time-limited diametric attacker strategy (Def. 3.3.12)
$\boldsymbol{\phi}_{\mathrm{poly}}$	Polygonal attacker strategy (Def. 3.3.15)
$\boldsymbol{\phi}_{\mathrm{epoly}}$	Exterior polygonal attacker strategy (Def. 3.3.17)
$\boldsymbol{\phi}_{\mathrm{W}}$	Weighted attacker strategy (Def. 4.2.3)
$\boldsymbol{\phi}_{\mathrm{tc}}$	Time-centred attacker strategy (Def. 4.2.5)
$\boldsymbol{\phi}_{\mathrm{type}}$	Type-centred attacker strategy (Def. 4.3.4)
$\boldsymbol{\phi}_{\rho-\mathrm{stc}}$	ρ-simplified time-centred attacker strategy (see pg. 4.2.10, Def. 4.2.18, 4.2.20, 4.2.22)
$\boldsymbol{\phi}_{\mathrm{ts}}$	Time-spread attacker strategy (Def. 4.4.2)
$\boldsymbol{\phi}_{\mathrm{apu}}$	Augmented position uniform attacker strategy (Def. 6.2.10)

Chapter 1

Introduction

In this introduction, we provide our motivation behind studying the ideas of 'Allocating patrolling resources to effectively thwart intelligent attackers' in order to prevent security issues from arising in a multitude of scenarios. We then provide an overview of the content which is presented in this thesis, which will focus on patrolling games from [16], the content of which is primarily focused on analytical solutions to such games. Finally this chapter contains a literature review, explaining various models which have some resources to allocate/control in order to find, stop or capture another resource. Our literature review concludes with a summary of past work on patrolling games; with the mathematical background to patrolling games taking place in chapter 2.

1.1 Motivation

Security is becoming an ever-increasing problem in today's fast developing society. Threats to the public are created in various ways, from physical acts of terror to cybercrime and technological breaches exposing confidential information. Terrorism has become an increasing threat within the 21st century, all over the world. Extremists, varying in their opinions and religion, often feel that an act of terror will help convey their beliefs. Unfortunately, these acts mainly target innocent civilians in differing numbers. However, a terror attack even injuring one civilian, is one injury and one terror attack too many. Terror attacks are not specific to countries, nationalities, gender or economic status, leaving everyone in the world susceptible to possible attacks. One of the most memorable terror attacks occurred on the 11th September in 2001 in New York and other locations in the United States. The twin towers; the North and South building of the World Trade Centre, were crashed into by two commercial airliners. This devastating act killed nearly 3,000 people including 19 hijackers involved in the attack ([32]). Not only that, 6,000 people were injured, and the clean-up of the horrific crash-site took a year to complete ([111]). This resulted in global shock and many nations offered their support and solidarity. To this day, this attack has never been forgotten and has left an ever-lasting impression. Terrorism and their attacks have occurred throughout history but continue to this day. More recently in 2017, a suicide bomber blew up the Manchester Arena in the United Kingdom, during Ariana Grande's music concert. This particular event killed 22 innocent people, including many children and young adults, as well as injuring over 800 people ([133],[80]). As stated earlier, terror attacks do not only effect developed countries. Many non-developed countries are also subject to terror attacks ([131]); however, they are often less reported in the media. In Kajuru, Nigeria on the 11th February in 2019, 141 innocent civilians were killed following an attack on an Adara settlement. The frequency of terror attacks in recent years have elevated concerns of attacks on personal and national safety across the globe.

Physical acts of terror are not the only threat that people, businesses or governments need to be protected from. As technological advances are developing each year, threats and attacks of a cyber nature are becoming increasingly more prevalent. With more and more information being stored online from bank details to addresses, this type of information is attractive to cyber-attackers and hackers. Not only is public personal data at risk from being used in online scams, much more delicate technological information is at risk from being taken. Military strategies and weaponry information/deployment are often coordinated and controlled by computer programs. Because of attackers who aim to steal extremely sensitive, confidential and powerful information, it is vital that this information and computer code are protected.

To ensure the physical safety of populations, parliaments and data, nations must increase their security efforts to prevent and catch any potential attacks. This includes a number of preventative interventions including technology detectors in airports for explosive devices to advanced coding and online data protection. Physical protection of patrolling security personnel often increases the feel of protection for the public as well as forming a physical barrier for terrorists to contend with.

Safety and security are a crucial part of infrastructure to enable the protection of all humans and their information and valuables. When security is mentioned, the usual thought is the policing and protection of human beings, such as the security of concerts, festivals and conventions. However, security can be applied to multiple scenarios that do not just relate to the protection of human life. These strategies can be used to monitor and protect wildlife, particularly endangered species in regions that are susceptible to poachers. These situations may use either stationary or mobile camera monitoring systems and alarms to signify when, for example, poachers enter a particular area, alerting rangers to the area of potential danger. Alongside these uses, security and policing strategies can be used to monitor and protect buildings of interest such as museums which contain objects of historic and monetary value. These scenarios will also use monitoring systems and motion detectors to alert security of any adversaries entering the building.

These situations can be modelled using mathematical models to provide us with the most optimal methods of patrolling considering a number of parameters. A real-life situation that has been implemented in Los Angeles airport (LAX), is the Assistant for Randomised Monitoring over Routes (ARMOR). The three main important characteristics of ARMOR are that it: 1) can provide quality guarantees in randomisation by appropriately weighing the damages and benefits of multiple options i.e. choosing to protect fatalities over economic damage, 2) addresses the uncertainty and limited information that policing and security departments have on potential adversaries and 3) enables a mixed-initiative interaction with potential users as ARMOR may be unaware of users' real-world constraints and hence users must be able to contribute to the schedule ([112]). Following a successful six month trial of ARMOR at LAX, it was permanently installed and is still used to this present day ([128]).

1.2 Overview of thesis content

In section 1.3, we provide a literature review which covers an overview of work related to allocating a moving resource (often called a searcher) in an effort to find other entities which may or may not want to be found. The literature review contains works distantly and closely related to patrolling games, which this thesis focuses on, but all of which have security applications in which similar techniques and strategies are used to allocate a model's resources. In all cases the main aim is to solve the game by finding optimal strategies and the optimal value.

Concluding the literature review we provide a short explanation of what a patrolling game is and why the game was developed. Following this, chapter 2 provides a detailed mathematical background (using some new notation) to the work seen in [16]. In this chapter we define the patrolling game $G(Q, T, m)$, which is a two-player, zero-sum, simultaneous game with a mobile patroller moving around a graph to find an immobile attacker. In particular we define pure and mixed strategies for both players and the payoff using such strategies result in. We see why the game has an optimal value (and Nash equilibrium) when mixed strategies are included and state methods that can be used to solve patrolling games. For developed strategies we state their performance and conclude chapter 2 by stating for which classes (sets of parameters) the patrolling game $G(Q, T, m)$ has been solved. For a particular strategy called the diametric attacker strategy we show that the lemma stated in [16] is incorrect. We find the performance of the diametric strategy, and thus correct the lemma, in chapter 3.

In chapter 3 we provide some results which ease the computation needed for finding the performance of an attacker strategy by reducing the set of pure patroller strategies which are considered. We then develop some more generalised strategies for both players by extending some previously studied strategies. Of particular importance is the development of a random minimal full-node cycle strategy $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ which provides a 'Hamiltonian like' bound for non-Hamiltonian graphs by visiting nodes at constant time intervals. We then look precisely at the performance of the this strategy at different nodes to find when the strategy performs weakly. This is followed by the development of the patrol improvement program(PIP) which can be used to improve a patroller strategy given a finite set of other patroller strategies. Chapter 3 is then concluded by using our developed techniques and strategies to solve the patrolling game $G(Q, T, m)$ when Q is a k-partite graph, focusing mainly on the case that the graph has all possible edges (complete k-partite graph), but with acknowledgements to when our strategies are optimal when only certain edges are present in the graph.

In chapter 4 we apply our techniques and strategies seen in chapter 3 to patrolling games when the graph is an extended star graph. Namely, we study the patrolling game when the graph is the elongated star graph S_n^k which is constructed by placing a star graph S_{n-1} at the end of a line graph L_{k+2}. This graph in the patrolling game means we can consider how a border with a location at one end should be patrolled. In doing so we develop bespoke strategies that are only applicable for the elongated star graph. A more generalised star graph is then studied, in which multiple borders are connected by a single central hub location, for which we find that the random minimal full-node cycle strategy $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ is optimal for a certain range of attack lengths. In addition the generalised star graph is used to model a central hub with cities at various distances away from the hub. However we find that it is better to model distances as later done in chapter 6 rather than use such a patrolling game with a generalised star graph.

In chapter 5 we consider patrolling games on any tree Q, meaning bespoke strategies cannot be created. We state optimal strategies for the game $G(Q,T,m)$ when $m=2$ and conjecture about the optimality of the random minimal full-node cycle strategy $\boldsymbol{\pi}^Q_{\text{RMFNC}}$, providing intuitive reasoning and computer testing as evidence that our conjecture holds. In addition, we mention under what conditions it is possible to adapt the solutions when the graph Q is not a tree.

In Chapter 6 we do not study the 'classic' patrolling game $G(Q,T,m)$, but we extend the patrolling game in order to model three different scenarios, further considering four payoff variations for one of the scenarios. Namely we add distances onto edges in $G(Q,D,T,m)$, we allow for vulnerable locations to be modelled by making one parameter depend on the node in $G(Q,T,\boldsymbol{m})$ and finally we provide four multiplayer variants where we there are k patrollers and l attackers in $G_i(k,l,Q,T,m)$ for $i=1,2,3,4$. For each extension we provide some general results and solve some particular classes of games by utilising work done for the classic patrolling game. Importantly for games with multiple attackers, we show that it is possible to reduce the problem of finding the collective attacker strategy to finding individual strategies for three of the four extensions.

The main findings of this thesis are concluded in chapter 7, which discusses the implications of our significant results, as well as suggesting areas in which future work could be carried out.

1.3 Literature review

While the main focus of this thesis is the patrolling game $G(Q,T,m)$ (as in [16]), we provide an overview of the work done on allocating an agent in a security setting when searching is involved. The following provides an overview of search theory and search games, including some variants, before reviewing applications and finally concluding with a recent variant, the patrolling game, which forms the main focus of this thesis.

1.3.1 Search theory

Early work considering the allocation of a searching agent focused on the problem of finding the optimal probability distribution, of effort, at locations in order to detect a target ([127]). A Bayesian approach was used, assuming a prior distribution for the list of locations of the target is known to the agent along with some conditional detection probabilities for detecting the target given the level of effort. This problem is one-sided as only the agent is allowed to choose how to allocate effort. Optimal distributions of effort against a stationary target are known when the conditional probability distribution for detecting a target is exponential ([86]). The exponential condition is generalised allowing for any stationary non-decreasing function of conditional probability of detection ([44]), where stationary was defined to mean that the density function is a function of the total effort level, regardless of how it is applied. The difference between stationary and non-stationary conditional dectection probabilities, when the serach is split into multiple parts is also discussed. This gives rise to solutions for maximal performance for a given effort, when the conditional detection probability is a function of the density of effort applied in that part, independent of that applied in previous parts ([49]). The problem was also reformulated as a convex programming optimization ([35]). The related problem, of minimizing the effort required to obtain a chosen performance level (the probability of the searcher finding the target) was solved ([43]). Necessary and sufficient conditions on the existence of solutions were found for the optimization problems considered above ([147]).

While the above work focuses on a continuous distribution for the target's location, work was also done with discrete distributions. In most discrete problems the objective is one of minimizing the time or cost to find the target, requiring the optimal solution to be function of time, as well as a function of position. The minimum time to find a target in one of n boxes is solved when the probability of the target appearing in the i^{th} box is known, with the time of its appearance being uniform over a large interval ([29]). A modified version of this game was studied in which objects arrive randomly to boxes, according to a Poisson process, with the objective now to maximize the gain, an non-increasing function of time and detection combined with the cost of scanning a box ([30]). Later the cost of switching between two boxes is studied ([69]) and the idea of a cost related to the distance between boxes is introduced ([72]). The idea of false detections are also considered in certain setups ([114]). While it not necessary for problems to have a periodic solution, it may be more applicable to real world applications and hence has been studied, with conditions for the existence of a periodic solution to the minimum cost problem being found ([98]). Rather than considering boxes in a line, it is possible to model a continuous line in which the objective is to minimize the expected distance traveled in order to find the target ([57]). Applications of the above results can be seen applied to communication synchronization ([73]), machine testing ([56],[71]) and locating lost satellites ([116]), which involved a two-stage search procedure.

While so far all work in search theory has considered an immobile target, it was seen to be of great interest to make the target mobile. A simple model for a moving target is one who moves in a Markovian fashion between boxes. In the special case of two boxes an optimal solution was given, and an approximate solution in the general case was also stated ([115]). From the solution to the two box case, it was clear that solutions to extensions would be difficult. So an extension into a continuous time model was done, finding the optimal solution for two boxes which seemed easier to generalise to n boxes ([50]). This can also be modeled as a partially observable Markov decision process over a finite time horizon ([126]). The FAB (Forward And Backward) algorithm was developed to produce search strategies which are at least as good as their predecessor, in turn giving necessary and sufficent conditions for optimality ([33],[141]).

An optimal search on a continuous, possibly unbounded, line region was suggested in [28], called the linear search problem. This problem has a mobile searcher and an immobile target, who is distributed along the line according to a probability distribution. While the general problem remains unsolved, for a discrete distribution dynamic programming can be used to produce a solution ([34]). An approximate solution can be found for any probability distribution to any desired accuracy level, at the cost of increasing computational time ([13]).

The above considers a moving target in a one-sided search, when only the agent is searching for a target and the target is not controlled by a malicious entity but is moving according to nature. However in a (two-sided) search game, the use of game theory becomes relevant, as in addition to a player controlling the agent, another player controls the target. In such setups both players are considered to be intelligent, and so this is much more useful in modelling military security. As the origins of search theory began during World War II in the Navy's Anti-Submarine Warfare Operations Research Group(ASWORG), the two-sided search game provides a better model against an equally considerate enemy.

1.3.2 Search game

The search game was initially introduced in [78], developed in [60] and updated in [13]. A search game is a two-player, zero-sum game which is characterised by a search space Q. The search space is assumed to be either a compact Euclidean space or an unbounded connected set, with an area μ. The searcher usually starts from a specified point called the origin O , but choosing the starting point is considered in some works. The searcher chooses some trajectory S inside the search space, with a velocity, usually constrained to be 1 under a normalization. The hider can be immobile or mobile, choosing a location or trajectory H. Neither player has any information about the other until they are within a capture radius r, of each other in which case the searcher captures the hider. For one-dimensional sets (such as graphs) this radius is usually considered to be zero. Search and hider trajectories (or simply a location for an immobile hider) give a payoff function $p(S, H)$, equivalent to the time taken for the searcher, using S, to find the hider, using H. Mixed strategies consists of players choosing a randomization among all possible trajectories (or locations for a immobile hider), denoted s and h for the searcher and hider respectively. So the payoff/expected search time becomes

$$p(s,h) = \int p(S,H)d(s \times h).$$

The value of the game V, is the minimum search time required for a searcher to find a hider, and its existence is given in [60] (and [12]).

Considered first was an immobile hider. In any two-dimensional convex region (a weaker condition is available in [60]) the searcher has to find a closed curve that covers all points of the region, in the sense that all points in the regions are within the capture radius r at some point along the curve. The length of this covering curve can be found to be less than $(1+\varepsilon)\frac{\mu}{2r}$ where $\varepsilon \to 0$ as $r \to 0$. Therefore by encircling this curve equiprobably in each direction the searcher ensures an expected search time of roughly $\frac{\mu}{4r}$. By using a uniform hiding distribution the hider can achieve the same exptected search time. Hence the value is asymptotic to a quarter of the ratio between the area and capture radius, i.e. $V \sim \frac{\mu}{4r}$ ([60]).

Linking with the linear search problem ([28]), the linear search game is considered, which deals with a region which is unbounded ([26]). To have a meaningful game, they require the hider's location distribution to be limited to λ from the origin. They assume the game is sequential with the hider picking a strategy to maximize the search time followed by a searcher picking a strategy to minimize it, called a minimax game. They show that the minimax trajectory is to oscillate around the origin, guaranteeing capture by at most a search time of 9λ. However the optimal search does not depend on the hider's restriction of λ ([26]). An alternative approach was used, without the restriction on the hider, but with a payoff that is normalized by the distance between the hider and the origin ([58]). This is now commonly refered to as the competitive ratio ([31]).

The idea of a minimax solution is often used for unbounded domains. A minimax search trajectory for a homogeneous unimodal hider function is always a geometric sequence for discrete problems and an expoential function for continuous problems ([63]). This result was developed into a general tool. In order to find the minimax trajectory one need only minimize a simple function over a single parameter (the generator of the sequence) instead of the entire trajectory space ([60]). Geometric trajectories are useful in other fields in order to obtain effective algorithms, such as, bidding, minimum latency tours, scheduling and clustering ([38]). In these areas a 'doubling' method is used, essentially using geometrically increasing estimates on an optimal solution in order to produce fragments of the algorithm's optimal solution ([37]). We note that we use a 'doubling' idea in chapter 4, section 4.2.10, in order to adapt strategies.

An important extension to the linear search game is the star search game, a problem on $M \geq 2$ unbounded rays. A periodic trajectory, visiting every ray every M^{th} time, with a montone-increasing step size was found to be optimal ([58]) using the general tools for minimax search trajectories ([60]). Contrasting the pure strategy case, as the optimality proof applied only to strategies which use periodic and monotone trajectories ([81]). Further extensions have been consider such as a bound on the distance from the origin λ ([96]) and when there are mutliple searchers ([97]). These extensions find the competitive ratio for the game.

When the search space Q is a finite connected graph with edges being of a prescribed length, the area μ is the total length of the distances associated with edges. A pure strategy for the searcher is a walk in the graph $S = S(t)$, satisfying a unit speed condition. The search time is $p(S, H) = \min\{t \mid S(t) = H\}$. It is easy to see by the hider choosing a location uniformly h_{Uni} that the value of the game $V \geq \frac{\mu}{2}$. This follows because any walk $S(t)$ has a unit discovery rate so $Pr(p(S, h_{\text{Uni}}) \geq t) \geq 1 - \frac{t}{\mu}$ and thus $E[p(S, h_{\text{Uni}})] \geq \frac{\mu}{2}$ ([60]).

When considering a fixed starting position for the searcher, a natural way to search the graph is to use a Chinese Postman Tour(CPT) ([54]), that is a closed walk which visits all edges with minimal length. Immediately if the graph Q is Eulerian the walk can choose any Eulerian tour $S_{\text{Eul}}(t)$ for $0 \leq t \leq \mu$, using the forward direction and backwards direction with equal probability (backwards tour is $S_{\text{Eul}}(\mu - t)$ for $0 \leq t \leq \mu$). Following this strategy ensures $V \leq \frac{\mu}{2}$ and hence it follows that $V = \frac{\mu}{2}$ for all Euluerian graphs. Furthermore this only holds for Eulerian graphs ([60]). For non-Euluerian graphs the strategy uses the random CPT, with such a strategy being optimal if and only if the graph is weakly Eulerian (a tree-like graph connecting Eulerian subgraphs) ([62]). In which case the optimal hider strategy is constructed by a recursive algorithm. When a graph is not weakly Eulerian the solution becomes very complicated ([109]). In general finding the optimal solution to the search game on a graph with an immobile is NP-hard ([138]). However if the search time is limited, logarithmically to the number of nodes, the optimal strategy can be found in polynomial time. The game can also be formulated as an infinite dimension linear program, so approximate solutions can be found ([18]).

While the search game assumes players have a knowledge of the graph, it is not necessary, but such a lack of knowledge costs the searcher a great deal of optimality. A normal depth-first method, commonly used in computer-science, was used and mistakenly claimed to be guaranteed to have a search time of μ ([17]). It was later found to actually have a guaranteed search time of 2μ ([64]), and then under a randomization of the depth-first search was able to achieve the original guaranteed search time for the searcher of μ ([64]), showing the importance of a correct randomization for mixed strategies.

Work on an arbitrary starting position has been done, using a walk related to the CPT, called the Chinese Postman Path(CPP), which visits all nodes of the graph Q and has minimal length. A 'Simple' strategy was developed by using the CPP with the searcher starting at each end with equal probability and then going to the other end. If the CPP is a Eulerian path (a Eulerian tour without the requirement to be closed) then this 'Simple' strategy is optimal, giving an optimal search time of $\frac{\mu}{2}$ ([42]). While this case is easy to understand it turns out the 'Simple' startegy is optimal for trees with an arbitrary start point ([42]). This work was extended to have multiple Eulerian graphs connected by a tree, where 'Simple' is still optimal ([5]). In a contradiction to the to the result of the fixed start serach game, it has been shown that graphical characteristics for which ‘Simple’ is optimal are impossible to classify ([7]).

1.3.3 Serach game variants

Having a mobile hider has also been considered. The princess and monster game is considered the most general form of the limited information problem of a mobile hider (the princess) and a mobile searcher (the monster). Played in a search space both players move along continuous paths, with the monster capturing the princess if they get some fixed distance apart. The objective of the game for the monster is to minimize the time until capture and for the princess it is to maximize the time until capture, making the game zero-sum ([78]). Under the condition that the search space is convex an optimal solution was given and the optimal performance of the princess and monster game is related to the ratio between the size of the search space and the distance at which the monster catches the princess, for small distances ([59]). Later it was found that a much weaker condition was needed for optimality ([60]). The optimal strategy found for princess is interesting, as it has the princess go to a location, wait for a period of time which is not too short but not too long and then repeat this process. The corresponding optimal search strategy for the monster is to subdivide the region into many narrow rectangles, searching one and then after some time moving to another rectangle ([59],[13]). This strategy is not robust and an optimal strategy for the monster which is more robust was developed which has the monster ‘bounces’ between boundaries ([89]). Such a strategy is robust to partial information, as even if the princess’ position is known then the monster is unable to predict the princess’s course for very long. However such a strategy is only effective in convex search region. The original strategy is effective for non-convex regions and can be easily adapted to general problems, such as where the probability of detection depends on the distance between players ([65]).

While the above deals with the princess and monster game on a search space, it was also initially proposed to play the game on a circle ([78]). When the search region for the game is the discrete circle (cyclic graph) an optimal solution was found ([142]). This optimal strategy consisted of having the monster start at some point and move to its antipode (diametrically opposite) point, going clockwise or anticlockwise depending on a fair coin flip, repeating this process (with a small modification if the number of nodes was odd). When using the continuous circumference of the circle as the search region (as initialy proposed in [78]) the same nature of an optimal solution was found ([148],[1]). With a fixed position for the monster, the hider can get an advantage by staying at the antipode for a length of time ([1]). Aside from the circle, solving the princess and monster problem on a graph is considered difficult. Even on the line the optimality of the game remains an open problem. Intuitively one may believe that the monster starting at one end and moving to the other end is optimal, however this is not true. In response to such a strategy the princess may start a distance from a random end and move to the other, leading to 0.75 expected search time on the unit interval. However the patroller can do better and the value of the game is approximately 0.7 ([10]). We note that the complexity of a game on a line is true holds true for patrolling games, as the solution to a line requires careful study and bespoke solutions ([107]).

While the search game considers a minimizing searcher and maximizing target/hider, aiming to affect the expected search time, the rendezvous game, has the target no longer hiding and now also aiming to minimize the search time. Thus in the rendezvous game there are two searchers attempting to minimize the search time. An example is two friends seeking to find each other in a shopping mall, but the space need not be physical, as we can also imagine two agents communicating with walkie-talkies consisting of several channels ([13]). This problem was first studied on discrete locations, which found an optimal solution when there are only two or three locations and for more locations only a bound on the value V was given for the strategy ([19]). The problem was later studied in the continuous location case which gave the foundations for the theory of rendezvous games ([3]). An important aspect regarding the cooperation of the two searchers is discussed and how they may be restricted to use the same search strategy. Such rendezvous games where players must use the same strategy are called symmetric, otherwise they are called asymmetric.

The rendezvous problem on the line has been thoroughly studied. In the asymmetric version the 'wait for mummy' strategy (where one player waits at their location and the other seeks them), a reduction to the linear search problem, is never optimal ([61]). Work on the symmetric version was also done, providing improvements on the bounds ([23]). Along with the line, as is common with search theory, the game was studied for graphs. Work on graphs extended the work done for the rendezvous on discrete locations. In the asymmetric version the 'wait for mummy' strategy is optimal for Hamiltonian graphs. In the symmetric version, bounds are found when considering a restriction to Markovian strategies ([6]). Deterministic algorithms have been determined to solve such graphical problems ([46]). A similar, but simpler, problem on a labelled graph has been solved in the symmetric version with solutions being much easier as they have a reduced selection of strategies ([4]).

A variation on the search game, known as the accumulation game, has a hider place objects at locations with the searcher choosing locations to search in order to find these objects. The hider wins if at any point in the time-horizon there is a critical mass of objects hidden. The accumulation games was first studied in the discrete case with discrete time, for which an algorithmic approach to calculate the value of the game is given ([83]). The accumulation game was later studied with continuous time, with objects assumed continuous, and in this case a condition for which the hider can always win by a uniform strategy is given ([84]). Further work looked at the accumulation game on a hypergraph structure, which has a searcher choosing groups of nodes, unlike the previous accumulation games, which are only optimization problems for the hider, this graph extension forms a true game where both players have to act strategically ([8]). The idea of having discrete locations with capacities, limiting the amount of material that can be stored at them was studied. Finding optimal searching strategies when the payoff for the patroller is determined by the amount of material found ([149]). We note that the last game defines itself from a similar game known as the (search and) ambush game.

The ambush game involves a searcher using an ambush position or distribution plan of barriers to capture a moving evader (target). The ambush game was fist studied on a lattice, for which optimal solutions were achieved when limitations on players were imposed ([120]). Work for the ambush game then considered a continuous region, were the evader's strategy is considered to be a continuous function, and subequently found a near optimal strategy for the evader which was developed algorithmically ([118],[121]). The techniques developed allow for a similar search game in the square, with limitations on the target's mobility and a square capture radius for the seracher, to be solved ([121]). Other models involving ambushing behaviour have been used, but in different contexts. The 'princess and monster' game is extended to include an additional mode of searching for the monster in which they are allowed to ambush. The ambushing mode allows for the capture of a moving hider and it was found the frequency at which the ambush mode is used must increase with time ([11]). Further work was done when the searcher is noisy and passes on some knowledge of their movements to the hider. This noisy ambush game was shown to have an expected search time of $\frac{2}{3}$ ([9]). As is common for search games, this result was previously conjectured (Conjecture 3.7 in [132]) but was unable to be shown without some new techniques.

The vast majority of work done in the field of search theory has security applications. In an effort to study how nations may conceal violations of arms treaties the inspection game was proposed ([51]). At each stage in the game, the inspectee has a chance to violate the treaty or comply, while the inspector dispatches inspection teams. Rewards are gained at each stage of the game, and forms a sequential two-player zero-sum game. Additional features were later incorporated into the model to allow for imperfect detection ([122]) and incomplete information ([77]). The model was also studied with more inspectees ([75]), as studies on the inspection game are mainly focused on finding an effective strategy for the inspector ([21]). An offshoot of inspection games are smuggling games, in which effective ways to shut down the flow of drugs into a country was analysed ([130],[24]). Consideration on the amount of contraband being smuggled was consider to see the effect on the smugglers decisions ([76]). It is worth noting that inspection/smuggling games are special search games as for the most part they have binary decisions for players.

Another class of two-player games with security applications are evasion-search games, a more general case of the princess and monster games. Initially modelled as an evader who moves along a line to reach a target point while knowing the searcher's position, no optimal solutions were achieved, but focused on improving the evader's motion ([101]). Another model has an evader moving between cells starting at one end of the line, with shelter at the other. The evader is able to move to a few neighbouring cells and the searcher knows the current evader position, with the payoff being the number of detections until the evader reaches their destination ([91],[92]). Similar evasion-search games with detection probabilities and targets for the evader have been studied ([103],[104]). On discrete cells dynamic programming was often powerful enough to derive equilibrium by recursion. When the evader has a goal, the game becomes one about a passage through a region. Initially used to model an evading boat seeking a target and searching bomber who knows the current position but does not know where the boat will be when the bomb lands, with the probability of a bomb hitting a ship being the payoff ([52]). Such games are often referred to as infiltration games. With the importance to the ASWORG, multiple researchers studied the problem with varying considerations about the bombs ([36],[90]) and the speed at which players move ([66]). Evasion is a feature of many other models which differ in payoff and information. Such as a payoff linked to the travelling cost with play continuing until detection for a given time-horizon ([140]). A multi-stage game where reward is determined at each stage by player positions, called the cumulative search-evasion game was studied ([53]). A game with no information between players and probabilistic movement on a cyclic graph has been studied ([119]). The probabilistic movement idea was used in other models ([41]).

The infiltration idea is of key importance to organisations which have to protect a key resource which is vulnerable to attack. The general game was introduced in [60] and has an immaterial seracher and an infiltrator moving on a graph. Initial work studied and solved the infiltration game for a single arc (line) ([88]) and following work solved the game for any number of arcs (star) ([20]) using the same core idea for player strategies. These ideas were generalised for arbitrary graphs ([2]). A generalisation involving the speed for the result on any number of arcs was also done ([25]). A simplification to the searchers strategy was found, which drastically reduces the number of potential pure strategies required ([67]).

We finish our review of search games to now look at various applications of these types of games which have not already been mentioned, along with some implementations of theory. Afterwords we will look at our final variant of the search game, the patrolling game $G(Q, T, m)$, which this thesis focuses on.

1.3.4 Applications

Beginning in World War II, search theory has developed to cover a vast array of problems relating to military and security issues where a search is required. Mainly focused on boats and submarines, implementation was hidden due to security concerns. In more recent years however, with the growing computer literacy and power, it is becoming more widely implemented. We first look at the suggested areas for the application of search theory before looking at actual implementations and some real life issues that arise.

Theoretical models for locating sunk ships, detecting submarines, bombing ships are some initial motivations for studying search theory in a military setting. Further security issues can be modelled, such as inspecting nations for nuclear arms, custom officers and bag searching. Search theory can be also used in the predator prey dynamic to explain animal behaviour. Some additional models using search theory follow.

In cyber security it has been applied to sample data packets with a given budget to detect malicious intruders ([85]). In terror protection the study of how to minimize the damage inflicted on an electrical grid has been studied ([123]). Further to this the protection of road traffic ([27]) and rail traffic ([110]) was studied to see how to minimize the effectiveness of a terrorist. The development of a model where the defender fortifies facilities which may be attacked by a terrorists was done using, as with most security applications, a Stackelberg game ([124]). Similarly a model to detect damage inflicted by a terror attack was developed ([22]). Search theory is also applied in non security applications, such as finance, where it is often applied to labour markets to study unemployment and goods markets ([102]).

While a lot of work deals with theoretical applications of search theory, the implementation of theory along with real life limiations is of key concern for security experts. An early implementation was for coast guard patrols in search and rescue operations, in CASP (Computer-Assisted Search Planning) ([117]). Using Bayesian updating and Monte Carlo simulations for original probabilities this software allowed for the planning of search routes which once complete would update and inform the users of the next search route.

A more recent and prominent real world application is that of ARMOR (Assistant for Randomized Monitoring over Routes), which was software developed for the patrol routes of various units in Los Angeles International Airport (LAX) ([112]). ARMOR was applied to LAX security in 2007. Their patrol and monitoring problem was modelled as a Bayesian Stackelberg game with the fastest known heuristic algorithm, DOBSS (Decomposed Optimal Bayesian Stackelberg Solver), applied ([108]). Their models allows for a variety of actions for LAX security and for any number of adversary types each with their own variety of actions. To allow for real life situations which are not predicted, the software allows for occasional use adjustments or overrides based on local constraints, alerting the user if there appears to be a degradation in the overall performance.

Following ARMOR, a similar application was studied in patrol routes for trains by the use of TRUSTS (Tactical Randomization for Urban Security inTransit Systems) in Los Angeles ([146]). Aimed mainly at fare-checking, the patrol units are present to avoid fare evasion. Before TRUST human schedulers were used, but were found to be incredibly poor at generating unpredictable schedules ([139],[129]), therefore the model requires employable strategies to require less cognitive strain on the patrol units. The idea of Markovian strategies were seen to be within 99% of the linear program upper bound ([79]). One of the main differences in the model used in TRUSTS is that the threat of fare evasion from the general populous is much higher than models used for counter-terrorism. Its implementation to criminal and counter-terrorism was later done to allow the use of TRUST to compute the best for each three scenarios and then combining these ([45]). More theoretical work has been done in areas related to fare evasion, such removing the modelling assumption that train users follow a fixed route through the graph ([40]).

Another use of software in allocating patrol units is IRIS (Intelligent Randomization In Scheduling), which is used to help federal air marshals provide protection on commercial flights in the U.S ([134]). This software has to cope with a larger search domain than that of ARMOR and the previous algorithm at ARMOR's heart (DOBSS) is unable to cope ([82]). Like the previous algorithm a mixed-integer linear program, called ERASER-C ((Efficient Randomized Allocation of SEcurity Resources - Constrained), is used. ERASER-C's additional insight is that the payoff of these games depend on whether or not the attacked target is in a given region. This allows for the solution to be implemented in a reasonable time frame. The tests on real life data seem to indicate IRIS's usefulness and as of 2008 it was under review for implementation.

Still related to airport security software, GUARDS (Game Theoretic Security Allocation on a National Scale Categories and Subject Descriptors) was developed to aid in the deployment of TSA agents ([113]) to various airports. Two unique challenges arise when developing such software, no centralized planning agency can produce an optimal strategy for all airports and wanting to provide a common standard for security among airports. After modeling considerations DOBSS ([108]) was deemed applicable and so, as in ARMOR, forms the heart of GUARDS solution. Numerical analysis on simulated data found that circumvention is important against an intelligent attacker.

A system for allocating port patrols, called PROTECT (Port Resilience Operational/Tactical Enforcement to Combat Terrorism), was developed for the coast guard in the port of Boston ([125]). PROTECT drops the assumption of the adversary being perfect rational and relies on bounded adversary responses, called quantal responses ([100]), which have seen benefits in applications to secuirty games ([144]). This quantal response allows for the use of an algorithm called PASAQ (Piecewise linear Approximation of optimal Strategy Against Quantal response) ([145]). Numerical analysis on real world simulations shows PROTECT visits higher valued targets consistently. It has been implemented in the port of Boston and with its success has also been implemented in the port of New York.

After the successful implementation in transportation related infrastructures security games have been applied to green security, such as protecting against overfishing ([74]) and protecting wildlife from poaching ([143]). Applying security games to green security provides multiple challenges, the particular challenge is that there are multiple adversaries who attack frequently but do not conduct extensive surveillance spending less time and effort in each attack. These initial models and their implementation modelled the green security problems as a game with multiple rounds, with each round being a security game. Three limitations of this modelling are; the full-observation of the defenders strategy by the attackers; the lag between observation and execution; and the use of Maximum Likelihood Estimation (MLE) to learn paramters. The use of the MLE for individual attackers has been shown to lead to skewed results and therefore a new model of green security games was developed to overcome these limitations. Generalised green security games, with general parameters, were studied and algorithms for such games were also developed ([55]).

1.3.5 Patrolling games and problems

Having given an overview of work done in the research field of search games, we now look at the main search game studied in this thesis, the *patrolling game* as defined in [16]. The patrolling game $G(Q, T, m)$ is a search game with an immobile hider on a graph. In the patrolling game we will refer to the searcher as the patroller and the hider as the attacker. Unlike search games however, the attacker not only chooses a location to hide but also a time at which to do so, called the commencement time τ. This change will require a utilisation of the times at which the attacker should commence their attacks. Along with this, the attacker is only at their chosen node for a period of time, called the attack length m, and if they are not found within this time window they win. Again this is unlike the majority of search games which are games of degree (meaning there are levels of success, using terminology from [78]) the patrolling game is a game of type (meaning the outcome is binary and players either win or lose). Unlike a search game's hider, the patrolling game's attacker more accurately models a terrorist. This is because they are able to commence their attack at any point in time and only need to remain undetected for a certain period of time before detonating their bomb. Another application could be seen in cyber security where a hacker attempting to destroy a network of computers need only have his virus remain undetected for a certain period of time before it is able to complete its task to destroy or harvest data from the network.

The initial paper ([16]) on patrolling games gave properties on the value of the game, reduction techniques and generic strategies for both players. With these strategies the solution to patrolling games for various graphs such as Hamiltonian and complete bipartite were given. In addition partial solutions for the patrolling game on the line graph were given. These results were given for certain ranges of the attack length $m \geq n-1$, where n is the number of nodes in the line graph. Later solutions for other attack lengths of $m < n-1$ were found in [107]. However the development of such bespoke strategies required much careful consideration with a heavy dependency on the attack length m.

In a similar regard work was done on the patrolling game in continuous time, with continuous space on the line. Analogous results to that of the original patrolling game on a line graph where found, with the patroller adopting the same core strategy and resulting in the same value for the game ([14]). Extensions to this work found optimal solutions for all Eulerian networks and studied the patrolling game on $\mathbb{R}^2$ asymptotically as the capture radius goes to zero ([68]).

A peroidic version of the patrolling game is also defined in [16], which forces the patroller to return to the starting location at the end of the game. Results are adapted from those for the original patrolling game and have later been improved upon to solve the periodic patrolling game on the line with some notion as to how it can apply to generic graphs ([15]).

Alongside the work on the patrolling game, the patrolling problem, a one-sided version of the game, was introduced. It assumed that the attacker attacks randomly in time, according a Poisson process, which they can manipulate the rate of ([94]). The idea of Whittle indices ([70]) and their generated policies are used in order to obtain heuristics for the problem. A further heuristic can then be applied to make a matrix/normal form game , which resembles the patrolling game (albeit with strategies removed) and which can be solved by a linear program. Further work extends the patrolling problem to allow for overlook, which required a different formulation of the state space upon which indices are developed ([95]). The idea to involve distances and speed as a way to extend the model are done to the original patrol problem without overlook ([99]). Work was also done on how to effectively utilise multiple patrollers ([99]).

1.3.6 Conclusion of literature review

In this literature review, we have provided an overview of search theory, including the one-sided search problem and the two-sided search game. We have also seen the vast majority of variants studied in the literature. We have seen that different structures in the type of search region, payoff, mobility make the most difference in how problems and games are searched. We have also noted that in general analytic optimal strategies either require the use of bespoke strategies arising from careful consideration of the game and its mechanisms. While computational methods have been developed and showcased to find optimal solutions it is infeasible to do so for large strategy spaces (according from a moderate graph or game length) and in this situation the best practical approach is often to find near optimal strategies. Our work leans towards the theoretical side and so finding strategies can only be done by considering known strategies and the careful construction of new bespoke ones.

Lastly we have given a recent account of patrolling games and patrolling problems with their currently solved extensions. In chapter 2 we provide a detailed background on patrolling games $G(Q, T, m)$, stating strategies and how to find optimal strategies. In addition, we will state the performance of known strategies and which classes of patrolling games have already been solved.

Chapter 2

Background on patrolling games

2.1 Introduction

In this chapter we explore a model for a game in which a patrolling entity, such as that of a museum guard, traverses between a set of locations in order to locate and thwart an attacking entity, such as a thief. We take the recent model propsed in [16], which is a game theoretic model for such a scenario. This model is a variant of the search game known as the *patrolling game*. In the patrolling game a searcher, called a *patroller*, wishes to find a target who is immobile, called an *attacker*. The patrolling game uses a graph structure for the search space and is played in discrete time. In each unit time the patroller either remains at the current node or moves to an adjacent node on the graph. This is done in an effort to locate the attacker. Unlike most variants of the search game, which have the target in the system from the start of the game until they are found, in the patrolling game the attacker is only at a chosen node for a certain length of time. In addition the attacker need not be at their chosen node initially and may choose some time at which to commence their attack. This results in the patroller only being able to find the attacker during a window of opportunity at their chosen node. If the patroller is at the same node as the attacker chose during this window of opportunity, then we say the patroller catches the attacker and so wins the game. Otherwise if the attacker is not caught during the window of opportunity they win the game. Therefore unlike most search games, where the objective is to the minimize the time until the target is found, in the patrolling game the goal is just to catch the attacker. Due to this, the patrolling game is a win-lose game.

This chapter is structured as follows, we begin by explaining the mathematical set-up of the pure version of the patrolling game in section 2.2.1. We then discuss why playing pure strategies with probabilities (mixed strategies) are studied for the patrolling game in section 2.2.2. This leads to a game in which we seek the highest probability of the patroller catching the attacker, called the game's *value*. This is followed by a survey of known results developed in the initial paper ([16]) in section 2.3.1. Known strategies and their lower and upper bounds on the value of the game, for both the patroller and attacker, are given in sections 2.3.2 and 2.3.3 respectively. We conclude these known results from the literature by stating known classes of the patrolling game which are solved in section 2.3.4. From the results for the line graph ([107]), we notice that the length of the window of opportunity is crucial in deciding which strategies are optimal for both the patroller and attacker.

2.2 Definition of patrolling games

2.2.1 Pure patrolling game definition

A pure patrolling game is a two-player, zero-sum game in which player one, henceforth called the *patroller*, attempts to catch player two, henceforth called the *attacker*. It models many situations such as the following:

- An officer patrolling an airport in search of a terrorist who is attempting to plant a bomb.
- A solider patrolling an occupied city with rebels propagating propaganda.
- Police units patrolling multiple districts in a city containing targets for thieves.

A pure patrolling game G is characterized by the following three parameters:

- An undirected *graph* $Q = (N, E)$, made up of a set of *nodes* N and a set of *edges* E. Nodes represent locations at which the patroller or attacker may be present at. Edges determine which locations are next to each other, called adjacent, allowing movement between locations.
- A *game length* $T \in \mathbb{N} \cup \{\infty\}$, representing how many units of time there are before the game is over. This forms the *time-horizon* for the game $\mathcal{J} = \{0, ..., T-1\}$.
- An *attack length* $m \in \mathbb{N} \cup \{\infty\}$, representing how many units of time the attacker must be present in the graph in order to complete their attack. This is the length of the *attack interval*.

Therefore a pure patrolling game G is written as a 3-tuple $G(Q, T, m)$ in which Q is an arbitrary graph and $T, m \in \mathbb{N} \cup \{\infty\}$. We note that for a game length $T = \infty$ the corresponding time-horizon $\mathcal{J} = \mathbb{N} \cup \{0\} \equiv N_0$. In the game $G(Q, T, m)$ the pure strategies for players are as follows:

- The patroller chooses a *pure patroller strategy* $W : \mathcal{J} \to N$, such that for all $0 \leq t \leq T-2$, $W(t) = W(t+1)$ or $(W(t), W(t+1)) \in E$. So the pure patroller strategy W is a *walk* of length T. That is at $t = 0$, the patroller chooses a start node from N and between each subsequent time period either waits at the current node or moves along an edge to another node. The walks position at time t is $W(t)$ and the image $W(\mathcal{J}) \in N^T$ is vector which equivalently defines the walk.

- The attacker chooses an *attack node*, $j \in N$, and a *commencement time*, $\tau \in \mathcal{J}$, forming an *pure attack strategy* $a = (j, \tau)$. This means the attacker is at their chosen node j for each time $t \in I$, where $I = \{\tau, ..., \tau + m - 1\}$ is the *attack interval*.

For clarity we note that the original defintion of the patrolling game, as in [16], uses only the attack interval notation for which the commencement time is equivalent. As pure patrolling strategies are allowed to wait at nodes even without a loop and as multiple edges offer no benefit, we can immediately assume that the graph Q is a *simple graph*.

For the game $G(Q, T, m)$, all pure patroller strategies (walks) are collected in the set $\mathcal{W}(Q, T, m)$ and similarly all pure attacker strategies are collected in the set $\mathcal{A}(Q, T, m)$. The sets of pure strategies are dependent on the parameters Q, T and m, however when the game's parameters are clear we use $\mathcal{W} = \mathcal{W}(Q, T, m)$ and $\mathcal{A} = \mathcal{A}(Q, T, m)$, for the pure patroller and pure attacker sets respectively. In the case that N and T are finite, enumeration of all pure attacker strategies is easy as $\mathcal{A} = N \times \mathcal{J}$ and thus $|\mathcal{A}| = |N|T$. However, enumeration of all pure patroller strategies is not as easy due to the multitude of choices depending on the set of edges E, at most every node is adjacent and so we have $|\mathcal{W}| \leq |N|^T$ with the exact value being combinatorially complex. While we do not enumerate the set $\mathcal{W}$ we can choose an arbitrary bijection $\beta_1 : \mathcal{W} \to \{1, ..., |\mathcal{W}|\}$ to give an order to the pure patroller strategies so that $W_{(x)} = \beta_1^{-1}(x)$ for all $x \in \{1, ..., |\mathcal{W}|\}$. In addition we choose another arbitrary bijection $\beta_2 : \mathcal{A} \to \{1, ..., |\mathcal{A}|\}$, which orders the pure attacker strategies so that $a_{(y)} = \beta_2^{-1}(y)$ for all $y \in \{1, ..., |\mathcal{A}|\}$. The bijections β_1 and β_2 allow us to write the pure patrolling game in normal/matrix form, once player payoffs are defined.

A variant of the pure patrolling game, known as the *periodic* pure patrolling game places a restriction on the walk performed by the patroller. In the periodic patrolling game, a single patrolling game, with a finite time-horizon ($T = k / \mathcal{J} = \{0, ..., k-1\}$), is repeated ad infinitum, to form a restricted infinite time-horizon patrolling game ($T = \infty / \mathcal{J} = \mathbb{N}_0$). We formally define this game as $G^p(Q, k, m)$, the periodic patrol game with period k, with the restriction that the patroller's walk must start and end at the same node (otherwise the game is not repeatable ad infinitum). This means in $G^p(Q, k, m)$, the set of patroller strategies (walks) is $\mathcal{W}^p = \{W \in \mathcal{W} \mid (W(k-1), W(0)) \in E\}$. Such a restriction to the periodic game may be important for modelling reasons. As an example, consider a police unit searching for criminal activity, which must return to a base location (which can be decided in the planning phase) after each shift, such that a new police unit can oversee the same patrol. In contrast the original patrolling game is called the *one-off* patrolling game in [16], however we will not use this terminology as the periodic patrolling game does not form the basis of our study. This restriction, to the periodic patrolling game, alleviates some of the issues with the combinatorial complexity of the patroller's strategy set in comparison to the (one-off) patrolling game.

With the patroller choosing $W \in \mathcal{W}$ and the attacker choosing $(j, \tau) \in \mathcal{A}$, a combination of strategies $(W, (j, \tau))$ is formed which decides who wins and who loses the patrolling game. The patroller wins the patrolling game if the attacker fails to complete their attack either by being caught by the patroller or running out of time. As the attacker is at node j from time τ onwards they are at the node for all times in the attack interval $I = \{\tau, ..., \tau + m - 1\}$ and so run out of time if and only if $\tau + m - 1 > T - 1$. It is immediately clear that patrolling games with $T < m$ have the attacker always run out of time, so we will now assume that $T \geq m$. Furthermore, the attacker is limited to choosing a commencement time $\tau \in \mathcal{T} = \{0, ..., T - m\} \subset \mathcal{J}$ so as to not run out of time, so with this $\mathcal{A} = N \times \mathcal{T}$. With this restriction the patroller wins if and only if $j \in W(I)$ and hence the *payoff* for the *patroller* playing $W \in \mathcal{W}$ against an attacker playing $(j, \tau) \in \mathcal{A}$ is

$$P_p(W, (j, \tau)) = \mathbb{I}_{\{j \in W(I)\}} = \begin{cases} 1 & \text{if } j \in W(I), \\ 0 & \text{if } j \notin W(I), \end{cases} \tag{2.1}$$

where $\mathbb{I}_{\{A\}}$ is the *indicator function* for event A. We know the attacker wins if and only if the patroller does not catch them and so the *payoff* for the *attacker* playing $(j, \tau) \in \mathcal{A}$ against a patroller playing $W \in \mathcal{W}$ is

$$P_a(W, (j, \tau)) = \mathbb{I}_{\{j \notin W(I)\}} = 1 - P_p(W, (j, \tau)). \tag{2.2}$$

From equation (2.2) it is clear the game is zero-sum (note that the sum of payoff's for any given combination is 1) and hence we can just use the patroller's payoff as in equation (2.1), dropping the subscript. With this we define the *pure payoff matrix*

$$\mathcal{P} = (P_p(W_{(x)}, a_{(y)}))_{x \in \{1, ..., |\mathcal{W}|\}, y \in \{1, ..., |\mathcal{A}|\}},$$

with a maximizing patroller and minimizing attacker. The order of play in the pure patroller game is important with the second player choosing knowing the first player's strategy choice, giving the first player the advantage. We say that the first player is the *leader* and the second player is the *follower*, with the follower choosing the best strategy against a known strategy choice for the leader. As the follower can respond with their best strategy, the evaluation of the leaders choice can be thought of as the worst-case scenario implementation.

We define the pure *MiniMax* patrolling game $G^{\triangle}(Q, T, m)$ in which the attacker is the leader and the patroller is the follower. The optimal value of the game $G^{\triangle}(Q, T, m)$ is

$$O^{\triangle}(Q, T, m) = \min_{a \in \mathcal{A}} \max_{W \in \mathcal{W}} P_p(W, (j, \tau)).$$

Optimal solutions to the game $G^{\triangle}(Q, T, m)$ are $W^{\triangle} \in \mathcal{W}$ and $a^{\triangle} \in \mathcal{A}$ such that $P_p(W^{\triangle}, a^{\triangle}) = O^{\triangle}(Q, T, m)$. Similarly we define the pure *MaxiMin* patrolling game $G^{\triangledown}(Q, T, m)$ in which the patroller is the leader and the attacker is the follower. The optimal value of the game $G^{\triangledown}(Q, T, m)$ is

$$O^{\triangledown}(Q, T, m) = \max_{W \in \mathcal{W}} \min_{a \in \mathcal{A}} P_p(W, (j, \tau)).$$

Optimal solutions to the game $G^{\triangledown}(Q,T,m)$ are $W^{\triangledown} \in \mathcal{W}$ and $a^{\triangledown} \in \mathcal{A}$ such that $P_p(W^{\triangledown}, a^{\triangledown}) = O^{\triangledown}(Q,T,m)$.

When $W^* \in \mathcal{W}$ and $a^* \in \mathcal{A}$ are optimal solutions to both the MiniMax and MaxiMin patrolling games we say that the order of play is not important and the game can be played simultaneously. This is what defines the optimal value, if it exists, of a pure patrolling game as in in [16]. The optimal value of the pure (simultaneous) patrolling game $G(Q,T,m)$ is

$$O(Q,T,m) = \min_{a \in \mathcal{A}} \max_{W \in \mathcal{W}} P_p(W,(j,\tau)) = \max_{W \in \mathcal{W}} \min_{a \in \mathcal{A}} P_p(W,(j,\tau)).$$

Optimal solutions to the pure patrolling game are $W^* \in \mathcal{W}$ and $a^* \in \mathcal{A}$ such that $P_p(W^*, a^*) = O(Q,T,m)$. However, unlike the MiniMax and MaxiMin patrolling games in which optimal solutions (and an optimal value) always exist, optimal solutions to the pure patrolling do not always exist (and hence there may be no optimal value).

The optimal value of the pure MiniMax and MaxiMin patrolling games are easy to calculate by considering how the follower can respond. In the MiniMax game $O^{\triangle}(Q,T,m) = 1$ as the patroller can choose $W \in \mathcal{W}$ such that $W(\tau) = j$ against any attacker strategy $(j,\tau) \in \mathcal{A}$. In the MaxiMin game $O^{\triangledown}(Q,T,m) = 1$ if and only if $\exists W \in \mathcal{W}$ such that $\forall a \in \mathcal{A}$ the patroller's payoff $P_p(W,a) = 1$ and $O^{\triangledown}(Q,T,m) = 0$ otherwise, as the attacker can follow the patroller's strategy by choosing $a \in \mathcal{A}$ against $W \in \mathcal{W}$ such that $P_p(W,a) = 0$. Therefore $O(Q,T,m)$ only exists if $\exists W \in \mathcal{W}$ such that $P_p(W,a) = 1 \ \forall a \in \mathcal{A}$, that is there is some walk W such that it visits every node at least every m time units. In other words, the pure patrolling game has an optimal value if and only if there is a walk that *guarantees* to catch all possible attacker strategies.

Figure 2.2.1 shows the pure payoff matrix for the patrolling game $G(K_3, 3, 2)$. Notice that we need not enumerate all walks due the isomorphism which occurs upon relabelling nodes. In the Minimax patrolling game an optimal solution is the attack strategy $a = (1,1)$ and the walk $W = (1,1,1)$ giving an optimal value of 1. In the MaxiMin patrolling game an optimal solution is the walk $W = (1,2,3)$ and the attacker strategy $a = (1,1)$ giving an optimal value of 0. It is clear that the order in which the players go matters so there is no optimal value for the patrolling game. This is equivalent to saying that there is no walk which guarantees catching all attacker strategies, which would correspond to a row of 1s in the payoff matrix.

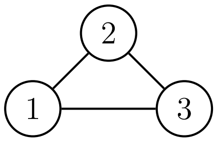
In an optimal strategy combination for the patrolling game $G(Q,T,m)$ no player would change their strategy knowing how the opponent chose, this means that a optimal strategy is equivalent to a Nash equilibrium ([105]) for the game.

Definition 2.2.1 (Nash equilibrium)**.** A pure patroller, attacker strategy combination, $(W^*, (j^*, \tau^*))$ is a *pure Nash equilibrium* of the patrolling game $G(Q,T,m)$ if and only if

$$P_p(W^*, (j^*, \tau^*)) \leq P_p(W^*, (j,\tau)) \quad \forall j \in N, \tau \in \mathcal{T},$$

and

$$P_p(W^*, (j^*, \tau^*)) \geq P_p(W, (j^*, \tau^*)) \quad \forall W \in \mathcal{W}.$$



$$
\begin{array}{cc}
 & \begin{array}{cccccc} & & (j,\tau) & & & \\ (1,0) & (1,1) & (2,0) & (2,1) & (3,0) & (3,1) \end{array} \\
W \; \begin{array}{c} (1,1,1) \\ (1,2,1) \\ (1,2,2) \\ (1,2,3) \\ \vdots \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}
\end{array}
$$

Figure 2.2.1: Complete graph K_3 and the payoff matrix $\mathcal{P}$ for the game $G(K_3, 3, 2)$, with associated walks written aside their row and associated attacks written atop their columns.

Essentially a Nash equilibrium is a strategy combination such that no player can do better by changing there current strategy knowing their opponents strategy. As Nash equilibria are equivalent to optimal strategy combinations for the patrolling game, we know they only exist if and only if there is walk which guarantees catching all attacker strategies. Therefore with pure strategies the patrolling game $G(Q, T, m)$ is solved, either having no optimal value or an optimal value of 1. To make the problem more interesting, we can now consider allowing players to randomize among their available pure strategies to form a *mixed* strategy. Mixed strategies choose a distribution over all pure strategies which are played with the distribution's appropriate probability. Allowing for mixed strategies means it is always possible to get the optimal value and optimal strategy combinations/Nash equilibria if $T \neq \infty$ as the game is zero-sum and finite ([105]). For this purpose, we will now assume that $T \neq \infty$ and note that pure strategies can still be played in the mixed game, so when pure Nash equilibria exist we can find them in the mixed game.

2.2.2 Mixed patrolling game

In this section we explain the strategies available to the patroller and attacker in the mixed patrolling game $G(Q, T, m)$. We will see that Nash equilibria/optimal strategy combinations are guaranteed and that this means the optimal value of the mixed patrolling game always exists. We also include some new notation which will make the proof of our contributions in chapter 3, section 3.2, easier to follow.

The mixed patrolling game $G(Q, T, m)$ uses the same three parameters as the pure patrolling game $G(Q, T, m)$, however mixed strategies are now probability distributions on the set of pure strategies. The patroller chooses some probability distribution $\boldsymbol{\pi} = (\pi_1, ..., \pi_{|\mathcal{W}|})$ such that π_i is the probability of playing the i^{th} ordered walk $W_{(i)} \in \mathcal{W}$. Similarly, the attacker chooses some probability distribution $\boldsymbol{\phi} = (\phi_1, ..., \phi_{|\mathcal{A}|})$ such that ϕ_i is the probability of playing the i^{th} ordered attack $a_{(i)} \in \mathcal{A}$. As $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ are probability distributions

$$\boldsymbol{\pi} \in \Pi(Q,T,m) = \left\{ \boldsymbol{x} \in [0,1]^{|\mathcal{W}(Q,T,m)|} \;\middle|\; \sum_{i=1}^{|\mathcal{W}(Q,T,m)|} x_i = 1 \right\},$$

$$\boldsymbol{\phi} \in \Phi(Q,T,m) = \left\{ \boldsymbol{x} \in [0,1]^{|\mathcal{A}(Q,T,m)|} \;\middle|\; \sum_{i=1}^{|\mathcal{A}(Q,T,m)|} x_i = 1 \right\},$$

with $\Pi(Q,T,m)$ being the set of mixed patroller strategies and $\Phi(Q,T,m)$ being the set of mixed attacker strategies for the game $G(Q,T,m)$. As with pure strategy sets when it is clear for the game $G(Q,T,m)$ we state these sets by $\Pi = \Pi(Q,T,m)$ and $\Phi = \Phi(Q,T,m)$ respectively. For a particular distribution $\boldsymbol{\pi} \in \Pi$ we say that a pure patroller strategy $W \in \mathcal{W}$ is a *potential walk* if $\pi_{\beta_1(W)} > 0$. Similarly for a particular distribution $\phi \in \Phi$ we say that the pure attacker strategy $a \in \mathcal{A}$ is a *potential attack* if $\phi_{\beta_2(a)} > 0$.

We introduce an equivalent notation for the attacker strategy $\boldsymbol{\phi} \in \Phi$ in the form of a matrix

$$\boldsymbol{\varphi} = (\varphi_{j,\tau})_{j\in N,\tau\in\mathcal{T}} = \begin{pmatrix} \varphi_{1,0} & \varphi_{1,1} & \cdots & \varphi_{1,T-m} \\ \varphi_{2,0} & \varphi_{2,1} & \cdots & \varphi_{2,T-m} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{n,0} & \varphi_{n,1} & \cdots & \varphi_{n,T-m} \end{pmatrix}, \tag{2.3}$$

such that $\varphi_{j,\tau} = \phi_{\beta_2((j,\tau))}$ for all $j \in N$ and $\tau \in \mathcal{T}$ is the probability of the attacker playing the pure attacker strategy (j,τ). The matrix notation, as in equation (2.3), allows us to more clearly see the probability distribution for an attacker strategy as a distribution for nodes and commencement times. Further to the matrix representation of an attacker distribution we can define the probability an attacker is at node $j \in N$ at time $t \in \mathcal{J}$ for a given distribution matrix $\boldsymbol{\varphi}$ (or $\boldsymbol{\phi}$) as

$$S_{j,t} = \sum_{k=t-m+1}^{t} \varphi_{j,k}, \tag{2.4}$$

and form the *space-time probability matrix* $\boldsymbol{S} = (S_{j,t})_{j\in N,t\in\mathcal{J}}$ for the attacker distribution matrix $\boldsymbol{\varphi}$. We note that the time-space probability matrix S is able to recover the matrix attacker distribution $\boldsymbol{\varphi}$ as $\varphi_{j,t} = S_{j,t} - S_{j,t-1} + \varphi_{j,t-m}$ and $\varphi_{j,t} = 0$ for $t \leq -1$ and hence is equivalent for the attacker to choose in the game. Figure 2.2.2 provides the visual representation of a space-time probability matrix, in which, for visualization purposes, space-time points are filled grey to denote potential attacks, i.e. (j,t) such that $\varphi_{j,t} > 0$ are filled grey. If the attacker strategy distribution is rational $\boldsymbol{\phi} \in \mathbb{Q}^{|\mathcal{A}|}$, then we can instead think of the distribution as a set potential agents, where each agent chooses some $(j,\tau) \in \mathcal{A}$ and the attacker strategy chooses between all agents with equal probability. To do so we define *agent* equivalents to $\boldsymbol{\phi}$, such that the *agent attacker strategy* is given by $\boldsymbol{\phi}^{\mathrm{A}} = \boldsymbol{\phi} \times \operatorname{lcd}(\boldsymbol{\phi})$ with $\boldsymbol{\varphi}^{\mathrm{A}} = \boldsymbol{\varphi} \times \operatorname{lcd}(\boldsymbol{\phi})$ and the *space-time agent matrix* $\boldsymbol{S}^{\mathrm{A}} = \boldsymbol{S} \times \operatorname{lcd}(\boldsymbol{\phi})$, where $\operatorname{lcd}(\boldsymbol{\phi})$ is the lowest common multiple of all denominators of the rational ϕ_i for $i = 1, ..., |\mathcal{A}|$. The agent equivalent of figure 2.2.2 is shown in figure 2.2.3. While equivalent, using agents allows for a more instinctual way to adapt attacker strategies by adding, removing and/or moving agents in space-time.

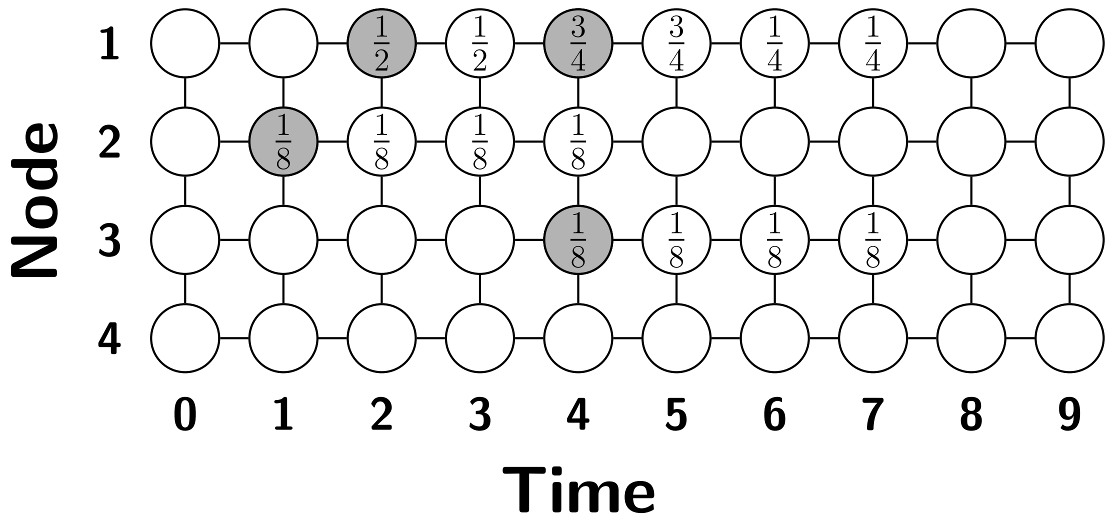


Figure 2.2.2: Space-time probability matrix $\boldsymbol{S}$ of $\boldsymbol{\phi}$ where $\varphi_{1,2} = \frac{1}{2}$, $\varphi_{1,4} = \frac{1}{4}$, $\varphi_{2,1} = \frac{1}{8}$ and $\varphi_{3,4} = \frac{1}{8}$ (with all other $\varphi_{j,t} = 0$).

A pure walk W can be chosen via a mixed strategy, by choosing $\boldsymbol{\pi}$ such that $\pi_i = 1$ for $i = \beta_1(W)$ and $\pi_i = 0$ otherwise. Similarly, a pure attack strategy a can be chosen by choosing $\boldsymbol{\phi}$ such that $\phi_i = 1$ for $i = \beta_2(a)$ and $\phi_i = 0$ otherwise. Now we have extended the strategies in the patrolling game to include mixed strategies we can redefine the patroller's payoff in the game $G(Q, T, m)$ for choosing $\boldsymbol{\pi} \in \Pi$ against an attacker playing $\boldsymbol{\phi} \in \Phi$ as

$$P_p(\boldsymbol{\pi}, \boldsymbol{\phi}) = \sum_{i=1}^{|W|} \sum_{j=1}^{|\mathcal{A}|} \mathcal{P}_{i,j} \pi_i \phi_j = \boldsymbol{\pi} \mathcal{P} \boldsymbol{\phi}^T, \tag{2.5}$$

and the attacker's payoff is $P_a(\boldsymbol{\pi}, \boldsymbol{\phi}) = 1 - P_p(\boldsymbol{\pi}, \boldsymbol{\phi})$, maintaining the zero-sum nature of the game. The patroller's payoff $P_p(\boldsymbol{\pi}, \boldsymbol{\phi})$ can be interpreted as the probability that the patroller choosing $\boldsymbol{\pi}$ catches the attacker choosing $\boldsymbol{\phi}$. For ease of notation we write $P_p(W, \boldsymbol{\phi}) = P_p(\boldsymbol{\pi}, \boldsymbol{\phi})$ where $\boldsymbol{\pi} \in \Pi$ is such that $\pi_i = 1$ for $i = \beta_1(W)$ and $\pi_i = 0$ otherwise. Similarly, $P_p(\boldsymbol{\pi}, a) = P_p(\boldsymbol{\pi}, \boldsymbol{\phi})$ where $\boldsymbol{\phi} \in \Phi$ is such that $\phi_i = 1$ for $i = \beta_2(a)$ and $\phi_i = 0$ otherwise. As the game is zero-sum we drop the patroller player subscript p from the pure patroller payoff equation (2.1) and the mixed patroller payoff (2.5) writing $P(\boldsymbol{\pi}, \boldsymbol{\phi})$ and $P(W, a)$ respectively.

When players are able to choose mixed strategies the MiniMax patrolling game $G^{\triangle}(Q, T, m)$ (attacker leads, patroller follows) has an optimal value of

$$V^{\triangle}(Q, T, m) = \min_{\boldsymbol{\phi} \in \Phi} \max_{\boldsymbol{\pi} \in \Pi} P(\boldsymbol{\pi}, \boldsymbol{\phi}). \tag{2.6}$$

Optimal solutions to which are $\boldsymbol{\pi}^{\triangle} \in \Pi$ and $\boldsymbol{\phi}^{\triangle} \in \Phi$ such that $P(\boldsymbol{\pi}^{\triangle}, \boldsymbol{\phi}^{\triangle}) = V^{\triangle}(Q, T, m)$. Similarly, the MaxiMin patrolling game $G^{\triangledown}(Q, T, m)$(patroller

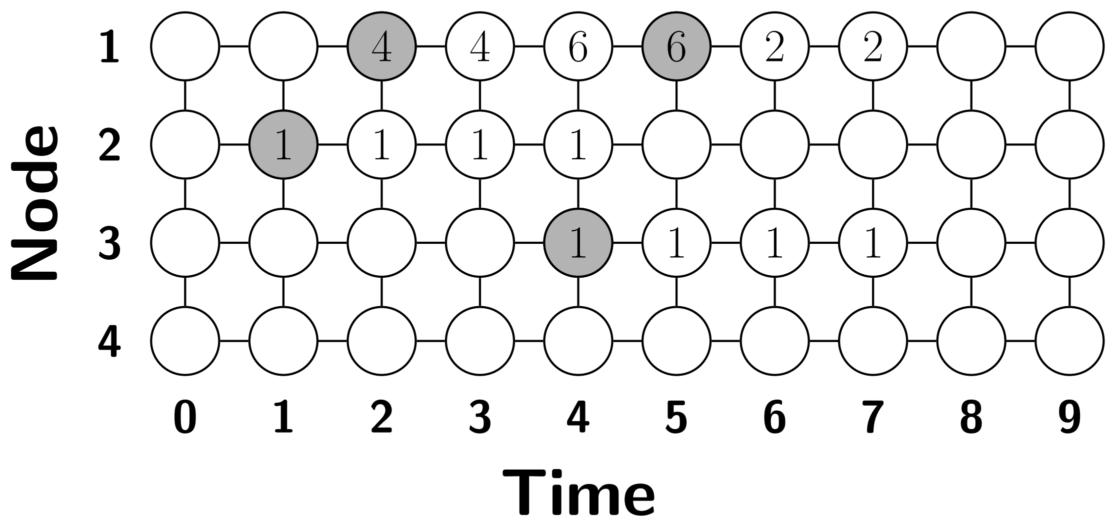


Figure 2.2.3: Space-time agent matrix $\boldsymbol{S}^{\mathrm{A}}$ of $\boldsymbol{\phi}^{\mathrm{A}}$ such that $\varphi^{\mathrm{A}}_{1,2} = 4$, $\varphi^{\mathrm{A}}_{1,4} = 2$, $\varphi^{\mathrm{A}}_{2,1} = 1$, $\varphi^{\mathrm{A}}_{3,4} = 1$ (with all other $\varphi^{\mathrm{A}}_{j,t} = 0$). In which $\boldsymbol{\phi}^{\mathrm{A}}$ is equivalent to $\boldsymbol{\phi}$ such that $\varphi_{1,2} = \frac{1}{2}$, $\varphi_{1,4} = \frac{1}{4}$, $\varphi_{2,1} = \frac{1}{8}$ and $\varphi_{3,4} = \frac{1}{8}$ (with all other $\varphi_{j,t} = 0$), whose space-time probability matrix $\boldsymbol{S}$ is shown in figure 2.2.2.

leads, attacker follows) has an optimal value of

$$V^{\triangledown}(Q,T,m) = \max_{\boldsymbol{\pi}\in\Pi}\min_{\boldsymbol{\phi}\in\Phi} P(\boldsymbol{\pi},\boldsymbol{\phi}). \tag{2.7}$$

Optimal solutions to which are $\boldsymbol{\pi}^{\triangledown} \in \Pi$ and $\boldsymbol{\phi}^{\triangledown} \in \Phi$ such that $P(\boldsymbol{\pi}^{\triangledown}, \boldsymbol{\phi}^{\triangledown}) = V^{\triangledown}(Q,T,m)$. In either game the follower knows the strategy of the leader and so choosing a mixed strategy is not necessary with the follower able to choose a pure strategy that is the *best response* to the leader's mixed strategy. That is we have

$$V^{\vartriangle}(Q,T,m) = \min_{\boldsymbol{\phi}\in\Phi}\max_{W\in\mathcal{W}} P(W,\boldsymbol{\phi}),$$
$$V^{\triangledown}(Q,T,m) = \max_{\boldsymbol{\pi}\in\Pi}\min_{a\in\mathcal{A}} P(\boldsymbol{\pi},a),$$

with the best response to $\boldsymbol{\phi}^{\vartriangle} \in \Phi$ being $W^{\vartriangle} \in \mathcal{W}$ where $P(W^{\vartriangle}, \boldsymbol{\phi}^{\vartriangle}) = V^{\vartriangle}(Q,T,m)$ and the best response to $\boldsymbol{\pi}^{\triangledown} \in \Pi$ being $a^{\triangledown} \in \mathcal{A}$ where $P(\boldsymbol{\pi}^{\triangledown}, a^{\triangledown}) = V^{\triangledown}(Q,T,m)$.

The (optimal) value of the patrolling game $G(Q,T,m)$ is given by

$$V(Q,T,m) = \min_{\boldsymbol{\phi}\in\Phi}\max_{\boldsymbol{\pi}\in\Pi} P(\boldsymbol{\pi},\boldsymbol{\phi}) = \max_{\boldsymbol{\pi}\in\Pi}\min_{\boldsymbol{\phi}\in\Phi} P(\boldsymbol{\pi},\boldsymbol{\phi}), \tag{2.8}$$

in which optimal strategies $\boldsymbol{\pi}^* \in \Pi$ and $\boldsymbol{\phi}^* \in \Phi$ are optimal solutions to the both the MiniMax and MaxiMin patrolling game. While the optimal value to the patrolling game $G(Q,T,m)$ limited to pure strategies does not exist unless there is a walk which is guaranteed to catch all attacker strategies, when mixed strategies are allowed this optimal value always exists. This can be seen by the application of the following theorem from [137].

Theorem 2.2.2. *Let $X \subset \mathbb{R}^p$, $Y \subset \mathbb{R}^q$ be compact convex sets. Then if $f : X \times Y \to \mathbb{R}$ is continuous and convex, concave (i.e convex in X, concave in Y)*

we have

$$\max_{x\in X}\min_{y\in Y} f(x,y) = \min_{y\in Y}\max_{x\in X} f(x,y).$$

By noting that the mixed strategy sets Π and Φ are compact convex sets and that the payoff for any two mixed strategies $P(\boldsymbol{\pi},\boldsymbol{\phi})$ is continuous and convex, concave we know by theorem 2.2.2 that Maximin is equivalent to MiniMax. Therefore, we know the optimal value of the game $G(Q,T,m)$ as in equation (2.8) exists for any graph Q, for any $m \geq 1$ and for any $T \geq m$. We note that we cannot apply theorem 2.2.2 when limited to pure strategies as the payoff function is not continuous.

As the value of the patrolling game $V(Q,T,m)$ always exists, it is always possible to find optimal strategies $\boldsymbol{\pi}^*$ and $\boldsymbol{\phi}^*$. The strategy combination $(\boldsymbol{\pi}^*,\boldsymbol{\phi}^*)$ is equivalent to a mixed Nash equilibrium for the patrolling game $G(Q,T,m)$.

Definition 2.2.3 (Mixed Nash equilibrium)**.** A mixed patroller, attacker strategy pair, $(\boldsymbol{\pi}^*,\boldsymbol{\phi}^*)$ is a *mixed Nash equilibrium* of the mixed patrol game if,

$$P(\boldsymbol{\pi}^*,\boldsymbol{\phi}^*) \leq P(\boldsymbol{\pi}^*,\boldsymbol{\phi}) \quad \forall \phi \in \Phi,$$

and

$$P(\boldsymbol{\pi}^*,\boldsymbol{\phi}^*) \geq P(\boldsymbol{\pi},\boldsymbol{\phi}^*) \quad \forall \pi \in \Pi.$$

As studied in [105], mixed Nash equilibria exist for any two-player finite zero-sum game in which mixed strategies are considered, this is equivalent to applying theorem 2.2.2 to such games. We now look at methods to solve patrolling games $G(Q,T,m)$ by finding the value $V(Q,T,m)$ and optimal strategies $\boldsymbol{\pi}^*$ and $\boldsymbol{\phi}^*$. For the remainder of this thesis we will assume that strategies are mixed strategies in order to ensure that value of the patrolling game exists.

2.2.3 Solving patrolling games

In this section we provide two linear programs which can be used to find the value of the patrolling game $G(Q,T,m)$. The two linear programs follow from the MiniMax patrolling game $G^{\triangle}(Q,T,m)$ and the MaxiMin patrolling game $G^{\triangledown}(Q,T,m)$, which we have seen have equal optimal values. As many methods for solving linear programs are known this approach is common for two player, finite, zero-sum games in which mixed strategies are allowed ([135]). While the use of linear programs is theoretically easy, the computational time required to find the optimal solution grows with the number and nodes and the game length. Therefore the linear programming approach is only practical for patrolling games with a small number of nodes and a small game length and so the more common approach in search theory, of finding equal upper and lower bounds on $V(Q,T,m)$ is used.

The linear program for equation (2.6), the optimal value of the MiniMax patrolling game, is formed by making the optimal max value into a decision variable

$$v = \max_{\boldsymbol{\pi} \in \Pi} P(\boldsymbol{\pi}, \boldsymbol{\phi}) \in \mathbb{R}$$

along with the original decision variable of the attacker strategy $\boldsymbol{\phi} \in \Phi$. In doing so a constraint is enforced on the expected outcome of the attacker strategy for each pure walk and hence the linear program in equation (2.9) is formed.

$$\begin{aligned} &\text{minimize}_{v \in \mathbb{R}, \boldsymbol{\phi} \in \mathbb{R}^{|\mathcal{A}|}} && v \\ &\text{subject to} && v - \sum_{j=1}^{|\mathcal{A}|} \mathcal{P}_{i,j} \phi_j \geq 0, \text{ for all } i \in \{1, ..., |\mathcal{W}|\} \\ & && \boldsymbol{e}^T \boldsymbol{\phi} = 1, \\ & && \boldsymbol{\phi} \geq \mathbf{0}, \end{aligned} \tag{2.9}$$

where $\boldsymbol{e}$ is a row vector full of ones of appropriate size $[1 \times |\mathcal{A}|]$. At an optimal value to the linear program in equation (2.9) $(v^*, \boldsymbol{\phi}^*)$, we have that $v^* = V^{\triangle}(Q, T, m)$ and $\boldsymbol{\phi}^* \in \Phi$ is an optimal solution to the MiniMax patrolling game.

Similarly we can form the linear program for equation (2.7), the optimal value of the MaxiMin patrolling game, by making the optimal value into a decision variable

$$w = \min_{\boldsymbol{\phi} \in \Phi} P(\boldsymbol{\pi}, \boldsymbol{\phi}) \in \mathbb{R}$$

along with the original decision variable of the patroller strategy $\boldsymbol{\pi} \in \Pi$. In doing so a constraint is enforced on the expected outcome of the patroller strategy for each pure attack and hence the linear program in equation (2.10) is formed.

$$\begin{aligned} &\text{minimize}_{w \in \mathbb{R}, \boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{W}|}} && w \\ &\text{subject to} && w - \sum_{i=1}^{|\mathcal{W}|} \mathcal{P}_{i,j} \pi_i \leq 0, \text{ for all } j \in \{1, ..., |\mathcal{A}|\} \\ & && \boldsymbol{e}^T \boldsymbol{\pi} = 1, \\ & && \boldsymbol{\pi} \geq \mathbf{0}, \end{aligned} \tag{2.10}$$

where $\boldsymbol{e}$ is a row vector full of ones of appropriate size $[1 \times |\mathcal{W}|]$. At an optimal value to the linear program in equation (2.10) $(w^*, \boldsymbol{\pi}^*)$, we have that $w^* = V^{\triangledown}(Q, T, m)$ and $\boldsymbol{\pi}^* \in \Pi$ is an optimal solution to the MaxiMin patrolling game.

For any patrolling game $G(Q, T, m)$ to find it's value $V(Q, T, m)$ we need only solve one of the linear programs in equations (2.9) and (2.10), however in order to find an optimal strategy combination $(\boldsymbol{\pi}^*, \boldsymbol{\phi}^*)$ we are required to solve both linear programs. As both linear programs search regions grow according to the number of pure strategies these are computationally infeasible to use for moderately sized patrolling games. An alternative approach is commonly used in which leader strategies for the MiniMax and MaxiMin patrolling games are used to give upper and lower bounds on the value of the game.

For any attacker strategy $\boldsymbol{\phi} \in \Phi$ for the game $G(Q, T, m)$ we know that

$$V(Q, T, m) = V^{\triangle}(Q, T, m) \leq \max_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}) \leq 1, \tag{2.11}$$

meaning that the attacker choosing $\boldsymbol{\phi} \in \Phi$ provides an upper bound on the value of the game $G(Q, T, m)$. Similarly, for any patroller strategy $\boldsymbol{\pi} \in \Pi$ for the game $G(Q, T, m)$ we know that

$$V(Q, T, m) = V^{\triangledown}(Q, T, m) \geq \min_{a \in \mathcal{A}} P(\boldsymbol{\pi}, a) \geq 0, \tag{2.12}$$

meaning that the patroller choosing $\boldsymbol{\pi} \in \Pi$ provides a lower bound on the value of the game $G(Q, T, m)$. We define the *performance* of $\boldsymbol{\phi} \in \Phi$ as

$$V_{\bullet, \boldsymbol{\phi}}(Q, T, m) = \max_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}) \tag{2.13}$$

and the *performance* of $\boldsymbol{\pi} \in \Pi$ as

$$V_{\boldsymbol{\pi}, \bullet}(Q, T, m) = \min_{a \in \mathcal{A}} P(\boldsymbol{\pi}, a). \tag{2.14}$$

Evaluating these performances requires knowing the *best pure response* to the given strategy.

For all $\boldsymbol{\pi} \in \Pi$ and for all $\boldsymbol{\phi} \in \Phi$ we have

$$V_{\boldsymbol{\pi}, \bullet}(Q, T, m) \leq V(Q, T, m) \leq V_{\bullet, \boldsymbol{\phi}}(Q, T, m), \tag{2.15}$$

with equality if and only if the strategy combination $(\boldsymbol{\pi}, \boldsymbol{\phi})$ is optimal for the game $G(Q, T, m)$. Finding $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\phi} \in \Phi$ which have equal performances is the most common way to solve patrolling games. In addition, such performances can be calculated with generality allowing for classes of patrolling games to be solved. Before looking at the performances of some stated strategies we first look at some results for the patrolling game.

2.3 Known results

2.3.1 Known properties and techniques

In this section we provide an overview of properties on the value of the game and techniques which can be used to reduce the search space for optimal strategies, as given in [16]. More precisely, these properties consider changes in the three parameters of patrolling games Q, T and m and what effect they have on the value of the patrolling game. In addition, the reduction of the pure patroller strategy set $\mathcal{W}$ and mixed strategy sets Π and Φ are done by considering dominating strategies and the symmetry of the graph.

For the patrolling game $G(Q, T, m)$ we can consider an increase in the attack length from m to $m + 1$ forming the patrolling game $G(Q, T, m + 1)$. As the set of pure attacker strategies for the game $G(Q,T,m+1)$ is $\mathcal{A}(Q,T,m+1) = N \times \{0,...,T-m-2\}$ and the set of pure attacker strategies for the game $G(Q,T,m)$ is $\mathcal{A}(Q,T,m) = N \times \{0,...,T-m-1\}$ it is clear that $\mathcal{A}(Q,T,m+1) \subset \mathcal{A}(Q,T,m)$, meaning the attacker has less strategies when the attack length is increased. Hence for all simple graphs Q, for all $m \geq 1$ and for all $T \geq m$ we have

$$V(Q,T,m+1) \geq V(Q,T,m). \tag{2.16}$$

That is an upper bound is achieved for $V(Q,T,m)$, by the performance of an attacker strategy which is optimal for the game $G(Q,T,m+1)$. Likewise, we can compare $G(Q,T,m)$ and $G(Q,T+1,m)$ and see that $\mathcal{A}(Q,T,m) \subset \mathcal{A}(Q,T+1,m)$ and so increasing the game length increases the number of pure attacker strategies. Hence for all simple graphs Q, for all $m \geq 1$ and for all $T \geq m$ we have

$$V(Q,T+1,m) \leq V(Q,T,m). \tag{2.17}$$

That is a lower bound is achieved for $V(Q,T,m)$ by the performance of a patroller strategy which is optimal for the game $G(Q,T+1,m)$. We note that in the case of increasing game length changes the set of pure walks for the patroller as they must now go up to time T rather than time $T-1$. We can form $\mathcal{W}(Q,T+1,m)$ by taking each walk W in $\mathcal{W}(Q,T,m)$ and concatenating it with each node $j \in N$ such that $j = W(T-1)$ or $(W(T-1),j) \in E$ to form $W' = (W(0),...,W(T-1),j)$. However for any walk $W' \in \mathcal{W}(Q,T+1,m)$ against any attacker strategy $a \in \mathcal{A}(Q,T,m)$ we have

$$P(W',a)(Q,T+1,m) = P(W,a)(Q,T,m),$$

where $W = (W'(0),....,W'(T-1))$ and hence the bound.

When considering an alteration to the graph $Q = (N,E)$ we must consider how the graph is altered, has there been a change in the set of nodes N or the set of edges E. First consider introducing a new edge e to the current set of edges E to form a new set of edges E' and the new graph $Q' = (N,E')$. Then comparing the patrolling game $G(Q,T,m)$ to $G(Q',T,m)$ it is clear that $\mathcal{W}(Q,T,m) \subset \mathcal{W}(Q',T,m)$, as all walks in $G(Q',T,m)$ can use all previous edges in E as well as the additional edge e. Therefore, the addition of an edge increases the number of pure walks for the patroller. Hence for all simple graphs $Q = (N,E)$, for all $i,j \in N$, for all $m \geq 1$ and for all $T \geq m$ we have

$$V(Q',T,m) \geq V(Q,T,m), \tag{2.18}$$

where $Q' = ((N, E \cup \{(i,j)\}), T, m)$. That is a upper bound is achieved for $V(Q,T,m)$ by the performance of an attacker strategy which is optimal for the game $G(Q',T,m)$. It is possible to get a similar result for the introduction of additional nodes into a graph Q (see node-splitting in section 3.3.2), however in [16] they consider merging two nodes via *node-identification* (see [47]).

Definition 2.3.1. The graphical operator of *node-identification* maps a (simple undirected) graph $Q = (N,E)$ onto Q^- by identifying two nodes $u,v \in N$ written as $\mathcal{Q}^-(Q,u,v) = Q^-$. The resultant graph $Q^- = (N^-,E^-)$ is such that $N^- = (N \cup \{w\} \setminus \{u,v\})$ and $E^- = (E \setminus E') \cup \{(w,x) | (u,x) \in E \text{ or } (v,x) \in E\}$, where $E' = \{(u, v)\}$ if $(u, v) \in E$ and $E' = \emptyset$ if $(u, v) \notin E$. Further, the node-identification node and edge maps are given by $\mathcal{N}^-(Q, u, v, j)$ and $\mathcal{E}^-(Q, u, v, e)$ respectively, which map nodes $j \in N$ and edge $e \in E$ of Q to nodes and edges of Q^-.

The process of node-identification merges two nodes u and v known as the *parent* nodes into a *child* node w such that w is adjacent to all nodes that at least one of it's parents, u and v, were adjacent to. We can then consider the patrolling games $G(Q, T, m)$ and $G(Q^-, T, m)$ where $Q^- = \mathcal{Q}^-(Q, u, v)$ for some $u, v \in N$. Every walk $W \in \mathcal{W}(Q, T, m)$ can be mapped to a walk $W' = \mathcal{N}^-(Q, u, v)(W) \in \mathcal{W}(Q^-, T, m)$ such that $P(W', a)(Q^-, T, m) = P(W, a)(Q, T, m)$ for any pure attacker strategy $a \in \mathcal{A}(Q, T, m)$. However $\mathcal{A}(Q^-, T, m) \subset \mathcal{A}(Q, T, m)$, under an isomorphic relabelling of the nodes, and therefore we get that merging two nodes results in less pure attacker strategies and hence the following lemma.

Lemma 2.3.2 (Lemma 1, part 4, from [16])**.** *For any game $G(Q^-, T, m)$ where Q^- is node-identified from the graph Q, for any $m \geq 1$ and for for any $T \geq m$ we have*

$$V(Q^-, T, m) \geq V(Q, T, m),$$

with the upper bound on $V(Q, T, m)$ achieved by an attacker strategy which is optimal in the game $G(Q^-, T, m)$.

In addition to results on the properties of patrolling games, in [16] they show that some pure strategies can be removed from $\mathcal{W}$ and $\mathcal{A}$ when searching for an optimal solution. Namely, any walk W can be removed from $\mathcal{W}$ if $W(t) = W(t + 1) = W(t + 2)$ for some $t \in \{0, ..., T - 3\}$ and any pure attacker strategy (j, τ) can be removed from $\mathcal{A}$ if $m \geq 3$ and j is a penultimate node (that is it is adjacent to a leaf node). While these can be removed from the pure strategy sets, this is equivalent to setting strategy distributions to play such walks and pure attacks with zero probability. Further reductions to the strategy sets Π and Φ can be made by considering symmetric nodes. Consider two nodes $i, j \in N$ which are symmetric (that is there is an automorphism which swaps the symmetric nodes), then an optimal attacker strategy $\boldsymbol{\phi}$ exists such that $\varphi_{i,\tau} = \varphi_{j,\tau}$ for all $\tau \in \mathcal{T}$. Symmetric strategies in time can be considered by using $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ such that $\varphi_{1,j,\tau} = \varphi_{2,j,T+1-\tau-m}$, where $\varphi_{1,j,\tau}$ is the probability of (j, τ) under the distribution $\boldsymbol{\phi}_1$ and $\varphi_{2,j,\tau}$ is the probability of (j, τ) under the distribution $\boldsymbol{\phi}_2$. However the performance of such strategies are equal as given

$$W_1 = \underset{W \in \mathcal{W}}{\arg\max}\, P(W, \boldsymbol{\phi}_1)$$

then for $W_2(t) = W_1(T - 1 - t)$ for all $t \in \mathcal{J}$ we have

$$W_2 = \underset{W \in \mathcal{W}}{\arg\max}\, P(W, \boldsymbol{\phi}_2),$$

.

Having seen general properties of the value of the game, along with the reduction of strategy sets, we now consider known patroller strategies along with their respective performances, stating the lower bounds they provide on the value of the game.

2.3.2 Known patroller strategies

We now survey known patroller strategies $\boldsymbol{\pi} \in \Pi$ provided in [16] and their respective performances $V_{\boldsymbol{\pi},\bullet}(Q,T,m)$ giving lower bounds on the value $V(Q,T,m)$. We start with a patrolling strategy which chooses a node $j \in N$ with equal probability and waits at node j for the entire time-horizon.

Definition 2.3.3. For the game $G(Q,T,m)$ ($Q = (N,E)$) the *choose and wait* patroller strategy $\boldsymbol{\pi}_{\text{cw}} \in \Pi$ is such that $\pi_{\beta_1(W)} = \frac{1}{|N|}$ if W is such that $W(t) = j$ for all $t \in \mathcal{J}$ for some $j \in N$ and $\pi_{\beta_1(W)} = 0$ otherwise.

Lemma 2.3.4 (Lemma 2 from [16])**.** *For the game $G(Q,T,m)$ for any graph $Q = (N,E)$, for all $m \geq 1$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q,T,m) \geq V_{\boldsymbol{\pi}_{cw},\bullet}(Q,T,m) = \frac{1}{|N|},$$

where the lower bound on $V(Q,T,m)$ is achieved by the patroller choosing the choose and wait patroller strategy $\boldsymbol{\pi}_{cw}$.

While the choose and wait patroller strategy $\boldsymbol{\pi}_{\text{cw}}$ provides a lower bound on $V(Q,T,m)$ for any game $G(Q,T,m)$, it is clearly ineffective for a large class of patrolling games as we know for $T \geq 3$ and $m \geq 2$ that waiting for more than 2 time units at any node is dominated. Therefore, $\boldsymbol{\pi}_{\text{cw}}$ is really only useful for patrolling games in which $m = 1$.

We next consider a patroller strategy which can only be chosen in the class of patrolling games $G(Q,T,m)$ in which the graph Q is Hamiltonian.

Definition 2.3.5. A *Hamiltonian cycle* for the graph $Q = (N,E)$ is a walk H of length $|N|$, such that $H(0) = H(|N|)$ and there exists some $t \in \{0,1,...,|N|-1\}$ with $H(t) = i$ for all $i \in N$. If the graph Q contains a Hamiltonian cycle it is called a *Hamiltonian graph.* The set of all Hamiltonian graphs is denoted by $\mathcal{H}$.

A Hamiltonian graph $Q = (N,E) \in \mathcal{H}$ has a Hamiltonian cycle H which can be considered for use in creating pure patroller strategies/walks of length T. Let $W_i(t) = H(t+i \mod |N|)$ for all $t \in \mathcal{J}$ for $i = 0,...,|N|-1$, then W_i is the pure patroller strategy that strategy that starts at the i^{th} node along the Hamiltonian cycle H and then follows it, repeating H as required for the time-horizon. Then if $m \geq |N|$ we know that $W_i(\{\tau,...,\tau+m-1\}) = N$ and hence for any $j \in N, \tau \in \mathcal{T}$ we know $j \in W_i(\{\tau,...,\tau+m-1\})$ for any $i = 0,...,|N|-1$. Hence, $W_i \in \mathcal{W}$ is guaranteed to catch any pure attacker strategy for any $i = 0,...,|N|-1$ and so $V(Q,T,m) = 1$. Similarly, we can define a patroller strategy which plays the set of walks $\{W_0,...,W_{|N|-1}\}$ with equal probability.

Definition 2.3.6. For the game $G(Q,T,m)$ where $Q = (N,E) \in \mathcal{H}$ with a Hamiltonian cycle H, the *Hamiltonian* pure patroller strategy (using H) is $W \in \mathcal{W}$ such that $W(t) = H(t+i \mod |N|)$ for all $t \in \mathcal{J}$. The *random Hamiltonian* patroller strategy (using H) is $\boldsymbol{\pi}_{\text{rH}}$ such that $\pi_{\beta_1(W)} = \frac{1}{|N|}$ if $W \in \{W_0,...,W_{|N|-1}\}$, where $W_i(t) = H(t+i \mod |N|)$ for all $t \in \mathcal{J}$ for $i = 0,...,|N|-1$, and $\pi_{\beta_1(W)} = 0$ otherwise.

Using a random Hamiltonian patroller strategy $\boldsymbol{\pi}_{rH}$ the probability of a patroller being at any node $j \in N$ at any time $t \in \mathcal{J}$ is $\frac{1}{|N|}$, hence any attacker strategy $(j, \tau) \in \mathcal{A}$ has a probability of catching the attacker of $\frac{m}{|N|}$.

Lemma 2.3.7 (Theorem 13 from [16])**.** *For the game $G(Q, T, m)$ for any Hamiltonian graph $Q = (N, E) \in \mathcal{H}$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q, T, m) \geq V_{\boldsymbol{\pi}_{rH}, \bullet}(Q, T, m) = \frac{m}{|N|},$$

where the lower bound on $V(Q, T, m)$ is achieved by the patroller choosing a random Hamiltonian patroller strategy $\boldsymbol{\pi}_{rH}$ (for any Hamiltonian cycle of Q).

As a patroller choosing a random Hamiltonian strategy has a constant probability of $\frac{1}{|N|}$ for being at each node for each time, it can be thought to provide uniformity across both space and time for the patroller. In chapter 3, section 3.3.4, we build on this work, to allow us to use the same idea in non-Hamiltonian graphs. While doing so no longer provides uniformity across nodes, we see that it is still possible to obtain patroller strategies which retain their uniformity in time.

For the game $G(Q, T, m)$ there may not be a pure patroller strategy which guarantees catching all pure attacks at each node within the graph Q. However we can consider a subgraph of Q, Q_1, in which there is a pure patroller strategy which can catches all pure attacks meaning $V(Q_1, T, m) = 1$. This concept of having a pure patroller strategy which is guaranteed to catch all pure attacks at all nodes within its walk is called *intercepting*.

Definition 2.3.8. For the game $G(Q, T, m)$ a walk $W \in \mathcal{W}$ is called *intercepting*, if

$$P(W, (j, \tau)) = 1,$$

for all $(j, \tau) \in W(\mathcal{J}) \times \mathcal{J}$.

In the case that no intercepting walk exists for the whole graph Q, we can consider multiple intercepting walks to ensure that at each node in the graph Q is in at least one of the intercepting walks. That is we consider R intercepting walks each distinct on subgraphs $Q_1, ..., Q_R$, such that

$$Q = \bigcup_{r=1}^{R} Q_r.$$

This idea of having a collection of intercepting walks, on subgraphs which collectively form the graph is called a *covering set*.

Definition 2.3.9. For the game $G(Q, T, m)$ a set of intercepting walks $C = \{W_1, ..., W_R\}$ is called a *covering set*, if for each node $j \in N$ there exists $r \in \{1, ..., R\}$, such that $j \in W_r(\mathcal{J})$.

For the game $G(Q, T, m)$ with a covering set $C = \{W_1, ..., W_R\}$ the patroller can consider playing each intercepting walk W_i for $i \in \{1, ..., R\}$ with equal probability.

Definition 2.3.10. For the game $G(Q, T, m)$ with a covering set C the covering patroller strategy using C is $\boldsymbol{\pi}_{\text{Cov}}$ such that $\pi_{\beta_1(W)} = \frac{1}{|C|}$ for $W \in C$ and $\pi_{\beta_1(W)} = 0$ otherwise.

Using a covering patroller strategy for some covering set C for the game $G(Q, T, m)$ gives a lower bound on the value of the game for all graphs Q, for all $m \geq 1$ and for all $T \geq m$ of

$$V(Q, T, m) \geq V_{\boldsymbol{\pi}_{\text{Cov}}, \bullet}(Q, T, m) = \frac{1}{|C|}. \tag{2.19}$$

This can be seen as any pure attack $(j, \tau) \in \mathcal{A}$ there exists some $i \in \{1, ..., R\}$ such that $P(W_i, (j, \tau)) = 1$ which is played with probability $\frac{1}{|C|}$. From equation (2.19) it is clear that forming a covering strategy using a covering set with minimal cardinality gives the best lower bound amongst all choices for covering strategies.

Definition 2.3.11. For a game $G(Q, T, m)$ a covering set C is called a *minimal covering set* if $|C| = \min_{C' \in S} |C'|$, where S is the set of all covering sets for $G(Q, T, m)$. Moreover, the cardinality of such a set is called the *covering number* for the game $G(Q, T, m)$, denoted as $\mathcal{C}_{Q,T,m}$.

We note that finding the covering number $\mathcal{C}_{Q,T,m}$ for a graph is related to the minimal edge covering for a graph. In graph theory an *edge covering* is a set of edges such that every node is incident (at one end) of an edge in the set. For $m = 2$ intercepting pure patroller strategies are equivalent to edges, as the patroller must alternate between two nodes. Therefore when $m = 2$ we may utilise known results in order to find the covering number and minimal covering set. However the problem of finding a minimal edge covering is widely known to be NP-Hard ([48]). While exact algorithms exist, their computation time is such that only small graphs, or graphs with structures which reduce the complexity of the problem, can be solved within a reasonable time.

Lemma 2.3.12 (Lemma 12 from [16])**.** *For the game $G(Q, T, m)$ for any graph $Q = (N, E)$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q, T, m) \geq \frac{1}{\mathcal{C}_{Q,T,m}},$$

where the lower bound on $V(Q, T, m)$ is achieved by the patroller choosing a covering patroller strategy $\boldsymbol{\pi}_{Cov}$, using a minimal covering set C.

The idea of using multiple intercepting walks in a covering set, divides the graph into subgraphs in which the game played upon them is a guaranteed win for the patroller (using the intercepting walks). However it is possible to use this idea of decomposition, without requiring that each subgraph games $G(Q_r, T, m)$ has a value $V(Q_r, T, m) = 1$. For a decomposition of the graph Q into $Q_1, ..., Q_R$ such that

$$Q = \bigcup_{i=1}^{R} Q_i,$$

the patroller can play the optimal strategy for each subgraph game with an appropriate weighting.

Definition 2.3.13. For the game $G(Q, T, m)$ with a decomposition of Q into $Q_1, ..., Q_R$, we form the subgraph games $G(Q_1, T, m), ..., G(Q_R, T, m)$ with optimal patroller strategies, $\boldsymbol{\pi}_1^*, ..., \boldsymbol{\pi}_R^*$. A *decomposition* patroller strategy using the decomposition above $\boldsymbol{\pi}_{\text{Dec}}$ is such that $\pi_{\beta_1(W)} = \sum_{i=1}^{R} p_i \pi^*_{i,\beta_1(W)}$, where

$$p_i = \frac{1}{V(Q_i, T, m) \sum_{r=1}^{R} \frac{1}{V(Q_r,T,m)}},$$

for $i \in \{1, .., R\}$.

That is a decomposition patroller strategy $\boldsymbol{\pi}_{\text{Dec}}$ using a decomposition of Q into $Q_1, ..., Q_R$ plays an optimal strategy for the subgraph $V(Q_i, T, m)$ with probability p_i.

Lemma 2.3.14 (Lemma 6 from [16])**.** *For the game $G(Q, T, m)$ for any graph Q with any decomposition Q_i for $i = 1, .., R$ ($Q = \bigcup_{i=1}^{R} Q_i$), for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q, T, m) \geq V_{\boldsymbol{\pi}_{Dec}}(Q, T, m) \geq \frac{1}{\sum_{i=1}^{R} \frac{1}{V(Q_i,T,m)}},$$

where the lower bound on $V(Q, T, m)$ is achieved by the patroller choosing a decomposition patroller strategy $\boldsymbol{\pi}_{Dec}$ using the decomposition Q_i for $i = 1, .., R$. Moreover, if the subgraphs Q_i for $i = 1, .., R$ are disjoint and disconnected we have

$$V(Q, T, m) = \frac{1}{\sum_{i=1}^{R} \frac{1}{V(Q_i,T,m)}}.$$

If Q is a disconnected graph, lemma 2.3.14 shows we can just treat each connected subgraph of Q separately. Henceforth we can assume that the graph Q is a connected simple graph. In chapter 3, section 3.3.1, we extend lemma 2.3.14 to cover the situation when the optimal strategies, $\boldsymbol{\pi}_i^*$ for some $i = 1, ..., R$, are not known. Having now seen known patroller strategies, which can be chosen by the patroller to give a lower bound on the value of the game, we now look at known attacker strategies.

2.3.3 Known attacker strategies

We now survey known attacker strategies $\boldsymbol{\phi} \in \Phi$ provided in [16] and their respective performances $V_{\bullet,\boldsymbol{\phi}}(Q, T, m)$, giving upper bounds on the value $V(Q, T, m)$. We start with an attacker strategy which for a fixed constant commencement time chooses amongst nodes uniformly.

Definition 2.3.15. For the game $G(Q,T,m)$ $(Q=(N,E))$ a *position-uniform* attacker strategy $\boldsymbol{\phi}_{\text{pu}}$, using a fixed commencement time $t \in \mathcal{T}$, is such that the probability of choosing node (j,τ) is $\varphi_{j,\tau} = \frac{1}{|N|}$ if $\tau = t$ for all $j \in N$ and $\varphi_{j,\tau} = 0$ otherwise.

As the position-uniform attacker strategy randomizes only over nodes and has a fixed commencement time t, it is clear that the best pure patroller is one that visits the most nodes during the time interval $I = \{t, ..., t+m-1\}$. So the performance $V_{\bullet,\boldsymbol{\phi}_{\text{pu}}}(Q,T,m) = P(W, \boldsymbol{\phi}_{\text{pu}})$ where W is such that $W(I)$ has the maximal number of distinct nodes $\omega^* \leq m$. Note that ω^* does not depend on the interval and hence does not depend on the fixed commencement time t.

Lemma 2.3.16 (Lemma 2 from [16]). *For the game $G(Q,T,m)$ for any graph $Q=(N,E)$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q,T,m) \leq V_{\bullet,\boldsymbol{\phi}_{pu}}(Q,T,m) = \frac{\omega^*}{|N|} \leq \frac{m}{|N|},$$

where ω^* is the maximum number of distinct nodes a pure patroller can visit in a walk of length m. The upper bound on $V(Q,T,m)$ is achieved by the attacker choosing a position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$ for any fixed commencement time $t \in \mathcal{T}$.

It is possible to randomize the commencement time uniformly among all $\tau \in \mathcal{T}$ to achieve the same result as lemma 2.3.16. This *uniform* attacker strategy $\boldsymbol{\phi}_{\text{U}}$ is such that the probability of choosing the pure attacker strategy (j,τ) is $\varphi_{j,\tau} = \frac{1}{|N|(T-m)}$ for all $j \in N, \tau \in \mathcal{T}$. While this result is not stated in [16] it is easy to see that $\boldsymbol{\phi}_{\text{U}}$ has an equal performance to $\boldsymbol{\phi}_{\text{pu}}$, as in lemma 2.3.16. This follows as $\boldsymbol{\phi}_{\text{U}}$ equally randomizations among a set of equally performing strategies $\{\boldsymbol{\phi}_{0,pu}, ..., \boldsymbol{\phi}_{T-m,pu}\}$, where $\boldsymbol{\phi}_{t,pu}$ is the position-uniform attacker strategy using the fixed commencement time t.

In a similar idea to covering sets for the patroller which contains intercepting walks which are guaranteed to catch any pure attacker at a node they visit, an independence set for the attacker can be formed in which no two pure attacks can be caught by a single pure patroller strategy. In order to define this strategy we must first define the distance between two nodes on a graph.

Definition 2.3.17. For a graph $Q=(N,E)$ the *distance* between nodes $i,j \in N$, denote $d(i,j)$, is the minimal length of a walk from i to j.

Thus if $d(j,j') \geq m$ then no pure patroller can catch both $(j,t) \in \mathcal{A}$ and $(j',t) \in \mathcal{A}$ for any $t \in \mathcal{T}$. In this case we call nodes j and j' *independent*. We define independent sets, which have each node independent of every other node.

Definition 2.3.18. For the game $G(Q,T,m)$ a set of pure attacks L is called an *independent* set, if $\tau = \tau'$ and $d(j,j') \geq m$ for all $(j,\tau),(j',\tau') \in L$.

Definition 2.3.19. For the game $G(Q,T,m)$ with an independent set L, the independent attacker strategy using L is $\boldsymbol{\phi}_{\text{Ind}}$ such that the probability of choosing (j,τ) is $\varphi_{j,\tau} = \frac{1}{|L|}$ for $(j,\tau) \in L$ and $\varphi_{j,\tau} = 0$ otherwise.

Using an independent attacker strategy for some independent set L for the game $G(Q,T,m)$ gives an upper bound on the value of the game for all graphs Q, for all $m \geq 1$ and for all $T \geq m$ of

$$V(Q,T,m) \leq V_{\bullet,\phi_{\mathrm{Ind}}}(Q,T,m) = \frac{1}{|L|}. \tag{2.20}$$

This can be seen as for any $W \in \mathcal{W}$ the m nodes visited $W(I)$ during the attack interval I can only contain one node in L. From equation (2.20) it is clear that forming an independent attacker strategy using an independent set with a maximal cardinality gives the best upper bound amongst all choices for independent strategies.

Definition 2.3.20. For the game $G(Q,T,m)$ an independent set L is called a *maximal independent set* if $|L| = \max\limits_{L' \in S} |L'|$ where S is the set of all independent sets for $G(Q,T,m)$. Moreover, the cardinality of such a set is called the *independence number* for the game $G(Q,T,m)$, denoted $\mathcal{L}_{Q,T,m}$.

As with the covering number, we note that independence number $\mathcal{L}_{Q,T,m}$ is related to the maximal *vertex independence* number for a graph. In graph theory, a vertex independent set is a set of nodes such that no two are adjacent. We refer to it as a vertex independent set to distinguish between our definition and that of some graph theory literature. While the definitions for use in the patrolling game do not exactly match these graph theory definitions unless $m = 2$, the core idea of having spatially separated nodes remains the same. Determining the maximal vertex independent set, like the minimal covering set, is NP-Hard ([39]). We consider vertex independence in chapter 5.

Lemma 2.3.21 (Lemma 12 from [16])**.** *For the game $G(Q,T,m)$ for any graph Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q,T,m) \leq \frac{1}{\mathcal{L}_{Q,T,m}},$$

where the upper bound on $V(Q,T,m)$ is achieved by the attacker choosing an independent attacker strategy ϕ_{Ind}, using a maximal independent set L.

Another way to use nodes which are a distance apart is to use two such they are the furthest distance apart.

Definition 2.3.22. For a graph $Q = (N,E)$ the *diameter* is

$$\bar{d} = \max_{(j,j') \in N \times N} d(j,j'),$$

where a pair of nodes (j,j') such that $d(j,j') = \bar{d}$ is called a *diametric pair.*

In [16] a diametric pair is considered to form the *diametric attacker strategy* such that ‘the Attacker’s diametrical strategy is to attack these nodes [the diametric nodes] equiprobably during a random time interval I’ and the performance of these diametric attacker strategies give an upper bound. However, the arbitrary statement about randomness in the attack interval I causes issues with the proposed upper bound in [16]. This issue of randomness amongst all possible commencement times (equivalent to attack intervals) is discused in section 3.3.3. For now we state the diametric attacker strategy and the upper bound proposed in [16] and provide a counter-example to show that this upper bound does not always hold.

Definition 2.3.23 (Diameteric attack as in lemma 9 from [16])**.** For the game $G(Q, T, m)$ a *diametric attacker strategy* $\boldsymbol{\phi}_{\text{di}}$, using the diametric pair (j, j'), is such that the probability of choosing the pure strategy $(i, \tau) \in \mathcal{A}$ is

$$\varphi_{i,\tau} = \begin{cases} \frac{1}{2(T-m+1)} & \text{if } i \in \{j, j'\} \text{ and } \tau \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.21}$$

Lemma 2.3.24 (Diametric attacker bound as in lemma 9 from [16])**.** *For the game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q, T, m) \leq \max\left(\frac{1}{2}, \frac{m}{2\bar{d}}\right),$$

where $\bar{d}$ is the diameter of the graph Q. The upper bound on $V(Q, T, m)$ is achieved by the attacker choosing a diametric attacker strategy $\boldsymbol{\phi}_{\text{di}}$, for any diametric pair (j, j').

Notice that the upper bound in lemma 2.3.24 is the same as

$$V(Q, T, m) \leq \begin{cases} \frac{1}{2} & \text{for } m \leq \bar{d}, \\ \frac{m}{2\bar{d}} & \text{for } \bar{d} < m \leq 2\bar{d}, \end{cases}$$

with the case of $m \leq \bar{d}$ being equivalent to the application of equation (2.20) with the independent set $L = \{(j, 0), (j', 0)\}$, where (j, j') is the diametric pair. Unfortunately, the upper bound provided by the attacker choosing a diametric attacker strategy in lemma 2.3.24 is incorrect. We provide counterexample 2.3.25 to demonstrate the issue with lemma 2.3.24. The issue we can see from using an equally chosen commencement time is that a pure patroller can catch more potential attacks. This is because of the distribution in commencement time and how many potential attacks a pure patroller who moves between the pair of diametric nodes (with some waiting) can capture.

Counterexample 2.3.25. Consider the game $G(L_4, 7, 4)$. That is the game on the line graph consisting of 4 nodes with a game length of 7 and an attack length of 4. A pair of diametric nodes for the graph L_4 is $(1, 4)$ and hence $\boldsymbol{\phi}_{\text{di}}$ can be formed and $\bar{d} = 3$. Figure 2.3.1 shows the space-time probability matrix $\boldsymbol{S}_{\text{di}}$ which is equivalent to $\boldsymbol{\phi}_{\text{di}}$ along with the pure patroller walk $W = (1, 2, 3, 4, 3, 2, 1)$ in red. From the figure it it is clear that $P(W, \boldsymbol{\phi}_{\text{di}}) = \frac{1}{8} + \frac{4}{8} + \frac{1}{8} = \frac{6}{8}$ and so have a performance of

$$V_{\bullet, \boldsymbol{\phi}_{\text{di}}}(L_4, 7, 4) \geq \frac{6}{8} > \frac{4}{6}.$$

However this contradicts the upper bound given in lemma 2.3.24 and therefore lemma 2.3.24 does not hold.

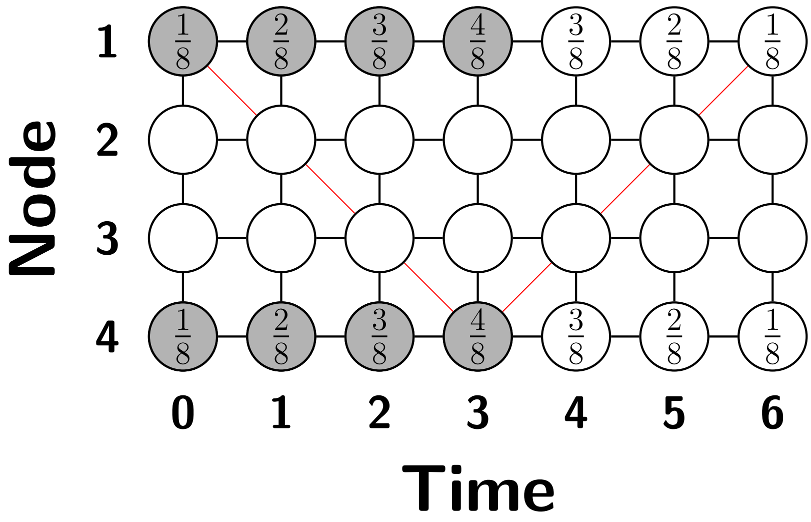


Figure 2.3.1: The space-time probability matrix $\boldsymbol{S}_{\mathrm{di}}$ for a diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{di}}$ used in the game $G(L_4, 7, 4)$ with a pure patroller strategy $W = (1, 2, 3, 4, 3, 2, 1)$ shown in red.

We note that in counterexample 2.3.25 limiting the distribution of the commencement times to choose equally from only $\{0, 1, 2\}$, rather than $\{0, 1, 2, 3\}$ would remedy the issue, achieving the upper bound as proposed in lemma 2.3.24.

In chapter 3, section 3.3.3, we find the corrected performance of the diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{di}}$, noticing a dependence on the game length T. Moreover, we will see that the upper bound in lemma 2.3.24 holds if $T = m - 1 + (k+1)\bar{d}$ for some $k \in \mathbb{N}_0$. We will also show that limiting the distribution for an attacker strategy in commencement time can achieve the upper bound equivalent to that in lemma 2.3.24 but requires a game length $T \geq m + \bar{d} - 1$. As lemma 2.3.24 is incorrect, it's subsequent use in the patrolling game on the line graph L_n, leads to an incorrect statement about the optimality of the game. In particular how using $\boldsymbol{\phi}_{\mathrm{di}}$ is only optimal when $T = m - 1 + (k+1)\bar{d}$ for some $k \in \mathbb{N}_0$. However, with our work in section 3.3.3 we are able to get the same optimal value as stated in [107] when $T \geq m + n - 2$ by using our developed attacker strategy, alleviating the aforementioned issue.

2.3.4 Solved patrolling games

In this section we will survey classes of patrolling games which have previously been solved. We showcase the value of the game as well as the known optimal strategies which were used to give tight bounds. We cover games in which $m = 1$, followed by games on Hamiltonian, complete bipartite, and line graphs.

Any patrolling game with $m = 1$ have tight/equal lower and upper bounds which are given by lemmas 2.3.4 and 2.3.16.

Lemma 2.3.26 (Lemma 2 from [16]). *For the game $G(Q, T, 1)$ for any graph*

$Q = (N, E)$ and for all $T \geq 1$ we have

$$V(Q, T, 1) = \frac{1}{|N|},$$

achieved by the choose and wait patroller strategy $\boldsymbol{\pi}_{cw}$ and any position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$ (using any fixed commencement time t).

For $G(Q, T, m)$ where $Q \in \mathcal{H}$ is Hamiltonian, the game is solved for any game length and attack length as we get equal lower and upper bounds given by lemmas 2.3.7 and 2.3.16.

Lemma 2.3.27 (Hamiltonian lemma)**.** *For the game $G(Q, T, m)$ for any graph $Q = (N, E) \in \mathcal{H}$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q, T, m) = \frac{m}{|N|},$$

achieved by a random Hamiltonian patroller strategy $\boldsymbol{\pi}_{rH}$ (using any Hamiltonian cycle H) and a position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$ (using any fixed commencement time t).

A Hamiltonian graph with the least edges, containing n nodes, is the *cyclic graph* of n nodes, C_n. Having a circular graphic structure, C_n forms the 'backbone' of any Hamiltonian graph and additional edges do not improve the value of the game (later such edges are called *superfluous*). The Hamiltonian graph with the most edges is the *complete graph* of n nodes K_n, in which each node is adjacent to every other node. We highlight that fact that

$$V(C_n, T, m) = V(K_n, T, m) \quad \forall n \in \mathbb{N}, \forall m \geq 1, \forall T \geq m,$$

as in chapter 3, section 3.3.4, we use a Hamiltonian graph to develop patroller strategies for non-Hamiltonian graphs, and so we use the cyclic graph for simplicity.

Next solved is the game $G(Q, T, m)$ where Q is a complete bipartite graph. A *bipartite graph* is graph such that there exists some non-empty node sets A and B such that $N = A \cup B$, $A \cap B = \emptyset$, $(i, j) \notin E$ for all $i, j \in A$ and $(i, j) \notin E$ for all $i, j \in B$. A *complete bipartite graph* is a bipartite graph with the additional condition on the sets A and B, that $(i, j) \in E$ for all $i \in A, j \in B$. A complete bipartite graphs Q may be written as $Q = K_{a,b}$ where $|A| = a$ and $|B| = b$ where without loss of generality $1 \leq a \leq b$. We denote the set of all biparite graphs by $\mathcal{P}_2$ and the set of all complete bipartite graphs by $\mathcal{KP}_2$. For $Q \in \mathcal{P}_2$, the use of repeated node-identification along with lemma 2.3.27 was used to develop a lower bound. An equal upper bound was found when $K_{a,b} \in \mathcal{KP}_2$ by having the attacker choose pure attack (j, τ) with probability

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{b} & \text{if } j \in B, \tau = t, m \text{ is even}, \\ \frac{1}{2b} & \text{if } j \in B, \tau = t, t+1, m \text{ is odd}, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 2.3.28 (Theorem 15 from [16])**.** *For the game $G(Q, T, m)$ for any graph $Q \in \mathcal{P}_2$ (where $1 \leq |A| = a \leq |B| = b$), for all $m \geq 1$ and for all $T \geq m + 1$ we have*

$$V(Q, T, m) \leq \frac{m}{2b}.$$

Further if the graph is complete, i.e $Q = K_{a,b} \in \mathcal{KP}_2$, then

$$V(K_{a,b}, T, m) = \frac{m}{2b}.$$

In chapter 3, section 3.5, we extend lemma 2.3.28 by discussing what is concretely needed to obtain this bipartite solution when $Q \in \mathcal{P}_2 \setminus \mathcal{KP}_2$. Furthermore we use our new bounds, developed in section 3.3, to solve all complete k-partite graphs for $k \geq 3$.

A special complete bipartite graphs is the star graph defined by $S_n \equiv K_{1,n}$ for $n \in \mathbb{N}$. By lemma 2.3.28 we know the value for all games on star graphs, $V(S_n, T, m) = \frac{m}{2n}$. While a special subset of complete bipartite graphs, we highlight the star graph as in chapter 4 we will elongate it by mixing its structure with that of a line graph.

Another special biparite graphs is the line graphs $L_n = (N, E)$ such that $N = \{1, 2, ..., n\}$ and $E = \{(i, i + 1) \mid i = 1, ..., n - 1\}$ for $n \in \mathbb{N}$. Any line graph L_n is bipartite but not complete bipartite (unless $n = 2, 3$) and so lemma 2.3.28 does not give the value. The value of the patrolling game $G(L_n, T, m)$ was found for all $n \geq 1$, for all $m \geq 1$ and for all $T \geq 2m$ in [107]. The optimal strategies which provided equal upper and lower bounds depend on the attack length m. The decomposition of the set of attack lengths $\mathbb{N}$ allowed for regions of attack lengths to be solved. In one set of attack lengths bespoke attacker strategies are needed which are only feasible strategies if $T \geq 2m$, thus requiring this condition. We will see that unlike previously solved graphs, increasing the attack length increases the value in a non-linear fashion. We provide the decomposition of the attack length set dependent on n. This is followed by the value of the game and optimal strategies in the game.

- $M_0^{L_n} = \{m : m = 1\}$,
- $M_1^{L_n} = \{m : m > 2(n - 1)\}$,
- $M_2^{L_n} = \{m : n - 1 < m \leq 2(n - 1)\}$,
- $M_3^{L_n} = \{m : m = 2, n \geq 3\}$,
- $M_4^{L_n} = \{m : m = n - 1,\ \text{or}\ m = n - 2 \text{ and } m \text{ even}\}$,
- $M_5^{L_n} = \{m : 3 \leq m \leq n - 3,\ \text{or}\ m = n - 2 \text{ and } m \text{ odd}\}$.

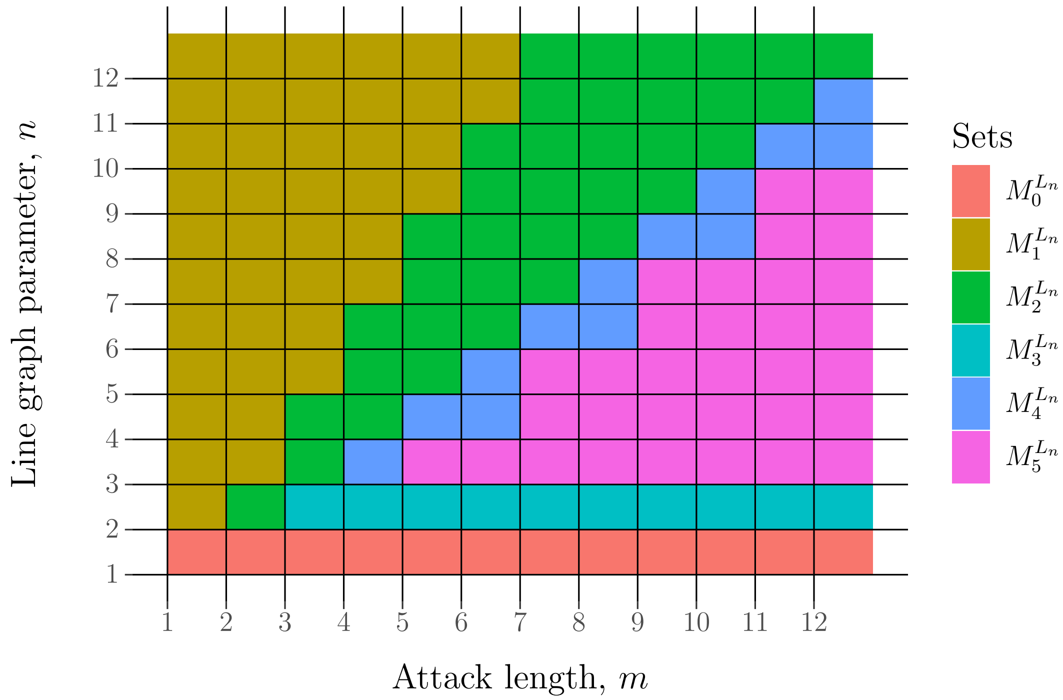


Figure 2.3.2: The decomposition of the attack length sets $\mathbb{N}$ into attack length regions $M_i^{L_n}$ for $i = 0, ..., 5$ dependent on n. Shown for $n, m = 1, ..., 12$.

We note that the order of these regions are arbitrary and chosen to match those used in [107]. Figure 2.3.2 shows the decomposition of $\mathbb{N} = \bigcup_{i=0}^{5} M_i^{L_n}$ for various $n \in \mathbb{N}$ in $(n, m) \in \mathbb{N}^2$ space.

Lemma 2.3.29. *For the game $G(L_n, T, m)$ for all $n \in \mathbb{N}$,*

- *for all $T \geq 1$ and $m = 1$ ($m \in M_0^{L_n}$)*

$$V(L_n, T, 1) = \frac{1}{n},$$

  achieved by the choose and wait patroller strategy $\boldsymbol{\pi}_{cw}$ and any position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$ (using any fixed commencement time t).

- *for all $T \geq m$ and $m \in M_1^{L_n}$,*

$$V(L_n, T, m) = 1,$$

  achieved by oscillating (embedded Hamiltonian) patroller and any attacker strategy.

- *for $T \geq m$ and $m \in M_2^{L_n}$,*

$$V(L_n, T, m) = \frac{m}{2(n-1)},$$

  achieved by the patroller choosing a random oscillation patroller strategy (for details see [107]) and a diametric attacker strategy.

- *for $T \geq 2$ and $m = 2$ ($m \in M_3^{L_n}$),*

$$V(L_n, T, 2) = \frac{1}{\lceil \frac{n}{2} \rceil},$$

achieved by a covering patroller strategy using a minimal covering set C and a independent attacker strategy using a maximal independent set L.

- *for $T \geq m$ and $m \in M_4^{L_n}$,*

$$V(L_n, T, m) = \frac{1}{2},$$

achieved by a covering patroller strategy using a minimal covering set C and a independent attacker strategy using a maximal independent set L.

- *for $T \geq 2m$ and $m \in M_5^{L_n}$,*

$$V(L_n, T, m) = \frac{m}{n - 1 + m},$$

achieved by an end-augmented Hamiltonian patroller strategy and a bespoke attacker strategies dependent on $n - 1 \mod m$ (for details see [107]).

Lemma 2.3.29 has the value $V(L_n, T, m) = \frac{m}{2(n-1)}$, for $m \in M_2^{L_n}$, achieved by the attacker choosing a diametric attacker strategy ϕ_{di} using the diametric pair $(1, n)$ (or equivalently $(n, 1)$). However, we saw in section 2.3.3 that performance of this strategy does not always equal the value of $\frac{m}{2(n-1)}$ for the line graph. In chapter 3, section 3.3.3, we provide an attacker strategy which does have an equal performance and therefore restate this part of lemma 2.3.29 with a correct optimal strategy. Note that this requires a restriction of the game lengths to $T \geq m + n - 2$.

The decomposition of the set of all considered attack lengths $\mathbb{N}$ into regions in which different optimal strategies and different values are found is needed in chapter 4, where we mix the graphical structure of the star, S_{n-1}, and line, L_{k+2}, to form the *elongated star* graph, S_n^k. Furthermore, while the decomposition of $\mathbb{N}$ into $M_i^{L_n}$ for $i \in \{0, 1, 2, 3, 4, 5\}$ depends only on one graph parameter n for L_n, our decomposition in chapter 4 depends on two graph parameters n and k.

To conclude our survey of known work we compare the values of two games $G(L_n, T, m)$ and $G(C_n, T, m)$ for various attack lengths m, assuming $T \geq 2m$, to highlight the drastic difference, in value, the single edge $(1, n)$ can make. Figure 2.3.3 shows the values of the two games for $n = 10$. The inclusion of the single edge $(1, n)$ causes drastic changes in optimal strategies as the patroller must repeat nodes $n - 1, n - 2, ..., 2$ to get back to node 1 in L_n compared to C_n. In non-Hamiltonian graphs the inclusion of additional edges can greatly affect the value of the game if they lower the number of nodes which need to be repeated to return to prior nodes. Another way to see this is look at L_n as $C_{2(n-1)}$ (by node splitting), which has almost double the amount of nodes as C_n. This idea of using a comparison to a cycle is extremely useful, as it allow us to get optimal bounds on graphs which are non-Hamiltonian. In chapter 3, section 3.3.4, we see that it is possible to get a 'Hamiltonian bound' for non-Hamiltonian graphs. We see in sections 3.5, 4.2.3 and 4.3.2 that this approach leads to optimal strategies for a variety of patrolling games on different graphs. Furthermore, in chapter 5 we conjecture that under some conditions that this strategy is optimal for all patrolling games on trees $Q = (N, E)$ in which $m \geq |N| - 1$.

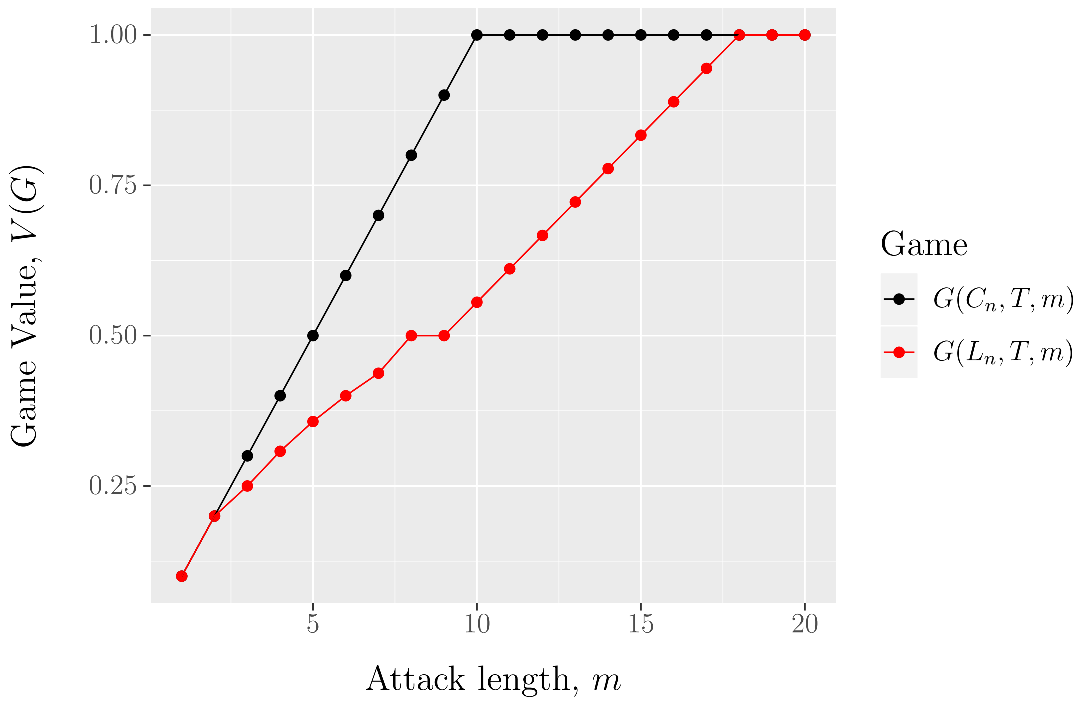


Figure 2.3.3: A comparison of the value of the game $G(Q, T, m)$ for the cyclic graph $Q = C_n$, shown in black, and the line graph $Q = L_n$, shown in red for $n = 10$, when $m = 1, ..., 2n + 2$ for any $T \geq 2m$.

2.4 Concluding comments

In this chapter we have defined the patrolling game $G(Q, T, m)$ and seen that the common way to find its value $V(Q, T, m)$ (and optimal strategies $\boldsymbol{\pi}^* \in \Pi$ and $\boldsymbol{\phi}^* \in \Phi$) is to find equal lower and upper bounds by looking at the performance of strategies as in equation (2.15). We saw some known strategies for the patroller and their performances, which provide a lower bound on the value and similarly known strategies for the attacker and their performances, which provide an upper bound on the value. We have seen the value of classes of patrolling games which have been solved, these include patrolling games with: an attack length $m = 1$, Hamiltonain graphs $Q \in \mathcal{H}$, complete bipartite graphs $Q \in \mathcal{KP}_2$, and line graphs $Q = L_n$ for some $n \in \mathbb{N}$. In particular we saw that solution to the patrolling game on the line graph required decomposition and in some regions bespoke attacker strategies (see [107] for details), leading in a complex solution for the class of patrolling games on line graphs.

This survey of the previous literature found an error with the performance of the diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{di}}$ provided in [16]. In chapter 3, section 3.3.3, we provide the performance of $\boldsymbol{\phi}_{\mathrm{di}}$ and improve the strategy to provide a better upper bound, with a lower performance matching that suggested in lemma 2.3.24.

In the following chapter we look at new techniques to reduce the strategy spaces for games and produce more general strategies along with their performances and bounds they provide on the value. These are applied to solve classes of patrolling games when $Q \in \mathcal{P}_2 \setminus \mathcal{KP}_2$ and when $Q \in \mathcal{KP}_k$ for $k \geq 3$.

Chapter 3

New techniques and strategies

3.1 Introduction

In this chapter we provide new techniques for strategy space reductions for general patrolling games $G(Q, T, m)$ and in particular provide large reductions in the search for the optimal walk $W \in \mathcal{W}$ in the evaluation of the performance of an attacker strategy strategy $\boldsymbol{\phi} \in \Phi$ in $V_{\bullet,\boldsymbol{\phi}}(Q, T, m)$ (as in equation (2.13)). We then provide general patroller and attacker strategies, evaluating their performances and therefore giving bounds on the value $V(Q, T, m)$. In particular, we find the performance of the diametric attacker strategy $\boldsymbol{\phi}_{\text{di}}$ and correct the result given in [16]. We adapt the random Hamiltonian strategy $\boldsymbol{\pi}_{rH}$, which is only feasible for Hamiltonian graphs, to non-Hamiltonian graphs and thus use the idea to find a lower bound on the value for any graph. Furthermore, we look at the Patroller Improvement Program (PIP) which can be solved in order to find an improved patroller strategy with a greater performance.

This chapter is structured as follows, we begin in section 3.2.1 by considering how shifting strategies in time can reduce the number of strategies we need to consider. In section 3.2.2 we provide some reductions in the pure patroller space when calculating the performance of an arbitrary attacker strategy. This is followed by section 3.2.3 which provides further reductions for an attacker strategy with certain properties. After considering some strategy set reductions we present some general strategies for both the patroller and attacker. In section 3.3.1 we consider decomposition into subgraph games in which the optimal strategies are not necessarily known. In section 3.3.2 we consider how node-identification can be repeated/reversed and how embedded strategies can be generated. In section 3.3.3 we find the correct performance of the diametric attacker strategy $\boldsymbol{\phi}_{\text{di}}$ and present an alternative strategy which performs better. In section 3.3.4 we find 'Hamiltonian like' lower bounds for non-Hamiltonian graphs via embedding, creating a strategy called the random minimal full-node cycle strategy $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$. This is followed by section 4.2.5, which identifies nodes for which $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ performs weakly at. Then in section 3.4 we present the Patrol Improvement Program (PIP) as a way to improve patroller strategies given a finite group of strategies. Finally, in section 3.5, we provide solutions to k-partite graphs by using techniques and strategies seen throughout this chapter.

3.2 Strategy reduction techniques

3.2.1 Time-shifting

In this section, we reduce the set of mixed attacker strategies Φ from which the attacker can choose, by showing that for any $\boldsymbol{\phi} \in \Phi$ that any s-time-shifted strategy $\boldsymbol{\phi}^s$ has an equal performance. An s-time-shift performed on player strategies moves each potentially played pure strategy s forward in time (or equivalently moves the time-horizon s backwards).

Definition 3.2.1. A pure patroller strategy $W \in \mathcal{W}$ can be can be *s-time-shifted*, for some $s \in \mathbb{Z}$ to form $W^s \in \mathcal{W}$, where

$$W^s(t) = \begin{cases} W(t-s) & \text{if } s \leq t \leq T+s-1, \\ W'(t) & \text{otherwise,} \end{cases}$$

where $W'(t)$ is an arbitrary path such that:

- $W'(s) = W(0)$ if $s \geq 0$.
- $W'(T+s-1) = W(T+s-1)$ if $s < 0$.

Let $\xi(W, s, W') : \mathcal{W} \to \mathcal{W}$ be the time-shifting function mapping W to W^s for the path choice W'. A patroller strategy $\boldsymbol{\pi} \in \Pi$ can be *s-time-shifted*, for some $s \in \mathbb{Z}$, to form a strategy $\boldsymbol{\pi}^s \in \Pi$ such that

$$\pi^s_{\beta_1(W)} = \sum_{U \in \mathcal{W}} \sum_{i=1}^{|\mathcal{W}|} \pi_{\beta_1(\xi(U,s,W'_i))},$$

where W'_i are the path choices for $i \in \{1, ..., |\mathcal{W}|\}$.

Notice that for any patroller strategy $\boldsymbol{\pi} \in \Pi$ a large set of s-time-shifted exist depending on the selection of the arbitrary path $W'(t)$ chosen for each $W \in \mathcal{W}$. Therefore, the performance of any particular s-time-shifted patroller strategy $\boldsymbol{\pi}^s$ compared to $\boldsymbol{\pi}$ depends greatly on the choice of W' for each $W \in \mathcal{W}$. This is true for any $s \in \mathbb{Z}$ so time-shifting patroller strategies is not useful for finding lower bounds. However, when time-shifting attacker strategies we limit the s such that their is no arbitrary pure attacker strategies which need to be chosen. That is, we limit the time-shifting such that all pure attack intervals for an attacker strategy remain within the time-horizon.

Definition 3.2.2. A pure attacker strategy $a = (j, \tau) \in \mathcal{A}$ can be s-time-shifted for some $s \in \mathbb{Z}$ such that $\tau + s \geq 0$ and $\tau + s + m - 1 \leq T - 1$ to form a^s, where $a^s = (j, \tau + s)$. An attacker strategy $\boldsymbol{\phi} \in \Phi$ (which plays a pure attack (j, τ) with probability $\varphi_{j,\tau}$) can be *s-time shifted* for some $s \in \mathbb{Z}$ such that

$$-\min \mathcal{T}_Y \leq s \leq T - m - \max \mathcal{T}_Y$$

where,

$$\mathcal{T}_Y = \{\tau \in \mathcal{T} \mid \exists j \in N \text{ s.t. } \varphi_{j,\tau} \neq 0\}$$

to form an attacker strategy $\boldsymbol{\phi}^s$, where $\boldsymbol{\phi}^s$ plays a pure attack $(j, \tau) \in \mathcal{A}$ with probability $\varphi^s_{j,\tau} = \varphi_{j,\tau-s}$.

Lemma 3.2.3. *For any $\boldsymbol{\phi} \in \Phi$ for any $s \in \mathbb{Z}$ such that*

$$-\min \mathcal{T}_Y \leq s \leq T - m - \max \mathcal{T}_Y$$

where,

$$\mathcal{T}_Y = \{\tau \in \mathcal{T} \mid \exists j \in N \textit{ s.t. } \varphi_{j,\tau} \neq 0\}$$

for any graph Q, for all $m \geq 1$ and for all $T \geq m$ we have

$$V_{\bullet,\boldsymbol{\phi}}(Q, T, m) = V_{\bullet,\boldsymbol{\phi}^s}(Q, T, m).$$

Moreover, if W is an optimal response to $\boldsymbol{\phi}$, then the s-time-shifted walk W^s is an optimal response to the s-time-shifted attacker strategy $\boldsymbol{\phi}^s$.

The proof of lemma 3.2.3 follows as the responding pure patroller can time-shift their walk by s, thus it is possible to get the same performance. Moreover, the pure patroller cannot do better against $\boldsymbol{\phi}^s$ than they could against $\boldsymbol{\phi}$.

Proof. We first notice that for any walk $W \in \mathcal{W}$, any attacker strategy $a \in \mathcal{A}$ and for any s, as given in the lemma, we have that $P_p(W, a) = P_p(W^s, a^s)$, where W^s and a^s are s-time-shifted versions of W and a respectively. So for any $W \in \mathcal{W}$, for any $\boldsymbol{\phi} \in \Phi$ and for any $s \in \mathbb{Z}$ we have that $P(W, \boldsymbol{\phi}) = P(W^s, \boldsymbol{\phi}^s)$, where $\boldsymbol{\phi}^s$ is the s-time-shifted version of $\boldsymbol{\phi}$.

Thus

$$\begin{aligned} V_{\bullet,\boldsymbol{\phi}}(Q, T, m) &= \min_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}) = P(W^*, \boldsymbol{\phi}) \\ &= P(W^{s,*}, \boldsymbol{\phi}^s) \geq \min_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}^s) = V_{\bullet,\boldsymbol{\phi}^s}(Q, T, m), \end{aligned} \tag{3.1}$$

where W^* is the optimal walk in response to $\boldsymbol{\phi}$ and $W^{s,*}$ is the s-time-shifted version of W^*. Similarly,

$$\begin{aligned} V_{\bullet,\boldsymbol{\phi}^s}(Q, T, m) &= \min_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}) = P(W^*, \boldsymbol{\phi}^s) \\ &= P(W^{-s,*}, \boldsymbol{\phi}) \geq \min_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}) = V_{\bullet,\boldsymbol{\phi}}(Q, T, m), \end{aligned} \tag{3.2}$$

where W^* is the optimal walk in response to $\boldsymbol{\phi}^s$ and $W^{-s,*}$ is the $-s$-time-shifted version of W^*. Hence, by equations (3.1) and (3.2), we have

$$V_{\bullet,\boldsymbol{\phi}}(Q, T, m) = V_{\bullet,\boldsymbol{\phi}^s}(Q, T, m).$$

□

Corollary 3.2.4. *For any game $G(Q,T,m)$ if $(\boldsymbol{\pi},\boldsymbol{\phi}) \in \Pi \times \Phi$ is an optimal strategy combination then $(\boldsymbol{\pi},\boldsymbol{\phi}^s)$ is an optimal strategy combination for any $s \in \mathbb{Z}$ such that*

$$-\min \mathcal{T}_Y \leq s \leq T - m - \max \mathcal{T}_Y$$

where,

$$\mathcal{T}_Y = \{\tau \in \mathcal{T} \mid \exists j \in N \ s.t. \ \varphi_{j,\tau} \neq 0\}.$$

Corollary 3.2.4 follows immediately from lemma 3.2.3 as the upper bound provided by choosing $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^s$ are equal. Furthermore lemma 3.2.3 informs us that we can find an optimal attacker strategy which has $\varphi_{j,0} > 0$ for some $j \in N$. This restriction reduces Φ to $\Phi' = \{\boldsymbol{\phi} \in \Phi \mid \exists j \in N$ such that $\varphi_{j,0} > 0\}$, meaning we will always consider an attacker strategy which has a potential attack commencing at time 0. In the following section we consider how to reduce the space of responses for a pure patroller against a given attacker strategy.

3.2.2 Reduction of patroller response space for arbitrary attacker strategy

In this section we aim to reduce the set of pure patroller responses for an arbitrary attacker strategy, thus making the performance of an arbitrary attacker strategy easier to find. That is for equation (2.13), viz.

$$V_{\bullet,\boldsymbol{\phi}}(Q,T,m) = \max_{W \in \mathcal{W}} P(W,\boldsymbol{\phi}), \tag{3.3}$$

we look at reducing the set of walks $\mathcal{W}$ played in response to any $\boldsymbol{\phi} \in \Phi$, to find the best pure patroller response W^* to $\boldsymbol{\phi}$ such that $V_{\bullet,\boldsymbol{\phi}}(Q,T,m) = P(W^*,\boldsymbol{\phi})$. The space-time probability matrix $\boldsymbol{S} = (S_{j,t})_{j \in N, t \in \mathcal{J}}$, as given in equation (2.4), can be used to find $P(W,\boldsymbol{\phi})$. Example 3.2.5 provides an idea of how $\boldsymbol{S}$ can be used to calculate $P(W,\boldsymbol{\phi})$ for some walk $W \in \mathcal{W}$.

Example 3.2.5. For the patrolling game $G(K_4,10,4)$ to calculate $P(W,\boldsymbol{\phi})$ for

$$W = (1,4,1,2,3,4,3,1,3,3)$$

and $\boldsymbol{\phi}$ such that the probability of choosing (j,τ) is given by

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2} & \text{if } (j,\tau) = (1,2), \\ \frac{1}{4} & \text{if } (j,\tau) = (1,4), \\ \frac{1}{8} & \text{if } (j,\tau) \in \{(2,1),(3,4)\}, \end{cases}$$

we can see the unique probabilities in the space-time probability matrix $\boldsymbol{S}$ of $\boldsymbol{\phi}$ caught by the walk W. Figure 3.2.1 shows $\boldsymbol{S}$ and W, giving that

$$P(W,\boldsymbol{\phi}) = 0 + 0 + \frac{1}{2} + \frac{1}{8} + \frac{1}{8} + 0 + 0 + \frac{1}{4} + 0 + 0 = \frac{7}{8}.$$

We note that the walk W at time 6 is at node 3 and does not gain any probability from this visit to node 3, as it would previous be caught by the prior visit to node 3 which occurred at time 4.

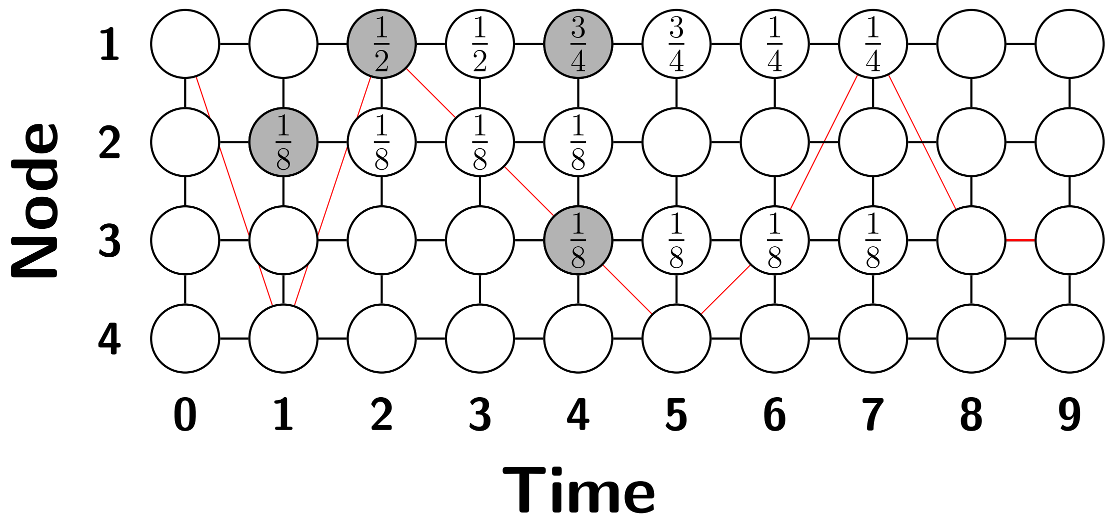


Figure 3.2.1: Space-time probability matrix $\boldsymbol{S}$ for example 3.2.5 with the pure walk $W = (1, 4, 1, 2, 3, 4, 3, 1, 3, 3)$ shown in red.

Example 3.2.5, while showing the usefulness of the grid sum probability matrix representation of a mixed attacker strategy $\boldsymbol{\phi}$, notes an issue with calculating $P(W, \boldsymbol{\phi})$. The issue arises as the walk W may return to a node within m time units of its previous visit, therefore possibly seeing the same pure attack again and not gaining any additional probability. In order to avoid this issue and calculate $P(W, \boldsymbol{\phi})$ we can define the space-time probability *dependent on the time l_j the node j was last visited at*, as

$$S_{j,t}(l_j) = \sum_{k=\max\{t-m+1,\, l_j+1,\, 0\}}^{t} \varphi_{j,k},$$

for all $j \in N$ and for all $t \in \mathcal{J}$. The *lv-space-time probability matrix*, for a given vector $\boldsymbol{l} = (l_1, ..., l_{|N|})$ of *last visit times* (where l_j is the last visit time to node j), is therefore $\boldsymbol{S}(\boldsymbol{l}) = (S_{j,t}(l_j))_{j \in N, t \in \mathcal{J}}$. For a given walk W we can write $\boldsymbol{l}(W, t)$ for the last visit times at time t, given by $l_j(W, 0) = -m$ for $j \neq W(0)$, $l_{W(0)}(W, 0) = 0$ and $l_j(t) = l_j(t-1)$ if $j \neq W(t)$, $l_{W(t)}(W, t) = t$ for $t \geq 1$. Thus we have

$$P(W, \boldsymbol{\phi}) = \sum_{t=0}^{T-1} S_{W(t),t}(\boldsymbol{l}(W, t)). \tag{3.4}$$

While $\boldsymbol{S}(\boldsymbol{l})$ is more complicated to visualize, due to the dependence on the last visit times $\boldsymbol{l}$, the payoff of a pure walk W against a mixed attacker strategy $\boldsymbol{\phi}$ is easier to calculate and will prove extremely useful in evaluating walks against attacker strategies.

In addition to this we introduce some notation given an attacker strategy $\boldsymbol{\phi} \in \Phi$ let

$$N_A = \{j \in N \mid \varphi_{j,t} \neq 0 \text{ for some } t \in \mathcal{T}\}$$

be the set of nodes which are possibly attacked under $\boldsymbol{\phi}$. With this we can restrict the last time nodes were visited along a walk $\boldsymbol{l}(W,t)$ to only contain $l_j(W,t)$ for $j \in N_A$. This restriction is useful for the direct computation of the performance $V_{\bullet,\boldsymbol{\phi}}(Q,T,m)$ by evaluating $P(W,\boldsymbol{\phi})$ for all $W \in \mathcal{W}$, when $|N|$ is large.

To show the complexity of decisions which need to be made to find a best response pure patroller against $\boldsymbol{\phi} \in \Phi$ we present example 3.2.6. Even with the complexity of determining the best response pure patroller for an arbitrary attacker strategy $\boldsymbol{\phi}$, the results in this section provide some extremely useful reductions to the search space $\mathcal{W}$, thus reducing computational time. Moreover, in section 3.2.3 we consider properties of the attacker strategies in order to get further reductions to the already reduced search space.

Example 3.2.6. For the game $G(Q,6,3)$ for which Q is the graph given in figure 3.2.2 consider the attacker strategy $\boldsymbol{\phi}$ whose space-time probability matrix $\boldsymbol{S}$ is given in figure 3.2.3. To find $V_{\bullet,\boldsymbol{\phi}}(Q,6,3)$ we must consider all walks $W \in \mathcal{W}$ and identify the best responses by choosing one that maximizes $P(W,\boldsymbol{\phi})$.

The payoff of the pure patroller strategy $W_1 = (1,2,3,4,5,6)$, seen in red in figure 3.2.3, against $\boldsymbol{\phi}$ can be calculated by using equation (3.4) giving

$$P(W_1,\boldsymbol{\phi}) = 0.08 + 0.12 + 0 + 0 + 0.04 + 0 = 0.24.$$

It is clear that we can adjust W_1, and do better, by simply avoiding moving from node 3 to node 4, as there is no probability of a pure attack at node 4 and the subsequent move to node 5 is available from node 3. This small adjustment from W_1 gives us a new pure patroller strategy W_2, seen in blue in figure 3.2.3, with

$$P(W_2,\boldsymbol{\phi}) = 0.08 + 0.12 + 0 + (0.04 + 0.04) + 0.2 + 0 = 0.48.$$

As $P(W_2,\boldsymbol{\phi}) = 2P(W_1,\boldsymbol{\phi})$ it is clear that W_1 cannot be a best response pure patroller against W_1, however we cannot be sure that W_2 is either. By a brute force calculation of $P(W,\boldsymbol{\phi})$ for all W we find that a best response pure patroller against $\boldsymbol{\phi}$ is $W^* = (5,6,5,6,5,6)$ such that $V_{\bullet,\boldsymbol{\phi}}(Q,6,3) = P(W^*,\boldsymbol{\phi}) = 0.56$.

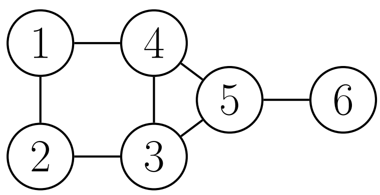


Figure 3.2.2: Graph Q for example 3.2.6

We can first reduce the space $\mathcal{W}$ by considering only pure patroller strategies such that $W(0) \in N_A$.

Lemma 3.2.7. *For the game $G(Q,T,m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for all $\boldsymbol{\phi}$ we have*

$$V_{\bullet,\boldsymbol{\phi}}(Q,T,m) = \max_{W \in \mathcal{W}'} P(W,\boldsymbol{\phi}),$$

where $\mathcal{W}' = \{W \in \mathcal{W} \mid W(0) \in N_A\}$.

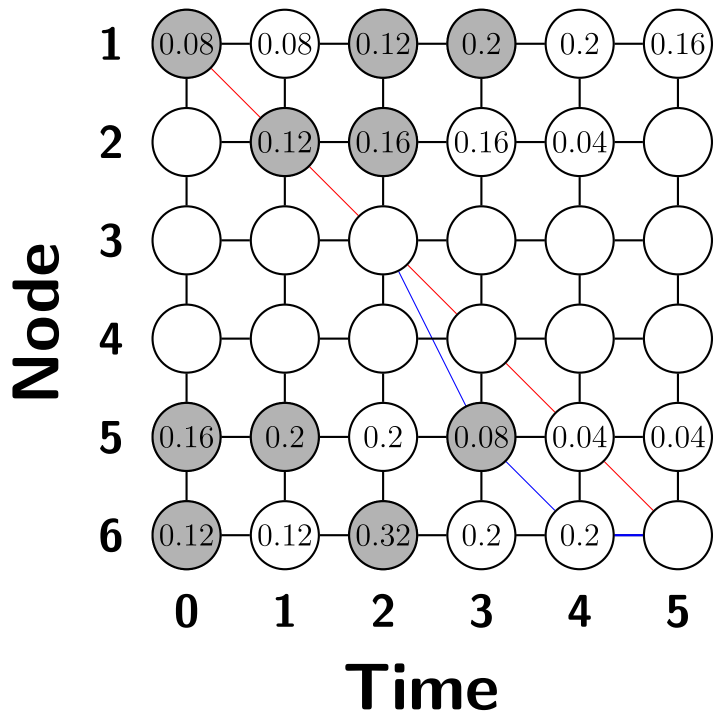


Figure 3.2.3: Space-time probability matrix $\boldsymbol{S}$ for example 3.2.6. The two pure patrollers W_1 and W_2, used in the example, are shown in red and blue respectively.

The proof of lemma 3.2.7 follows by the construction of a pure strategy in $\mathcal{W}'$ by changing the walk to wait at the first node in N_A that it visits. Figure 3.2.4 shows the idea behind this change to a pure walk.

Proof. For all $W \in \mathcal{W}$ we seek to show that there exists a $W' \in \mathcal{W}'$ such that $P(W', \boldsymbol{\phi}) \geq P(W, \boldsymbol{\phi})$.

For $W \in \mathcal{W}$ let $t_{\text{first}} = \min\{t \mid W(t) \in N_A\}$ and construct $W' \in \mathcal{W}'$ such that

$$W'(t) = \begin{cases} W(t_{\text{first}}) & \text{if } t \leq t_{\text{first}}, \\ W(t) & \text{otherwise.} \end{cases}$$

Immediately note that $\boldsymbol{l}(W', t) = \boldsymbol{l}(W, t)$ for all $t \geq t_{\text{first}}$ and $l_{W'(t)}(W', t) = t - 1$ for all $t \leq t_{\text{first}}$ (Note that we restricted the last visit time vector to only have elements $j \in N_A$). So we have that

$$\begin{aligned}
\sum_{t=0}^{t_{\text{first}}} S_{W(t_{\text{first}}),t}(\boldsymbol{l}(W',t)) &= \sum_{t=0}^{t_{\text{first}}} \sum_{k=\max(t-m+1,t,0)}^{t} \varphi_{W(t_{\text{first}}),k} \\
&= \sum_{t=0}^{t_{\text{first}}} \varphi_{W(t_{\text{first}}),t} \\
&\geq \sum_{k=\max(t-m+1,-m+t,0)}^{t} \varphi_{W(t_{\text{first}}),k} \\
&= \sum_{t=0}^{t_{\text{first}}} S_{W(t),t}(\boldsymbol{l}(W,t)).
\end{aligned}$$

Therefore,

$$\begin{aligned}
P(W',\boldsymbol{\phi}) &= \sum_{t=0}^{T-1} S_{W'(t),t}(\boldsymbol{l}(W',t)) \\
&= \sum_{t=0}^{t_{\text{first}}} S_{W(t_{\text{first}}),t}(\boldsymbol{l}(W',t)) + \sum_{t=t_{\text{first}}+1}^{T-1} S_{W(t),t}(l_{W(t)}(W,t)) \\
&\geq \sum_{t=0}^{t_{\text{first}}} S_{W(t),t}(\boldsymbol{l}(W,t)) + \sum_{t=t_{\text{first}}+1}^{T-1} S_{W(t),t}(l_{W(t)}(W,t)) \\
&= P(W,\boldsymbol{\phi}).
\end{aligned}$$

Hence, as $P(W',\boldsymbol{\phi}) \geq P(W,\boldsymbol{\phi})$ for any $W \in \mathcal{W}$ we have

$$\max_{W\in\mathcal{W}} P(W,\boldsymbol{\phi}) = \max_{W\in\mathcal{W}'} P(W,\boldsymbol{\phi}),$$

concluding the proof of the lemma. □

From the proof of lemma 3.2.7 we are able to see when W' has a strictly better payoff than W against $\boldsymbol{\phi}$. That is $P(W',\boldsymbol{\phi}) > P(W,\boldsymbol{\phi})$ if and only if $t_{\text{first}} \geq m$ and $\varphi_{W(t_{\text{first}}),t} > 0$ for some $t \leq t_{\text{first}} - m$. Note that choosing an initial node to start at, $j \in N_A$, is equivalent to starting the patrol at the time $t = \min\{t \mid \varphi_{j,t} > 0\}$, as this is when the first potential attack at node j commences.

Now that we are restricted to some $W \in \mathcal{W}'$, that is a pure patroller response that is initially at some node $j \in N_A$, the walk is still relatively general and can waste time by wandering between nodes which are attacked. We can further restrict the pure patroller response to $\boldsymbol{\phi}$ to be one that takes a shortest path between any two consecutively attacked nodes.

Lemma 3.2.8. *For the game $G(Q,T,m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for all $\boldsymbol{\phi} \in \Phi$ we have*

$$V_{\bullet,\boldsymbol{\phi}}(Q,T,m) = \max_{W\in\mathcal{W}''} P(W,\boldsymbol{\phi}),$$

where $W'' = \{W \in \mathcal{W}' \mid t_2 - t_1 = d(W(t_1),W(t_2))$ where $t_2 = \min\{t \geq t_1 \mid W(t) \in N_A \setminus \{W(t_1)\}\} \quad \forall t_1 \in \mathcal{J}$ such that $W(t_1) \in N_A\}$.

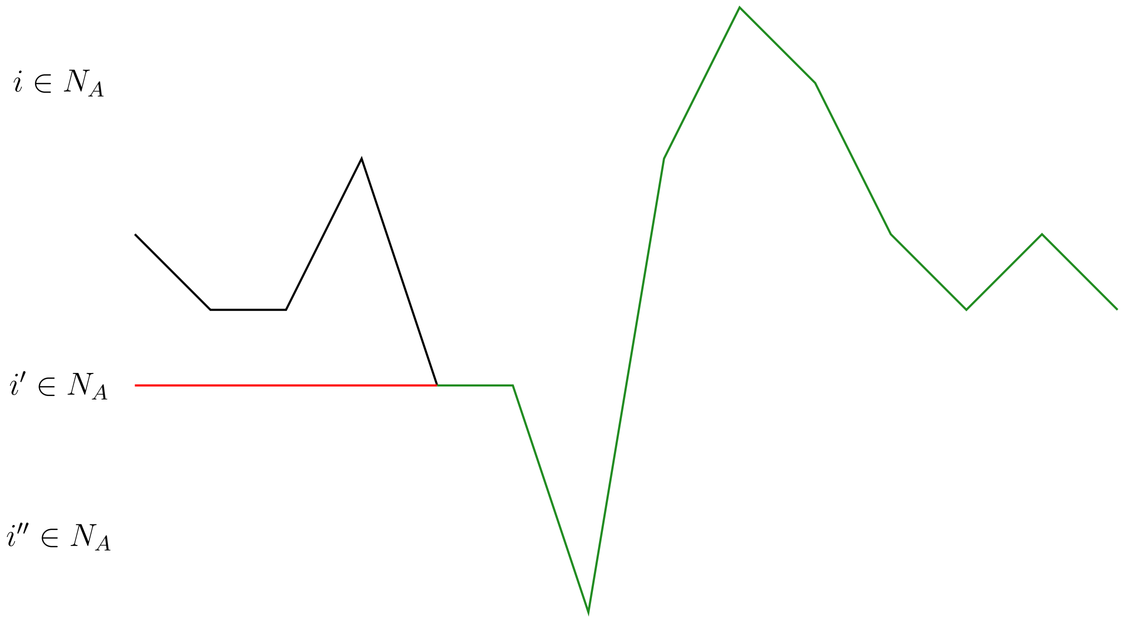


Figure 3.2.4: The original pure patroller walk (black), with the first node visited in N_A being i', can be improved initially by starting at $i' \in N_A$ (red), and the pure walks agree after arriving at i' (green).

The proof of lemma 3.2.8 follows by the construction of a pure strategy in $\mathcal{W}''$ from one in $\mathcal{W}'$ by changing all paths between consecutive nodes in N_A to a shortest path between them (with length equal to the distance between the nodes). Figure 3.2.5 shows the idea behind the proof.

Proof. For all $W \in \mathcal{W}'$ we seek to show there exists a $W' \in \mathcal{W}''$ such that $P(W', \boldsymbol{\phi}) \geq P(W, \boldsymbol{\phi})$.

For $W \in \mathcal{W}' \setminus \mathcal{W}''$ then there exists some $t_1 \in \mathcal{T}$ with $t_2 = \min\{t \geq t_1 \mid W(t) \in N_A \setminus \{W(t_1)\}\}$ such that $t_2 - t_1 > d(W(t_1), W(t_2))$. From $W \in \mathcal{W}'$ construct $W' \in \mathcal{W}''$ such that

$$W'(t) = \begin{cases} W(t) & \text{for } t \leq t_1 - 1, \\ P_{W(t_1),W(t_2)}(t - t_1) & \text{for } t_1 \leq t \leq t_1 + d(W(t_1), W(t_2)), \\ W(t_2) & \text{for } t_1 + d(W(t_1), W(t_2)) + 1 \leq t \leq t_2, \\ W(t) & \text{for } t \geq t_2 + 1, \end{cases}$$

where $P_{i,j}(t)$ is a shortest path between nodes i and j, with distance $d(i, j)$. For notational convenience let $i = W(t_1)$, $j = W(t_2)$ and $t_2' = t_1 + d(W(t_1), W(t_2))$. Immediately note that $\boldsymbol{l}(W', t) = \boldsymbol{l}(W, t)$ for all $t \leq t_2' - 1$ and for all $t \geq t_2$. In addition, $l_j(W', t) = t - 1$ for $t_2' + 1 \leq t \leq t_2$.

So we have that

$$\begin{aligned}
\sum_{t=t_2'}^{t_2} S_{W'(t),t}(\boldsymbol{l}(W',t)) &= \sum_{k=\max(t_2'-m+1,l_j(W,t),0)}^{t_2'} \varphi_{j,k} + \sum_{t=t_2'+1}^{t_2} \sum_{k=\max(t-m+1,t,0)}^{t} \varphi_{j,k} \\
&= \sum_{k=\max(t_2'-m+1,l_j(W,t_2'),0)}^{t_2'} \varphi_{j,k} + \sum_{t=t_2'+1}^{t_2} \varphi_{j,t} \\
&= \sum_{k=\max(t_2'-m+1,l_j(W,t_2'),0)}^{t_2} \varphi_{j,t} \\
&\geq \sum_{k=\max(t_2-m+1,l_j(W,t_2),0)}^{t_2} \varphi_{j,t} \\
&= \sum_{t=t_2'}^{t_2} S_{W(t),t}(\boldsymbol{l}(W,t)).
\end{aligned}$$

Therefore,

$$\begin{aligned}
P(W',\boldsymbol{\phi}) &= \sum_{t=0}^{T-1} S_{W'(t),t}(\boldsymbol{l}(W',t)) \\
&= \sum_{t=0}^{t_2'-1} S_{W'(t),t}(\boldsymbol{l}(W',t)) + \sum_{t=t_2'}^{t_2} S_{W'(t),t}(\boldsymbol{l}(W',t)) + \sum_{t=t_2+1}^{T-1} S_{W'(t),t}(\boldsymbol{l}(W',t)) \\
&= \sum_{t=0}^{t_2'-1} S_{W(t),t}(\boldsymbol{l}(W,t)) + \sum_{t=t_2'}^{t_2} S_{W'(t),t}(\boldsymbol{l}(W',t)) + \sum_{t=t_2+1}^{T-1} S_{W(t),t}(\boldsymbol{l}(W,t)) \\
&\geq \sum_{t=0}^{t_2'-1} S_{W(t),t}(\boldsymbol{l}(W,t)) + \sum_{t=t_2'}^{t_2} S_{W(t),t}(\boldsymbol{l}(W,t)) + \sum_{t=t_2+1}^{T-1} S_{W(t),t}(\boldsymbol{l}(W,t)) \\
&= \sum_{t=0}^{T-1} S_{W(t),t}(\boldsymbol{l}(W,t)) = P(W,\boldsymbol{\phi}).
\end{aligned}$$

Hence, as $P(W',\boldsymbol{\phi}) \geq P(W,\boldsymbol{\phi})$ for any $W \in \mathcal{W}' \setminus \mathcal{W}''$, repeating this construction process until we get some $W'' \in \mathcal{W}''$ for which we known $P(W'',\boldsymbol{\phi}) \geq P(W,\boldsymbol{\phi})$ for any $W \in \mathcal{W}' \setminus \mathcal{W}''$. Therefore,

$$\max_{W \in \mathcal{W}'} P(W,\boldsymbol{\phi}) = \max_{W \in \mathcal{W}''} P(W,\boldsymbol{\phi})$$

and along with lemma 3.2.7 the proof of the lemma is concluded.

□

We now define $d(i,j,N_A)$ as the distance between nodes $i,j \in N$, which does not use nodes in N_A. $d(i,j,N_A)$ is required as we need to considered the distance

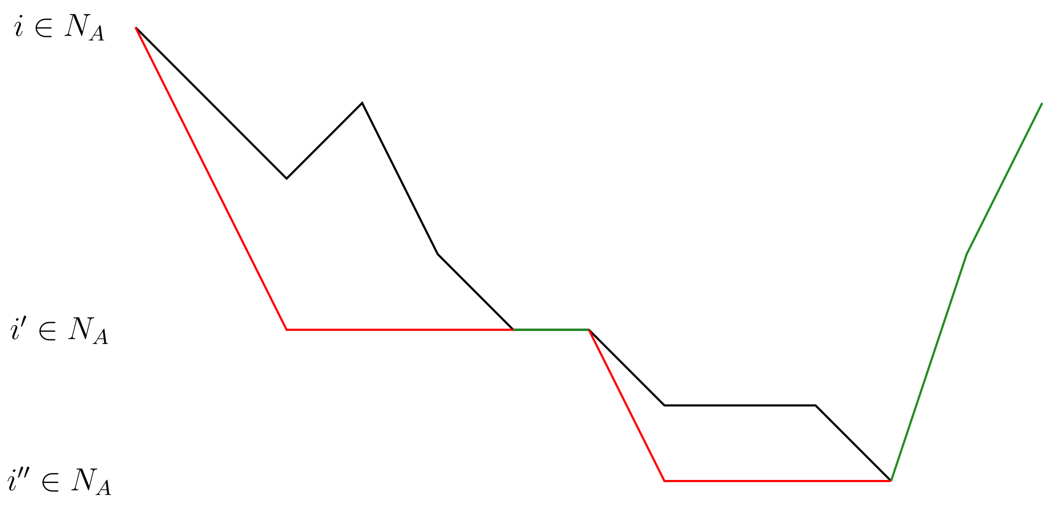


Figure 3.2.5: The original pure patroller walk (black) can be improved by replacing any path between any two potentially attacked nodes with the shortest path. Here i to i' and i' to i'' are replaced by the shortest path between them followed by waiting (red). The improvement follows the same path outside of these improvements (green)

between consecutively chosen nodes. By lemma 3.2.8 in order to find the performance of an attacker strategy we need only consider walks in the set $\mathcal{W}''$. Any walk $W \in \mathcal{W}''$ can be written as

$$W(t) = \begin{cases} j_1 & \text{if } t_1 = 0 \leq t \leq \nu_1, \\ P_{j_1,j_2}(t-\nu_1) & \text{if } \nu_1 \leq t \leq t_2, \\ j_2 & \text{if } t_2 \leq t \leq t_2+\nu_2, \\ P_{j_2,j_3}(t-t_2-\nu_2) & \text{if } t_2+\nu_2 \leq t \leq t_3, \\ \vdots & \vdots \\ P_{j_{k-1},j_k}(t-t_{k-1}-\nu_{k-1}) & \text{if } t_{k-1}+\nu_{k-1} \leq t \leq t_k, \\ j_k & \text{if } t_k \leq t \leq t_k+\nu_k = T-1, \end{cases} \tag{3.5}$$

for some $j_i \in N_A$, $\nu_i \in \{0, ..., T-1-t_i\}$ for all $i \in \{1, ..., k\}$ for some $k \in \mathbb{N}$. In equation (3.5) t_i, the time of arrival at node j_k is given by

$$t_i = \begin{cases} 0 & \text{if } i = 1, \\ t_{i-1}+\nu_{i-1}+d(j_{i-1}, j_i, N_A) & \text{if } i \in \{2, 3, ..., k\}, \end{cases}$$

and $P_{j,j'}(t)$ for $t \in \{0, ..., d(j, j', N_A)\}$ is a shortest path from j to j' such that $P_{j,j'}(0) = j$, $P_{j,j'}(d(j, j', N_A)) = j'$ and $P_{j,j'}(t) \notin N_A$ for all $t \in \{1, ..., d(j, j', N_A) - 1\}$, where $d(j, j', N_A)$ is the length of such a path $P_{j,j'}(t)$.

As any $W \in \mathcal{W}''$ can be written as in equation (3.5) for some choice of $j_i \in N_A$, some choice of $\nu_i \in \{0, ..., T-1-t_i\}$ for $i = 1, ..., k$ and some choices of shortest path for each $P_{j_i,j_{i+1}}$ for $i = 1, ..., k-1$. Immediately we can remove all but an arbitrary choice of shortest paths $P_{j_i,j_{i+1}}$ for $i = 1, ..., k-1$ as $P(0) = j_1 \in N_A$, $P(d(j_i, j_{i+1}, N_A)) = j_{i+1} \in N_A$ are necessary by definition and the real choice is

$P(t) \notin N_A$ for $1 \leq t \leq d(j_i, j_{i+1}, N_A) - 1$. As $\varphi_{j,t} = 0$ for all $j \notin N_A$, this choice of shortest path does not affect the payoff of walks. Therefore, it is possible to write a walk $W \in \mathcal{W}''$ in an equivalent form considering only the choice of $j_i \in N_A$ and $\nu_i \in \{0, ..., T-1-t_i\}$ for $i = 1, ..., k$.

Definition 3.2.9. A walk $W \in \mathcal{W}''$, as in equation (3.5), has *move-wait* form which is

$$\omega = ((j_1, \nu_1), (j_2, \nu_2) ..., (j_k, \nu_k))$$

for some node, waiting time pair $(j_i, \nu_i) \in N_A \times \mathcal{J}$ for $i = 1, ..., k$ for some $k \in \mathbb{N}$ such that

- $j_i \neq j_{i+1}$ for all $i \in \{1, ...k-1\}$,
- $P_{j_i, j_{i+1}}(\{1, ..., d(j_i, j_{i+1}, N_A) - 1\}) \cap N_A = \emptyset$ for all $i \in \{0, ..., k-1\}$,
- $\nu_k + \sum_{i=1}^{k-1} (\nu_i + d(j_i, j_{i+1}, N_A)) = T - 1$.

For a given move-wait walk ω the arrival time of the visit to j_i is given by $t_i(\omega) = \sum_{r=1}^{i-1} (\nu_r + d(j_r, j_{r+1}, N_A))$. Let Ω be the set of move-wait walks.

When required we write $j_i(\omega)$ and $\nu_1(\omega)$ to be the i^{th} node and wait time of move-wait walk ω. In addition $k(\omega)$ is the number of nodes visited in the move-wait walk ω.

The payoff for choosing $\omega \in \Omega$ as a response to $\boldsymbol{\phi} \in \Phi$ is given by

$$P(\omega, \boldsymbol{\phi}) = \sum_{i=1}^{k} \left(S_{j_i, t_i(\omega)}(l_{j_i}(\omega)) + \sum_{t=t_i(\omega)+1}^{t_i(\omega)+\nu_i} \varphi_{j_i,t} \right) = \sum_{i=1}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_i,t}, \tag{3.6}$$

where

$$n_i(\omega) = \max(t_i(\omega) - m + 1, l_i(\omega) + 1, 0),$$

is the new start time, in which

$$l_i(\omega) = \max_{x=1,...,i-1 \text{ s.t } j_x = j_i} (t_x + \nu_x)$$

is the last visit time for node j_i, before the current arrival time $t_i(\omega)$. As previously mentioned the chosen shortest paths do not affect the payoff and so choosing $W \in \mathcal{W}''$ is equivalent to choosing $\omega \in \Omega$ and hence we may write $P(W, \boldsymbol{\phi}) = P(\omega, \boldsymbol{\phi})$ for all $W \in \mathcal{W}''$ with the corresponding move-wait form $\omega \in \Omega$. Therefore the performance of $\boldsymbol{\phi} \in \Phi$ is given by

$$V_{\bullet, \boldsymbol{\phi}}(Q, T, m) = \max_{\omega \in \Omega} P(\omega, \boldsymbol{\phi}), \tag{3.7}$$

which is much easier to compute than equation (2.13). Further to this it is possible to consider how slight changes to any $\omega \in \Omega$ affect the payoff in equation (3.6).

A simple consideration is the manipulation of the amount of time a walk waits at each node. For some $i \in \{1, ..., k\}$ such that $\nu_i > 0$ we can consider decreasing this waiting at j_i to $\nu_i - 1$ and increasing the waiting at either j_{i-1} to $\nu_{i-1} + 1$ or j_{i+1} to $\nu_{i+1} + 1$. That is for

$$\omega = ((j_1, \nu_1), ..., (j_{r-1}, \nu_{r-1}), (j_r, \nu_r), (j_{r+1}, \nu_{r+1}), ..., (j_k, \nu_k)) \tag{3.8}$$

we can consider a forward transfer of waiting in the move-wait walk

$$\omega^f = ((j_1, \nu_1), ..., (j_{r-1}, \nu_{r-1}), (j_r, \nu_r - 1), (j_{r+1}, \nu_{r+1} + 1), ..., (j_k, \nu_k)) \tag{3.9}$$

and a backwards transfer of waiting in the move-wait walk

$$\omega^b = ((j_1, \nu_1), ..., (j_{r-1}, \nu_{r-1} + 1), (j_r, \nu_r - 1), (j_{r+1}, \nu_{r+1}), ..., (j_k, \nu_k)). \tag{3.10}$$

Theorem 3.2.10. *For the game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for all $\boldsymbol{\phi} \in \Phi$ we have*

- ***Forward Transfer:***

$$P(\omega^f, \boldsymbol{\phi}) \begin{cases} > P(\omega, \boldsymbol{\phi}) & \text{if } G^f(\omega, r) > L^f(\omega, r), \\ = P(\omega, \boldsymbol{\phi}) & \text{if } G^f(\omega, r) = L^f(\omega, r), \\ < P(\omega, \boldsymbol{\phi}) & \text{if } G^f(\omega, r) < L^f(\omega, r), \end{cases}$$

for all $\omega \in \Omega$, where ω^f is a forward transfer move-wait walk, as in equation (3.9) for some $r \in \{0, ..., k-1\}$ where $\nu_r > 0$ for ω, as in equation (3.8), and where the gain is

$$G^f(\omega, r) = \mathbb{I}_{\{E_1(\omega,r)\}} \varphi_{j_{r+1}, t_{r+1}(\omega) - m},$$

in which $E_1(\omega, r)$ is the event that $t_{r+1}(\omega) - m + 1 > \max(l_{r+1}(\omega) + 1, 0)$, and the loss is

$$L^f(\omega, r) = \mathbb{I}_{\{E_2(\omega,r)\}} \varphi_{j_r, t_r(\omega) + \nu_r(\omega)},$$

in which $E_2(\omega, r)$ is the event that either; $\{i \in \{r+2, ..., k\} \mid j_i = j_r\} = \emptyset$ or; both $\{i \in \{r+2, ..., k\} \mid j_i = j_r\} \neq \emptyset$ and $t_r(\omega) + \nu_r(\omega) < \max(t_{\min\{i \in \{r+2,...,k\} | j_i = j_r\}}(\omega) - m + 1, 0)$.

- ***Backwards Transfer:***

$$P(\omega^b, \boldsymbol{\phi}) \begin{cases} > P(\omega, \boldsymbol{\phi}) & \text{if } G^b(\omega, r) > L^b(\omega, r), \\ = P(\omega, \boldsymbol{\phi}) & \text{if } G^b(\omega, r) = L^b(\omega, r), \\ < P(\omega, \boldsymbol{\phi}) & \text{if } G^b(\omega, r) < L^b(\omega, r), \end{cases}$$

for all $\omega \in \Omega$ where ω^b is a backward transfer move-wait walk, as in equation (3.10) for some $r \in \{2, ..., k\}$ where $\nu_r > 0$ for ω, as in equation (3.8) and where the gain is

$$G^b(\omega, r) = L^f(\omega^b, r-1) = \mathbb{I}_{\{E_3(\omega,r)\}} \varphi_{j_{r-1}, t_{r-1}(\omega) + \nu_{r-1}(\omega) + 1},$$

in which $E_3(\omega, r)$ *is the event that either;* $\{i \in \{r+1, ..., k\} \mid j_i = j_r\} = \emptyset$ *or; both* $\{i \in \{r+1, ..., k\} \mid j_i = j_r\} \neq \emptyset$ *and* $t_{r-1}(\omega) + \nu_{r-1}(\omega) + 1 < \max(t_{\min\{i \in \{r+1,...,k\} | j_i = j_r\}}(\omega) - m + 1, 0)$ *and the loss is*

$$L^b(\omega, r) = G^f(\omega^b, r-1) = \mathbb{I}_{\{E_4(\omega,r)\}} \varphi_{j_r, t_r(\omega)-m+1},$$

in which $E_4(\omega, r)$ *is the event that* $t_r(\omega) - m + 2 > \max(l_r(\omega) + 1, 0)$.

We note the use of the indicator function in theorem 3.2.10 in the gains G^f, G^b and losses L^f, L^b as it possible that these gains and losses are already caught elsewhere in the walk. The proof of theorem 3.2.10 follows by comparing $P(\omega, \boldsymbol{\phi})$ to $P(\omega^f, \boldsymbol{\phi})$ and $P(\omega^b, \boldsymbol{\phi})$. Figure 3.2.6 shows the idea behind the proof for a forward transfer of waiting.

Proof. For forwards transfer ω^f let $X = \{i \in \{r+2, ..., k\} \mid j_i = j_r\}$ then for all $i \in \{1, ..., k\}$ we have

$$t_i(\omega^f) = \begin{cases} t_{r+1}(\omega) - 1 & \text{if } i = r+1 \\ t_i(\omega) & \text{otherwise,} \end{cases}$$

and

$$l_i(\omega^f) = \begin{cases} l_i(\omega) - 1 & \text{if } X \neq \emptyset \text{ and } i = \min X, \\ l_i(\omega) & \text{otherwise,} \end{cases}$$

and hence

$$n_i(\omega^f) \begin{cases} \max(t_i(\omega) - m + 1, l_i(\omega), 0) & \text{if } X \neq \emptyset \text{ and } i = \min X, \\ \max(t_{r+1}(\omega) - m, l_{r+1}(\omega) + 1, 0) & \text{if } i = r+1, \\ n_i(\omega) & \text{otherwise.} \end{cases}$$

For ease of notation we write $x = \min X$ if $X \neq \emptyset$.

Now we calculate the payoff of ω^f against $\boldsymbol{\phi}$ (as in equation 3.6) is

$$P(\omega^f, \boldsymbol{\phi}) = \sum_{i=1}^{k} \sum_{t=n_i(\omega^f)}^{t_i(\omega^f)+\nu_i(\omega^f)} \varphi_{j_i,t} = A_1 + A_2 + A_3 + A_4,$$

where $A_1 = \sum_{i=1}^{r-1} \sum_{t=n_i(\omega^f)}^{t_i(\omega^f)+\nu_i(\omega^f)} \varphi_{j_i,t}$, $A_2 = \sum_{t=n_r(\omega^f)}^{t_r(\omega^f)+v_r(\omega^f)} \varphi_{j_r,t}$, $A_3 = \sum_{t=n_{r+1}(\omega^f)}^{t_{r+1}(\omega^f)+v_{r+1}(\omega^f)} \varphi_{j_{r+1},t}$ and $A_4 = \sum_{i=r+2}^{k} \sum_{t=n_i(\omega^f)}^{t_i(\omega^f)+\nu_i(\omega^f)} \varphi_{j_i,t}$. We now manipulate A_1,A_2,A_3,A_3 to achieve the desired result.

$$A_1 = \sum_{i=1}^{r-1} \sum_{t=n_i(\omega^f)}^{t_i(\omega^f)+\nu_i(\omega^f)} \varphi_{j_i,t} = \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t}.$$

$$A_2 = \sum_{t=n_r(\omega)}^{t_r(\omega)+v_r(\omega)-1} \varphi_{j_r,t} = \sum_{t=n_r(\omega)}^{t_r(\omega)+v_r(\omega)} \varphi_{j_r,t} - \varphi_{j_r,t_r(\omega)+v_r(\omega)}.$$

$$\begin{aligned}
A_3 &= \sum_{t=n_{r+1}(\omega^f)}^{t_{r+1}(\omega^f)+v_{r+1}(\omega^f)} \varphi_{j_{r+1},t} = \sum_{t=\max(t_{r+1}(\omega)-m,l_{r+1}(\omega)+1,0)}^{t_{r+1}(\omega)-1+v_{r+1}(\omega)+1} \varphi_{j_{r+1},t} \\
&= \mathbb{I}_{\{t_{r+1}(\omega)-m+1>\max(l_{r+1}(\omega)+1,0)\}}\varphi_{t_{r+1}(\omega)-m} + \sum_{t=\max(t_{r+1}(\omega)-m+1,l_{r+1}(\omega)+1,0)}^{t_{r+1}(\omega)+v_{r+1}(\omega)} \varphi_{j_{r+1},t} \\
&= G^f(\omega,r) + \sum_{t=n_{r+1}(\omega)}^{t_{r+1}(\omega)+v_{r+1}(\omega)} \varphi_{j_{r+1},t}.
\end{aligned}$$

$$\begin{aligned}
A_4 &= \sum_{i=r+2}^{k} \sum_{t=n_i(\omega^f)}^{t_i(\omega^f)+\nu_i(\omega^f)} \varphi_{j_i,t} \\
&= \mathbb{I}_{\{X\neq\emptyset\}} \left(\sum_{t=\max(t_x(\omega)-m+1,l_x(\omega),0)}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x,t} + \sum_{i\in\{r+2,\dots,k\}\setminus\{x\}} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \right) \\
&\quad + \mathbb{I}_{\{X=\emptyset\}} \sum_{i=r+2}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \\
&= \mathbb{I}_{\{X\neq\emptyset\}} \Bigg(\mathbb{I}_{\{l_x(\omega)+1>\max(t_x(\omega)-m+1,0)\}}\varphi_{j_x,l_x(\omega)} + \sum_{t=\max(t_x(\omega)-m+1,l_x(\omega)+1,0)}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x,t} \\
&\quad + \sum_{i\in\{r+2,\dots,k\}\setminus\{x\}} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \Bigg) + \mathbb{I}_{\{X=\emptyset\}} \sum_{i=r+2}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \\
&= \mathbb{I}_{\{X\neq\emptyset\}} \left(\mathbb{I}_{\{l_x(\omega)+1>\max(t_x(\omega)-m+1,0)\}}\varphi_{j_x,l_x(\omega)} + \sum_{i=r+2}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \right) \\
&\quad + \mathbb{I}_{\{X=\emptyset\}} \sum_{i=r+2}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \\
&= \mathbb{I}_{\{X\neq\emptyset \text{ and } t_r(\omega)+v_r(\omega)+1>\max(t_{\min X}(\omega)-m+1,0)\}}\varphi_{j_r,t_r(\omega)+v_r(\omega)} + \sum_{i=r+2}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t} \\
&= (1-\mathbb{I}_{E_2})\varphi_{j_r,t_r(\omega)+v_r(\omega)} + \sum_{i=r+2}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i,t}.
\end{aligned}$$

Hence we have

$$\begin{aligned} P(\omega^f, \boldsymbol{\phi}) &= P(\omega, \boldsymbol{\phi}) - \varphi_{j_r, t_r(\omega)+v_r(\omega)} + G^f(\omega, r) + (1 - \mathbb{I}_{E_2})\varphi_{j_r, t_r(\omega)+v_r(\omega)} \\ &= P(\omega, \boldsymbol{\phi}) + G^f(\omega, r) - L^f(\omega, r) \end{aligned}$$

and thus the results for forward transfer in the theorem are obtained.

For the results on the backward transfer we note that ω is a forward transfer on ω^b with the index $r-1$ (instead of r). Hence we have

$$P(\omega, \boldsymbol{\phi}) \begin{cases} > P(\omega^b, \boldsymbol{\phi}) & \text{if } G^f(\omega^b, r-1) > L^f(\omega^b, r-1), \\ = P(\omega^b, \boldsymbol{\phi}) & \text{if } G^f(\omega^b, r-1) = L^f(\omega^b, r-1), \\ < P(\omega^b, \boldsymbol{\phi}) & \text{if } G^f(\omega^b, r-1) < L^f(\omega^b, r-1). \end{cases}$$

Equivalently,

$$P(\omega^b, \boldsymbol{\phi}) \begin{cases} < P(\omega, \boldsymbol{\phi}) & \text{if } G^f(\omega^b, r-1) > L^f(\omega^b, r-1), \\ = P(\omega, \boldsymbol{\phi}) & \text{if } G^f(\omega^b, r-1) = L^f(\omega^b, r-1), \\ > P(\omega, \boldsymbol{\phi}) & \text{if } G^f(\omega^b, r-1) < L^f(\omega^b, r-1), \end{cases}$$

and thus the results for the backwards transfer in the theorem are obtained.

□

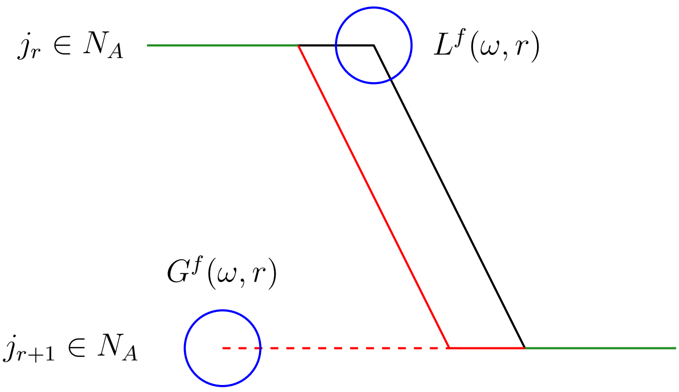


Figure 3.2.6: The loss and gain by transferring forward one unit of waiting time from ω, shown in black, creating ω^f, shown in red. ω and ω^f are the same for the majority of the walk shown in green with the loss $L^f(\omega, r)$ and gain $G^f(\omega, r)$ for the forward transfer shown as circled in blue.

While theorem 3.2.10 provides the most generalised condition for waiting transfer, it is rather complex to apply. However, we can consider simpler cases, such as where there is definitely no loss in forward transfer to get reductions to the set Ω for the performance of an arbitrary attacker strategy.

Lemma 3.2.11. *For the game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for all $\boldsymbol{\phi} \in \Phi$ we have*

$$V_{\bullet, \boldsymbol{\phi}}(Q, T, m) = \max_{\omega \in \Omega'} P(\omega, \boldsymbol{\phi}),$$

where $\Omega' = \{\omega \in \Omega \mid \nu_r(\omega) = 0$ for all $r \in \{1, ..., k-1\}$ such that $\varphi_{j_r, t} = 0$ for all $t \geq t_r(\omega) + 1\}$.

The proof of lemma 3.2.11 follows by a forward transfer of any $r \in \{1, ..., k\}$ such that $\nu_r(\omega) \geq 1$ and $\varphi_{j_r,t} = 0$ for all $t \geq t_r$ and applying theorem 3.2.10.

Proof. We first aim to show that

$$P(\omega', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$$

for all $\omega \in \Omega$ such that there exists some $r \in \{1, ..., k-1\}$ such that $\nu_r \geq 1$ and $\varphi_{j_r,t} = 0$ for all $t \geq t_r(\omega) + 1$, where ω' is such that $\nu_i(\omega') = \nu_i(\omega)$ for $i \neq r$ and $\nu_r(\omega') = 0$.

This follows from theorem 3.2.10 by sequentially constructing ω_x by forward transferring the waiting time for the r^{th} index. So that $\omega_x = \omega_{x-1}^f$ for $x \in \{1, ..., \nu_r(\omega)\}$ and $\omega_0 = \omega$. As $\varphi_{j_r,t} = 0$ for all $t \geq t_r(\omega) + 1$ we have $L^f(\omega_x, r) = 0$ for all $x = 1, ..., \nu_r(\omega)$ and hence

$$P(\omega_x, \boldsymbol{\phi}) \geq P(\omega_{x-1}, \boldsymbol{\phi}),$$

for all $x \in \{1, ..., \nu_r(\omega)\}$ and noting $\omega_{\nu_r(\omega)} = \omega'$ we arrive at the result that $P(\omega', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$. Hence repeating this process as required gives

$$\max_{\omega \in \Omega} P(\omega, \boldsymbol{\phi}) = \max_{\omega \in \Omega'} P(\omega, \boldsymbol{\phi}), \tag{3.11}$$

and along with equation (3.7) the proof is concluded. □

A further reduction of the search space Ω', as in lemma 3.2.11, can be made by considering the initial waiting time given a chosen initial node j_1. For $\boldsymbol{\phi} \in \Phi$ we define

$$\mathcal{T}(j, \boldsymbol{\phi}) = \{\tau \in \mathcal{T} \mid \varphi_{j,\tau} \neq 0\},$$

to be the set, for node $j \in N$, of commencement times for which there is a potential attack at under $\boldsymbol{\phi}$.

Lemma 3.2.12. *For the game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for all $\boldsymbol{\phi} \in \Phi$ we have*

$$V_{\bullet,\boldsymbol{\phi}}(Q, T, m) = \max_{\omega \in \Omega''} P(\omega, \boldsymbol{\phi}),$$

where $\Omega'' = \{\omega \in \Omega' \mid \nu_1(\omega) \geq \min \mathcal{T}(j_1, \boldsymbol{\phi})\}$.

The proof of lemma 3.2.12 follows by constructing a walk which skips this initial node if $\nu_1(\omega) < \min \mathcal{T}(j_1, \boldsymbol{\phi})$.

Proof. We aim to show that

$$P(\omega', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$$

for all $\omega \in \Omega' \setminus \Omega''$ where

$$\omega' = ((j_2, \nu_1 + d(j_1, j_2, N_A) + \nu_2), (j_3, \nu_3), ..., (j_k, \nu_k)).$$

We have for the walk ω' that

$$t_i(\omega') = \begin{cases} 0 & \text{if } i = 1, \\ t_{i+1}(\omega) & \text{if } i \geq 2, \end{cases}$$

and letting $X = \{i \in \{2, ..., k\} \mid j_i = j_1\}$ we have

$$l_i(\omega') = \begin{cases} -m & \text{if } X \neq \emptyset \text{ and } i = \min X, \\ l_{i+1}(\omega) & \text{otherwise.} \end{cases}$$

Then

$$n_i(\omega') = \begin{cases} 0 & \text{if } i = 1, \\ \max(t_{i+1}(\omega) - m + 1, -m + 1, 0) & \text{if } X \neq \emptyset \text{ and } i = \min X, \\ n_{i+1}(\omega) & \text{otherwise,} \end{cases}$$

and so $n_i(\omega') \geq n_{i+1}(\omega)$ for all $1 \leq i \leq k - 1$. Then as $\varphi_{j_1,t} = 0$ for all $t \leq \nu_1$ we have

$$\begin{aligned} P(\omega', \boldsymbol{\phi}) &= \sum_{i=1}^{k(\omega')} \sum_{t=n_i(\omega')}^{t_i(\omega')+v_i(\omega')} \varphi_{j_i(\omega'),t} \\ &= \sum_{t=0}^{\nu_1(\omega)+d(j_1(\omega),j_2(\omega),N_A)+\nu_2(\omega)} \varphi_{j_2(\omega),t} + \sum_{i=3}^{k(\omega)} \sum_{t=n_i(\omega)}^{t_i(\omega)+v_i(\omega)} \varphi_{j_i(\omega),t} \\ &= \sum_{t=0}^{\nu_1(\omega)} \varphi_{j_1(\omega),t} + \sum_{t=0}^{\nu_1(\omega)+d(j_1(\omega),j_2(\omega),N_A)+\nu_2(\omega)} \varphi_{j_2(\omega),t} + \sum_{i=3}^{k(\omega)} \sum_{t=n_i(\omega)}^{t_i(\omega)+v_i(\omega)} \varphi_{j_i(\omega),t} \\ &\geq \sum_{i=1}^{k(\omega)} \sum_{t=n_i(\omega)}^{t_i(\omega')+v_i(\omega)} \varphi_{j_i(\omega),t} = P(\omega, \boldsymbol{\phi}). \end{aligned}$$

Hence repeating this as required we can get a $\omega'' \in \Omega''$ for any $\omega \in \Omega'$ such that $P(\omega'', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$. Therefore,

$$\max_{\omega \in \Omega'} P(\omega, \boldsymbol{\phi}) = \max_{\omega \in \Omega''} P(\omega, \boldsymbol{\phi})$$

and along with lemma 3.2.11 the proof is concluded. □

We can get further reductions by considering which nodes should not be visited. We already know that only nodes in N_A have a non-zero probability of catching an attacker at, but further to this we can consider if there is still a non-zero probability at time t given a walks prior choices. We define

$$N_A(\omega, x) = \{j \in N_A \mid \exists s \geq t_x(\omega) + \nu_x(\omega) + d(j_x, j, N_A) - m + 1 \text{ s.t. } \varphi_{j,s} > 0\},$$

to be the set of nodes with a non-zero probability of catching the attacker under walk ω, when leaving node j_x (at time $t_x(\omega)+\nu_x(\omega)$) and arriving at the node $j \in N_A(\omega, x)$.

Theorem 3.2.13. *For the game $G(Q,T,m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for all $\boldsymbol{\phi} \in \Phi$ we have*

$$V_{\bullet,\boldsymbol{\phi}}(Q,T,m) = \max_{\omega \in \Omega'''} P(\omega, \boldsymbol{\phi}),$$

where $\Omega''' = \{\omega \in \Omega'' \mid j_{r'}, ..., j_k \in N_A(\omega, r'-1)\, \forall r' \in \{2,...,k\}\}$.

The proof of lemma 3.2.13 follows by constructing a walk that skips any node $j_r \notin N_A(\omega, r')$ for some $r' \geq r$.

Proof. We first aim to show that it is possible to construct a walk ω' such that

$$P(\omega', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$$

for all $\omega \in \Omega''$. Consider any $\omega \in \Omega'' \setminus \Omega'''$ then there exists some r such that $j_r \notin N_A(\omega, r-1)$. Then to construct a walk that skips j_r we consider three cases.

Firstly if $r = k$ then we construct ω' such that $j_i(\omega') = j_i(\omega)$ for all $i \in \{1,...,k-1\}$ and $\nu_i(\omega') = \nu_i(\omega)$ for all $i \in \{1,...,k-2\}$ and $\nu_{k-1}(\omega') = \nu_{k-1}(\omega) + t_{\text{ext}}$ where $t_{\text{ext}} = d(j_{k-1}(\omega), j_k(\omega), N_A) + \nu_k(\omega)$. So $t_i(\omega') = t_i(\omega)$ for all $i \in \{1,...,k-1\}$, $l_i(\omega') = l_i(\omega)$ for all $i \in \{1,...,k-1\}$ and $n_i(\omega') = n_i(\omega)$ for all $i \in \{1,...,k-1\}$. As $\varphi_{j_k(\omega),t} = 0$ for all $t \geq t_k(\omega)$ we have

$$\begin{aligned}
P(\omega', \boldsymbol{\phi}) &= \sum_{i=1}^{k-1} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{k-2} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_{k-1}(\omega)}^{t_{k-1}(\omega)+\nu_{k-1}(\omega)+t_{\text{ext}}} \varphi_{j_{k-1}(\omega),t} \\
&= \sum_{i=1}^{k-2} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_{k-1}(\omega)}^{t_{k-1}(\omega)+\nu_{k-1}(\omega)+t_{\text{ext}}} \varphi_{j_{k-1}(\omega),t} + \sum_{t=n_k(\omega)}^{t_k(\omega)+\nu_k(\omega)} \varphi_{j_k(\omega),t} \\
&\geq \sum_{i=1}^{k-2} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_{k-1}(\omega)}^{t_{k-1}(\omega)+\nu_{k-1}(\omega)} \varphi_{j_{k-1}(\omega),t} + \sum_{t=n_k(\omega)}^{t_k(\omega)+\nu_k(\omega)} \varphi_{j_k(\omega),t} \\
&= \sum_{i=1}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} = P(\omega, \boldsymbol{\phi}).
\end{aligned}$$

Secondly if $r \neq k$ and $j_{r-1} = j_{r+1}$ then we can construct a walk ω' such that

$$j_i(\omega') = \begin{cases} j_i(\omega) & \text{if } i \leq r-1, \\ j_{i+2}(\omega) & \text{if } r \leq i \leq k-2, \end{cases}$$

and

$$\nu_i(\omega') = \begin{cases} \nu_i(\omega) & \text{if } i \leq r-2, \\ \nu_{r-1}(\omega) + t_{\text{ext}} & \text{if } i = r-1, \\ \nu_{i+2}(\omega) & \text{if } r \leq i \leq k-2, \end{cases}$$

in which $t_{\text{ext}} = \nu_r(\omega) + d(j_{r-1}(\omega), j_r(\omega), N_A) + \nu_{r+1}(\omega) + d(j_r(\omega), j_{r+1}(\omega), N_A)$. Therefore,

$$t_i(\omega') = \begin{cases} t_i(\omega) & \text{if } i \leq r-1, \\ t_{i+2}(\omega) & \text{if } r \leq i \leq k-1. \end{cases}$$

and with $X = \{i \in \{r+2, ..., k\} \mid j_i(\omega) = j_r(\omega)\}$ and if $X \neq \emptyset$ let $x = \min X$ then

$$l_i(\omega') = \begin{cases} l_i(\omega) & \text{if } i \leq r-1, \\ l_r(\omega) & \text{if } X \neq \emptyset \text{ and } i = x-2, \\ l_{i+2}(\omega) & \text{otherwise.} \end{cases}$$

Hence,

$$n_i(\omega') = \begin{cases} n_i(\omega) & \text{if } i \leq r-1, \\ \max(t_x(\omega) - m + 1, l_r(\omega) + 1, 0) & \text{if } X \neq \emptyset \text{ and } i = x-2, \\ n_{i+2}(\omega) & \text{otherwise.} \end{cases}$$

We know that $\varphi_{j_r(\omega),t} = 0$ for all $t \geq n_r(\omega)$. In particular $\varphi_{j_x(\omega),t} = 0$ for all

$t \geq n_x(\omega)$ and $\varphi_{j_{x-2}(\omega'),t} = 0$ for all $t \geq n_{x-2}(\omega')$. Therefore we have

$$\begin{aligned}
P(\omega', \boldsymbol{\phi}) &= \sum_{i=1}^{k-2} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{r-2} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} + \sum_{t=n_{r-1}(\omega')}^{t_{r-1}(\omega')+\nu_{r-1}(\omega')} \varphi_{j_{r-1}(\omega'),t} + \sum_{i=r}^{k-2} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{r-2} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_{r-1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r-1}(\omega),t} + \sum_{i=r}^{k-2} \sum_{t=n_{i+2}(\omega)}^{t_{i+2}(\omega)+\nu_{i+2}(\omega)} \varphi_{j_{i+2}(\omega),t} \\
&\quad - \mathbb{I}_{\{X \neq \emptyset\}} \left(\sum_{t=n_x(\omega)}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} - \sum_{t=n_{x-2}(\omega')}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} \right) \\
&= \sum_{i=1}^{r-2} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{i=r}^{k-2} \sum_{t=n_{i+2}(\omega)}^{t_{i+2}(\omega)+\nu_{i+2}(\omega)} \varphi_{j_{i+2}(\omega),t} \\
&\quad + \sum_{t=n_{r-1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r-1}(\omega),t} + \sum_{t=n_r(\omega)}^{t_r(\omega)+\nu_r(\omega)} \varphi_{j_r(\omega),t} \\
&\geq \sum_{i=1}^{r-2} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{i=r}^{k-2} \sum_{t=n_{i+2}(\omega)}^{t_{i+2}(\omega)+\nu_{i+2}(\omega)} \varphi_{j_{i+2}(\omega),t} \\
&\quad + \sum_{t=n_{r-1}(\omega)}^{t_{r-1}(\omega)+\nu_{r-1}(\omega)} \varphi_{j_{r-1}(\omega),t} + \sum_{t=n_{r+1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+1}(\omega),t} + \sum_{t=n_r(\omega)}^{t_r(\omega)+\nu_r(\omega)} \varphi_{j_r(\omega),t} \\
&= \sum_{i=1}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} = P(\omega, \boldsymbol{\phi}).
\end{aligned}$$

Finally if $r \neq k$ and $j_{r-1} \neq j_{r+1}$ then we can construct a walk ω' such that

$$j_i(\omega') = \begin{cases} j_i(\omega) & \text{if } i \leq r-1, \\ j_{i+1}(\omega) & \text{if } r \leq i \leq k-1, \end{cases}$$

and

$$\nu_i(\omega') = \begin{cases} \nu_i(\omega) & \text{if } i \leq r-1, \\ \nu_r(\omega) + t_{\text{ext}} & \text{if } i = r, \\ \nu_{i+1}(\omega) & \text{if } r+1 \leq i \leq k-1, \end{cases}$$

in which $t_{\text{ext}} = t_{r+1}(\omega) - t_{r-1}(\omega) - \nu_{r-1}(\omega) - d(j_r(\omega), j_{r+1}(\omega), N_A)$. Therefore,

$$t_i(\omega') = \begin{cases} t_i(\omega) & \text{if } i \leq r-1, \\ t_r(\omega) - t_{\text{ext}} & \text{if } i = r, \\ t_{i+1}(\omega) & \text{if } r+1 \leq i \leq k-1. \end{cases}$$

and with $X = \{i \in \{r+2, ..., k\} \mid j_i(\omega) = j_r(\omega)\}$ and if $X \neq$ letting $x = \min X$ then

$$l_i(\omega') = \begin{cases} l_i(\omega) & \text{if } i \leq r-1, \\ l_r(\omega) & \text{if } X \neq \emptyset \text{ and } i = x-1, \\ l_{i+1}(\omega) & \text{otherwise.} \end{cases}$$

Hence,

$$n_i(\omega') = \begin{cases} n_i(\omega) & \text{if } i \leq r-1, \\ \max(t_{r+1}(\omega) - t_{\text{ext}} - m + 1, l_{r+1}(\omega) + 1, 0) & \text{if } i = r, \\ \max(t_x(\omega) - m + 1, l_r(\omega) + 1, 0) & \text{if } X \neq \emptyset \text{ and } i = x-1, \\ n_{i+1}(\omega) & \text{otherwise.} \end{cases}$$

We immediately have that $n_i(\omega') \leq n_{i+1}(\omega)$ for all $i \geq r$. In addition we know that $\varphi_{j_r(\omega),t} = 0$ for all $t \geq n_r(\omega)$. In particular $\varphi_{j_x(\omega),t} = 0$ for all $t \geq n_x(\omega)$ and $\varphi_{j_{x-1}(\omega'),t} = 0$ for all $t \geq n_{x-1}(\omega')$.

Therefore we have

$$\begin{aligned}
P(\omega', \boldsymbol{\phi}) &= \sum_{i=1}^{k-1} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{r-1} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} + \sum_{t=n_r(\omega')}^{t_r(\omega')+\nu_r(\omega')} \varphi_{j_r(\omega'),t} + \sum_{i=r+1}^{k-1} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_r(\omega')}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+1}(\omega),t} + \sum_{i=r+1}^{k-1} \sum_{t=n_i(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&\quad - \mathbb{I}_{\{X \neq 0\}} \left(\sum_{t=n_x(\omega)}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} - \sum_{n_{x-1}(\omega')}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} \right) \\
&= \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_r(\omega)}^{t_r(\omega)+\nu_r(\omega)} \varphi_{j_r(\omega),t} \\
&\quad + \sum_{t=n_r(\omega')}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+1}(\omega),t} + \sum_{i=r+1}^{k-1} \sum_{t=n_i(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&\geq \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_r(\omega)}^{t_r(\omega)+\nu_r(\omega)} \varphi_{j_r(\omega),t} \\
&\quad + \sum_{t=n_{r+1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+1}(\omega),t} + \sum_{i=r+1}^{k-1} \sum_{t=n_i(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&= \sum_{i=1}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} = P(\omega, \boldsymbol{\phi}).
\end{aligned}$$

Now we have seen that $P(\omega', \boldsymbol{\phi}) > P(\omega, \boldsymbol{\phi})$ for all $\omega \in \Omega''$ for some constructed move-wait ω'. Repeating this construction while there exists some $r, x \in \{1, ..., k\}$ such that $r \geq x + 1$ and $j_r \notin N_A(\omega, x)$ means there is a resultant walk $\omega''' \in \Omega'''$ such that $P(\omega''', \boldsymbol{\phi}) > P(\omega, \boldsymbol{\phi})$ for all $\omega \in \Omega''$. Hence

$$\max_{\omega \in \Omega''} P(\omega, \boldsymbol{\phi}) = \max_{\omega \in \Omega'''} P(\omega, \boldsymbol{\phi}),$$

along with lemma 3.2.12 the proof is concluded.

□

In order to aid in the computation of the performance of an arbitrary attacker strategy $\boldsymbol{\phi}$ we can restrict the time-horizon of the game such that the game ends when the maximal commencement time attack with a non-zero probability ends. We define the *restricted game length* for $\boldsymbol{\phi} \in \Phi$ as

$$T'(\boldsymbol{\phi}) = \max_{j \in N_A} \mathcal{T}(j, \boldsymbol{\phi}) + m,$$

the minimal time at which in the future there is a zero probability of catching a pure attacker henceforth, under $\boldsymbol{\phi}$.

Lemma 3.2.14. *For any $\boldsymbol{\phi} \in \Phi$ we have for all graphs Q, for all $m \geq 1$ and for all $T \geq T'(\boldsymbol{\phi})$ we have*

$$V_{\bullet, \boldsymbol{\phi}}(Q, T, m) = V_{\bullet, \boldsymbol{\phi}}(Q, T'(\boldsymbol{\phi}), m).$$

Proof. First we note that by the definition of $T'(\boldsymbol{\phi})$ that $\boldsymbol{\phi}$ is feasible in the game $G(Q, T'(\boldsymbol{\phi}), m)$ (i.e $\boldsymbol{\phi} \in \Phi(Q, T'(\boldsymbol{\phi}), m)$. Then as $\varphi_{j,t} = 0$ for all $T \geq T'(\boldsymbol{\phi})$ we have,

$$\begin{aligned}
\max_{W \in \mathcal{W}(Q,T,m)} P(W, \boldsymbol{\phi}) &= \max_{W \in \mathcal{W}} \sum_{t=0}^{T-1} S_{W(t),t}(\boldsymbol{l}(W, t)) \\
&= \max_{W \in \mathcal{W}(Q,T,m)} \left(\sum_{t=0}^{T'(\boldsymbol{\phi})-1} S_{W(t),t}(\boldsymbol{l}(W, t)) + \sum_{t=T'(\boldsymbol{\phi})}^{T-1} S_{W(t),t}(\boldsymbol{l}(W, t)) \right) \\
&= \max_{W \in \mathcal{W}(Q,T,m)} \sum_{t=0}^{T'(\boldsymbol{\phi})-1} S_{W(t),t}(\boldsymbol{l}(W, t)) \\
&= \max_{W \in \mathcal{W}(Q,T'(\boldsymbol{\phi}),m)} \sum_{t=0}^{T'(\boldsymbol{\phi})-1} S_{W(t),t}(\boldsymbol{l}(W, t)) = \max_{W \in \mathcal{W}(Q,T'(\boldsymbol{\phi}),m)} P(W, \boldsymbol{\phi}).
\end{aligned}$$

Hence we conclude the proof. □

Theorem 3.2.13 and lemma 3.2.14 make calculating the performance of an arbitrary attacker strategy $\boldsymbol{\phi}$ much easier by having to search through all $\omega \in \Omega'''$,

Input: Graph Q, attack length m, attacker strategy $\boldsymbol{\phi}$.
Result: Set Ω'''
Working set $\Omega_w = \emptyset$ and final set $\Omega_f = \emptyset$;
for *each* j *in* N_A **do**
  Set $j_1 = j$;
  if $\varphi_{j,t} = 0$ *for all* $t \geq 1$ **then**
    Set $V = \{0\}$;
  end
  else
    Set $V = \{\min \mathcal{T}(j, \boldsymbol{\phi}), ..., T - 1\}$;
  end
  for *each* v *in* V **do**
    Set $\nu_1 = v$;
    Form $\omega = ((j_1, \nu_1))$;
    if $\nu_1(\omega) = T'(\boldsymbol{\phi}) - 1$ **then**
      Add ω to Ω_f ;
    end
    else
      Add ω to Ω_w ;
    end
  end
end
while $\Omega_w \neq \emptyset$ **do**
  for *each* $\omega \in \Omega_w$ **do**
    if $N_A(\omega, k(\omega)) = \emptyset$ **then**
      remove ω from Ω_w ;
    end
    else
      for *each* $j \in N_A(\omega, k(\omega))$ **do**
        Set $j_{k(\omega)+1} = j$;
        if $\varphi_{j,t} = 0$ *for all* $t_{k(\omega)+1} + 1 \leq t \leq T'(\boldsymbol{\phi}) - 1$ **then**
          Set $V = \{0\}$;
        end
        else
          Set $V = \{0\} \cup \{1, 2, ..., T - t_{k(\omega)+1} - 1\}$;
        end
        for *each* v *in* V **do**
          Set $\nu_{k(\omega)+1} = v$;
          Form $\omega' = (\omega, (j_{k(\omega)+1}, \nu_{k(\omega)+1}))$;
          if $t_{k(\omega)+1}(\omega) + \nu_{k(\omega)+1}(\omega) = T'(\boldsymbol{\phi}) - 1$ **then**
            Add ω' to Ω_f ;
          end
          else
            Add ω' to Ω_w ;
          end
        end
      end
    end
  end
end
return $\Omega''' = \Omega_f$

Algorithm 1: Algorithm to find the set Ω''' for an attacker strategy $\boldsymbol{\phi}$.

rather than all $W \in \mathcal{W}$, for a game with a reduced game length $T'(\boldsymbol{\phi})$. Algorithm 1 constructs the set Ω''' for which we find $P(\omega, \boldsymbol{\phi})$ for all $\omega \in \Omega'''$ and take the maximum, thus finding the performance $V_{\bullet,\boldsymbol{\phi}}(Q, T, m)$.

To conclude this section, we remark that we know that an optimal response walk to any attacker strategy $\boldsymbol{\phi}$ admits a move, wait form. Further, we saw that the chosen nodes and waiting times can be restricted given the attacker strategy. This reduction from searching through $\mathcal{W}$ to searching through Ω''' as in theorem 3.2.13, makes the performance of any attacker strategy easier to find. Thus making it easier to find upper bounds on the value of a patrolling game. We will see the results from this section applied in section 3.3.3 to find the performance of a diametric attacker strategy $\boldsymbol{\phi}_{\text{di}}$. In addition the results are used throughout chapter 4 to find the performances of a variety of attacker strategies.

3.2.3 Response to attacker strategies with special properties

In this section we consider attacker strategies with certain properties that allow for further reduction of the best response patroller strategy. That is we further reduce the already reduced move-wait walk set Ω''', as in theorem 3.2.13, when $\boldsymbol{\phi} \in \Phi$ has certain properties. We first define some classes of attacker strategies which the majority of known attacker strategies belong to, before finding the reduction of Ω'''.

Definition 3.2.15. An attacker strategy $\boldsymbol{\phi} \in \Phi$ is called:

- *non-decreasing on* $N' \subset N_A$ if $\varphi_{j,t+1} \geq \varphi_{j,t}$ for all $j \in N'$ and for all $0 \leq t \leq T - m - 1$.
- *non-increasing on* $N' \subset N_A$ if $\varphi_{j,t+1} \leq \varphi_{j,t}$ for all $j \in N'$ and for all $0 \leq t \leq T - m - 1$.
- *node-identical on* $N' \subset N_A$ if $\varphi_{i,t} = \varphi_{j,t}$ for all $i, j \in N'$ and for all $0 \leq t \leq T - m - 1$.
- *semi-isolated on* $N' \subset N_A$ if $\lceil \frac{m}{2} \rceil \leq d(i,j) \leq m$ for all $i, j \in N'$.

Lemma 3.2.16. *For the game $G(Q, T, m)$ for any Q, for all $m \geq 1$, for all $T \geq m$ and for any $\boldsymbol{\phi}$ which is node-identical on N_A we have*

$$V_{\bullet,\boldsymbol{\phi}}(Q, T, m) = \max_{\omega \in \Omega_{Nid}} P(\omega, \boldsymbol{\phi}),$$

where $\Omega_{Nid} = \Omega''' \setminus (\Omega_{con} \cup \Omega_{non})$ in which

$$\Omega_{con} = \{\omega \in \Omega''' \mid \exists r \in \{1, ..., k-2\} \text{ such that } j_r = j_{r+2}, d(j_r, j_{r+1}, N_A) \geq m\}$$

and

$$\Omega_{non} = \{\omega \in \Omega''' \mid \exists r \in \{1, ..., k-2\} \text{ such that } j_r \neq j_{r+2}, d(j_r, j_{r+2}, N_A) \leq d(j_r, j_{r+1}, N_A), \\ d(j_r, j_{r+1}, N_A) \geq m, d(j_{r+1}, j_{r+2}, N_A) \geq m\}$$

The proof of lemma 3.2.16 follows by considering any walk $\omega \in \Omega_{\text{con}}$ and showing that it is possible to find an at a walk $\omega' \in \Omega_{\text{Nid}}$ such that $P(\omega', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$. This is then repeated for all $\omega \in \Omega_{\text{non}}$.

Proof. First we consider any walk $\omega \in \Omega_{\text{con}}$, then there exists an $r \in \{1, ..., k-2\}$ such that $j_r(\omega) = j_{r+2}(\omega)$ and $d(j_r(\omega), j_{r+2}(\omega), N_A) \geq m$. Consider the walk ω' in which

$$j_i(\omega') = \begin{cases} j_i(\omega) & \text{if } i \leq r, \\ j_{i+2}(\omega) & \text{if } r+1 \leq i \leq k-2 \end{cases}$$

and

$$\nu_i(\omega') = \begin{cases} \nu_i(\omega) & \text{if } i \leq r-1, \\ \nu_r(\omega) + t_{\text{ext}} & \text{if } i = r, \\ \nu_{i+2}(\omega) & \text{if } r+1 \leq i \leq k-2, \end{cases}$$

where $t_{\text{ext}} = \nu_{r+1}(\omega) + \nu_{r+1}(\omega) + d(j_r(\omega), j_{r+1}(\omega), N_A) + d(j_{r+1}(\omega), j_r(\omega), N_A)$. Then

$$t_i(\omega') = \begin{cases} t_i(\omega) & \text{if } i \leq r, \\ t_{i+2}(\omega) & \text{if } r+1 \leq i \leq k-2, \end{cases}$$

and with $X = \{i \in \{r+2, ..., k\} \mid j_i(\omega) = j_{r+1}(\omega)\}$ and if $X \neq \emptyset$ let $x = \min X$, we have

$$l_i(\omega') = \begin{cases} l_i(\omega) & \text{if } i \leq r, \\ l_{r+1}(\omega) & \text{if } X \neq \emptyset \text{ and } i = x-2, \\ l_{i+2}(\omega) & \text{otherwise.} \end{cases}$$

Hence

$$n_i(\omega') = \begin{cases} n_i(\omega) & \text{if } i \leq r, \\ \max(t_x(\omega) - m + 1, l_{r+1}(\omega) + 1, 0) & \text{if } X \neq \emptyset \text{ and } i = x-2, \\ n_{i+2}(\omega) & \text{otherwise.} \end{cases}$$

Immediately note that $n_{x-2}(\omega') \leq n_x(\omega)$. We have that $n_{r+1}(\omega) \geq t_r(\omega) + \nu_r(\omega)$ and $n_{r+2}(\omega) \geq t_{r+1}(\omega) + \nu_{r+1}(\omega)$. In addition as $\boldsymbol{\phi}$ is node identical we have

$\varphi_{j_r,t} = \varphi_{j_{r+1},t} = \varphi_{j_{r+2},t}$ for all $t \in \mathcal{T}$. Therefore we have

$$
\begin{aligned}
P(\omega', \boldsymbol{\phi}) &= \sum_{i=1}^{k-2} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_r(\omega)}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_r(\omega),t} + \sum_{i=r+1}^{k-2} \sum_{t=n_{i+2}(\omega)}^{t_{i+2}(\omega)+\nu_{i+2}(\omega)} \varphi_{j_{i+2}(\omega),t} \\
&\quad - \mathbb{I}_{\{X \neq \emptyset\}} \left(\sum_{t=n_x(\omega)}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} - \sum_{t=n_{x-2}(\omega')}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} \right) \\
&\geq \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_r(\omega)}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_r(\omega),t} + \sum_{i=r+1}^{k-2} \sum_{t=n_{i+2}(\omega)}^{t_{i+2}(\omega)+\nu_{i+2}(\omega)} \varphi_{j_{i+2}(\omega),t} \\
&\geq \sum_{i=1}^{r-1} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} + \sum_{t=n_r(\omega)}^{t_r(\omega)+\nu_r(\omega)} \varphi_{j_r(\omega),t} + \sum_{t=n_{r+1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+1}(\omega),t} \\
&\quad + \sum_{t=n_{r+2}(\omega)}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_{r+2}(\omega),t} + \sum_{i=r+1}^{k-2} \sum_{t=n_{i+2}(\omega)}^{t_{i+2}(\omega)+\nu_{i+2}(\omega)} \varphi_{j_{i+2}(\omega),t} \\
&= \sum_{i=1}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} = P(\omega, \boldsymbol{\phi}).
\end{aligned}
$$

Thus by repeating this process to any walk $\omega \in \Omega_{\text{con}}$ constructing $\omega'' \in \Omega''' \setminus \Omega_{\text{con}}$ such that $P(\omega'', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$. Hence

$$\max_{\omega \in \Omega'''} P(\omega, \boldsymbol{\phi}) = \max_{\omega \in \Omega''' \setminus \Omega_{\text{con}}} P(\omega, \boldsymbol{\phi}). \tag{3.12}$$

Secondly we consider a walk $\omega \in \Omega_{\text{non}}$, then there exists an $r \in \{1, ..., k-2\}$ such that $j_r(\omega) \neq j_{r+2}(\omega)$, $d(j_r, j_{r+2}, N_A) \leq d(j_r, j_{r+1}, N_A)$, $d(j_r, j_{r+1}, N_A) \geq m$, $d(j_{r+1}, j_{r+2}, N_A) \geq m$. Consider the walk ω' in which

$$j_i(\omega') = \begin{cases} j_i(\omega) & \text{if } i \leq r, \\ j_{i+1}(\omega) & \text{if } r+1 \leq i \leq k-1 \end{cases}$$

and

$$\nu_i(\omega') = \begin{cases} \nu_i(\omega) & \text{if } i \leq r, \\ \nu_{r+1}(\omega) + t_{\text{ext}} & \text{if } i = r+1 \\ \nu_{i+1}(\omega) & \text{otherwise,} \end{cases}$$

where $t_{\text{ext}} = \nu_{r+2}(\omega) + d(j_r(\omega), j_{r+1}(\omega), N_A) + d(j_{r+1}(\omega), j_{r+2}(\omega), N_A) - d(j_r(\omega), j_{r+2}(\omega), N_A)$. Then

$$t_i(\omega') = \begin{cases} t_i(\omega) & \text{if } i \leq r, \\ t_r(\omega) + \nu_r(\omega) + d(j_r(\omega), j_{r+2}(\omega), N_A) & \text{if } i = r+1, \\ t_{i+1}(\omega) & \text{if } r+2 \leq i \leq k-1, \end{cases}$$

and with $X = \{i \in \{r+3, ..., k\} \mid j_i(\omega) = j_{r+1}(\omega)\}$ and if $X \neq \emptyset$ let $x = \min X$, we have

$$l_i(\omega') = \begin{cases} l_i(\omega) & \text{if } i \leq r, \\ l_{r+1}(\omega) & \text{if } X \neq \emptyset \text{ and } i = x-1, \\ l_{i+1}(\omega) & \text{otherwise.} \end{cases}$$

Hence

$$n_i(\omega') = \begin{cases} n_i(\omega) & \text{if } i \leq r, \\ n_{r+1}(\omega') & \text{if } i = r+1, \\ \max(t_x(\omega) - m + 1, l_{r+1}(\omega) + 1, 0) & \text{if } X \neq \emptyset \text{ and } i = x-1, \\ n_{i+2}(\omega) & \text{otherwise,} \end{cases}$$

where $n_{r+1}(\omega') = \max(t_r(\omega)+\nu_r(\omega)+d(j_r(\omega), j_{r+2}(\omega), N_A)-m+1, l_{r+2}(\omega)+1, 0)$. Immediately we have $n_{x-1}(\omega') \leq n_x(\omega)$ and $n_{r+1}(\omega') \leq n_{r+1}(\omega)$. In addition $n_{r+2}(\omega) \geq t_{r+1}(\omega)+\nu_{r+1}(\omega)$. In addition as $\boldsymbol{\phi}$ is node identical we have $\varphi_{j_{r+2}(\omega),t} = \varphi_{j_{r+1}(\omega),t}$ for all $t \in \mathcal{T}$.

Therefore we have

$$
\begin{aligned}
P(\omega', \boldsymbol{\phi}) &= \sum_{i=1}^{k-1} \sum_{t=n_i(\omega')}^{t_i(\omega')+\nu_i(\omega')} \varphi_{j_i(\omega'),t} \\
&= \sum_{i=1}^{r} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega')} \varphi_{j_i(\omega),t} + \sum_{t=n_{r+1}(\omega')}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_{r+2}(\omega),t} + \sum_{i=r+2}^{k-1} \sum_{t=n_{i+1}(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&\quad - \mathbb{I}_{\{X\neq\emptyset\}} \left(\sum_{t=n_x(\omega)}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} - \sum_{t=n_{x-1}(\omega')}^{t_x(\omega)+\nu_x(\omega)} \varphi_{j_x(\omega),t} \right) \\
&\geq \sum_{i=1}^{r} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega')} \varphi_{j_i(\omega),t} + \sum_{t=n_{r+1}(\omega')}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_{r+2}(\omega),t} + \sum_{i=r+2}^{k-1} \sum_{t=n_{i+1}(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&\geq \sum_{i=1}^{r} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega')} \varphi_{j_i(\omega),t} + \sum_{t=n_{r+1}(\omega)}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_{r+2}(\omega),t} + \sum_{i=r+2}^{k-1} \sum_{t=n_{i+1}(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&\geq \sum_{i=1}^{r} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega')} \varphi_{j_i(\omega),t} + \sum_{t=n_{r+1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+2}(\omega),t} + \sum_{t=n_{r+2}(\omega)}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_{r+2}(\omega),t} \\
&\quad + \sum_{i=r+2}^{k-1} \sum_{t=n_{i+1}(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&= \sum_{i=1}^{r} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega')} \varphi_{j_i(\omega),t} + \sum_{t=n_{r+1}(\omega)}^{t_{r+1}(\omega)+\nu_{r+1}(\omega)} \varphi_{j_{r+1}(\omega),t} + \sum_{t=n_{r+2}(\omega)}^{t_{r+2}(\omega)+\nu_{r+2}(\omega)} \varphi_{j_{r+2}(\omega),t} \\
&\quad + \sum_{i=r+2}^{k-1} \sum_{t=n_{i+1}(\omega)}^{t_{i+1}(\omega)+\nu_{i+1}(\omega)} \varphi_{j_{i+1}(\omega),t} \\
&= \sum_{i=1}^{k} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i(\omega)} \varphi_{j_i(\omega),t} = P(\omega, \boldsymbol{\phi}).
\end{aligned}
$$

Thus by repeating this process to any walk $\omega \in \Omega_{\text{non}}$ constructing $\omega'' \in \Omega''' \setminus \Omega_{\text{non}}$ such that $P(\omega'', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$. Hence

$$
\max_{\omega\in\Omega'''} P(\omega, \boldsymbol{\phi}) = \max_{\omega\in\Omega'''\setminus\Omega_{\text{non}}} P(\omega, \boldsymbol{\phi}). \tag{3.13}
$$

Thus the result of the lemma is obtained by equations (3.12) and (3.13) along with theorem 3.2.13. □

Further to the reduction in lemma 3.2.16, in which $\boldsymbol{\phi}$ is node-identical on N_A, when $\boldsymbol{\phi}$ is semi-isolated and non-increasing or non-decreasing on N_A we can get further reductions by considering when such walks such wait.

Corollary 3.2.17. *For a game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$, for all $T \geq m$ and for any $\boldsymbol{\phi} \in \Phi$ which is node-identical on N_A, semi-isolated on N_A and*

- *non-decreasing on* N_A *we have*
$$V_{\bullet,\boldsymbol{\phi}}(Q,T,m)=\max_{\omega\in\Omega_\uparrow}P(\omega,\boldsymbol{\phi}),$$
where $\Omega_\uparrow=\{\omega\in\Omega_{Nid}\mid \nu_r=0$ *for all* $r\in\{2,...,k-1\}\}$.

- *non-increasing on* N_A *we have*
$$V_{\bullet,\boldsymbol{\phi}}(Q,T,m)=\max_{\omega\in\Omega_\downarrow}P(\omega,\boldsymbol{\phi}),$$
where $\Omega_\downarrow=\{\omega\in\Omega_{Nid}\mid \nu_r=0$ *for all* $r\in\{i\in\{3,...,k-1\}\mid \exists i'\in\{1,...,i\}$ *such that* $j_i=j_{i'}\}\}$.

The proof of corollary 3.2.17 follows from theorem 3.2.10 by considering that the waiting times can be transferred backwards in the case of non-decreasing and forward in the case of non-increasing.

Proof. When $\boldsymbol{\phi}$ is semi-isolated on N_A we have that the event $E_3(\omega,r)$ is true for all $r\in\{2,...,k-1\}$ and for all $\omega\in\Omega_{\text{Nid}}\setminus\Omega_\uparrow$ as

$$\begin{aligned}&\max(t_{\min\{i\in\{r+1,...,k\}\mid j_i=j_r\}}(\omega)-m+1,0)\geq t_{r+1}(\omega)-m+1\\&\quad=t_{r-1}(\omega)+\nu_{r-1}(\omega)+d(j_{r-1},j_r,N_A)+\nu_r(\omega)+d(j_r,j_{r+1},N_A)-m+1\\&\quad\geq t_{r-1}(\omega)+v_{r-1}(\omega)+\nu_r(\omega)+1>t_{r-1}(\omega)+\nu_{r-1}(\omega)+1.\end{aligned}$$

In addition $t_r(\omega)-m+1=t_{r-1}(\omega)+\nu_{r-1}(\omega)+d(j_{r-1},j_r,N_A)-m+1\leq t_{r-1}(\omega)+\nu_{r-1}(\omega)+1$ and so when $\boldsymbol{\phi}$ is also node-identical on N_A and non-decreasing on N_A we have whilst $t_{r-1}(\omega)+\nu_{r-1}(\omega)+1\leq T-m$ that

$$\varphi_{j_{r-1},t_{r-1}(\omega)+\nu_{r-1}(\omega)+1}=\varphi_{j_r,t_{r-1}(\omega)+\nu_{r-1}(\omega)+1}\geq\varphi_{j_r,t_r(\omega)-m+1}.$$

Hence $G^b(\omega,r)\geq L^b(\omega,r)$ for all $r\in\{2,...,k-1\}$ such that $t_{r-1}(\omega)+\nu_{r-1}(\omega)+1\leq T-m$ for all $\omega\in\Omega_{\text{Nid}}\setminus\Omega_\uparrow$ and so by theorem 3.2.10 we have $P(\omega^b,\boldsymbol{\phi})\geq P(\omega,\boldsymbol{\phi})$. Therefore we can transfer all waiting times backwards for any $r\in\{2,...,k-1\}$ such that $t_{r-1}(\omega)+\nu_{r-1}(\omega)+1\leq T-m$ without decreasing the payoff. For $r\in\{2,...,k-1\}$ such that $t_{r-1}(\omega)+\nu_{r-1}(\omega)+1>T-m$ then $t_r+1>T-m$ and so $\varphi_{j_r,t}=0$ for all $t\geq t_r+1$ and so lemma 3.2.11 enforces that $v_r=0$ for all such r (by forward transfer of the waiting). So we end up with some $\omega'\in\Omega_\uparrow$ such that $P(\omega',\boldsymbol{\phi})\geq P(\omega,\boldsymbol{\phi})$ for all $\omega\in\Omega_{\text{Nid}}\setminus\Omega_\uparrow$. Hence

$$\max_{\omega\in\Omega_{\text{Nid}}}P(\omega,\boldsymbol{\phi})=\max_{\omega\in\Omega_\uparrow}P(\omega,\boldsymbol{\phi}),$$

along with lemma 3.2.16 we attain the result.

For the second statement a similar approach can be taken. When $\boldsymbol{\phi}$ is semi-isolated on N_A we have that the event $E_1(\omega,r)$ is true for all $r\in\{i\in\{3,...,k-1\}\mid \exists i'\in\{1,...,i\}$ such that $j_i=j_{i'}\}$ for all $\omega\in\Omega_{\text{Nid}}\setminus\Omega_\downarrow$ as

$$\begin{aligned}t_{r+1}(\omega)-m+1&=t_{r-1}(\omega)+\nu_{r-1}(\omega)+d(j_{r-1},j_r,N_A)+\nu_r(\omega)+d(j_r,j_{r+1},N_A)-m+1\\&\geq t_{r-1}(\omega)+\nu_{r-1}(\omega)+\nu_r(\omega)+1\geq l_{r+1}(\omega)+1.\end{aligned}$$

In addition $t_{r+1}(\omega) - m = t_r(\omega) + \nu_r(\omega) + d(j_r, j_{r+1}, N_A) - m \leq t_r(\omega) + \nu_r(\omega)$ and so when $\boldsymbol{\phi}$ is also node-identical on N_A and non-increasing on N_A we have

$$\varphi_{j_{r+1}, t_{r+1}(\omega)-m} = \varphi_{j_r, t_{r+1}(\omega)-m} \geq \varphi_{j_r, t_r(\omega)+\nu_r(\omega)}$$

Hence $G^f(\omega, r) \geq L^f(\omega, r)$ for all $r \in \{i \in \{3, ..., k\} \mid \exists i' \in \{1, ..., i\}$ such that $j_i = j_{i'}\}$ for all $\omega \in \Omega_{\text{Nid}} \setminus \Omega_{\downarrow}$ and so by theorem 3.2.10 we have $P(\omega^f, \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$. Therefore we can transfer any waiting forward and end up with some $\omega' \in \Omega_{\downarrow}$ such that $P(\omega', \boldsymbol{\phi}) \geq P(\omega, \boldsymbol{\phi})$ for all $\omega \in \Omega_{\text{Nid}} \setminus \Omega_{\downarrow}$. Hence

$$\max_{\omega \in \Omega_{\text{Nid}}} P(\omega, \boldsymbol{\phi}) = \max_{\omega \in \Omega_{\downarrow}} P(\omega, \boldsymbol{\phi}),$$

along with lemma 3.2.16 we attain the result.

□

Notice that in corollary 3.2.17 if $\boldsymbol{\phi}$ is non-decreasing, semi-isolated and node identical on N_A then only $\nu_1, \nu_k \neq 0$, which is equivalent to considering only a choice of the initial waiting ν_1 as $t_k + \nu_k = T - 1$ by definition. Essentially for such $\boldsymbol{\phi}$ there is one degree of freedom in the waiting of walks to consider, while there may be an unknown amount of node sequences in the move-wait form walk to consider. We will see corollary 3.2.17 used in section 3.3.3 to find the correct performance of $\boldsymbol{\phi}_{\text{di}}$ for all game lengths $T \geq m$. This work concludes our contributions to the reduction of the response space for the patroller when evaluating the performance of an attacker strategy.

3.3 General strategies

3.3.1 Decomposition into subgraph games

While lemma 2.3.14 provides a lower bound on the value $V(Q, T, m)$ this bound requires knowledge of the values of all subgraph games, this requirement can be relaxed by considering arbitrary patroller strategies on subgraph games.

Definition 3.3.1. For the game $G(Q, T, m)$ with a decomposition of Q into $Q_1, ..., Q_R$, we form the subgraph games $G(Q_1, T, m), ..., G(Q_R, T, m)$. Let $\boldsymbol{\pi}_i \in \Pi(Q_i, T, m)$ be an arbitrary patroller strategy chosen for the game $G(Q_i, T, m)$ then the *arbitrary decompsoition* patroller strategy $\boldsymbol{\pi}_{\text{ADec}}$ using these strategies is such that $\pi_{\beta_1(W)} = \sum_{i=1}^{R} p_i \pi_{i, \beta_1(W)}$, where

$$p_i = \frac{1}{V_{\boldsymbol{\pi}_i, \bullet}(Q_i, T, m) \sum_{r=1}^{R} \frac{1}{V_{\boldsymbol{\pi}_r, \bullet}(Q_r, T, m)}}$$

for $i = 1, ..., R$.

An arbitrary decomposition is a more general decomposition idea than that previously presented and allows us to use strategies which we do not know are optimal in the subgraph games. While it is intuitive to use optimal strategies such strategies may not be known and therefore arbitrary strategies can be used. By evaluating the performance of any arbitrary decomposition patroller strategy we can achieve a lower bound on the value of the game similar to that in lemma 2.3.14.

Lemma 3.3.2. *For the game $G(Q,T,m)$ for any graph Q, with any decomposition Q_i for $i = 1,...,R$ ($Q = \bigcup_{i=1}^{R} Q_i$), for any strategies $\boldsymbol{\pi}_i \in \Pi(Q_i,T,m)$ such that $V_{\boldsymbol{\pi}_i,\bullet}(Q_i,T,m) \neq 0$ for $i = 1,...,R$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q,T,m) \geq V_{\boldsymbol{\pi}_{ADec},\bullet}(Q,T,m) \geq \frac{1}{\sum_{i=1}^{R} \frac{1}{V_{\boldsymbol{\pi}_i,\bullet}(Q_i,T,m)}},$$

where the lower bound on $V(Q,T,m)$ is achieved by the patroller using a arbitrary decomposition patroller strategy $\boldsymbol{\pi}_{ADec}$ using $\boldsymbol{\pi}_i$ for $i = 1,...,R$.

The proof of lemma 3.3.2 follows similarly to the proof of lemma 2.3.14 by bounding the performance of an arbitrary decomposition $\boldsymbol{\pi}_{\text{ADec}}$ using $\boldsymbol{\pi}_i$ for $i = 1,...,R$.

Proof. As any node $j \in N$ is in at least one subgraph Q_i for some $i \in \{1,...,R\}$ we have that with probability p_i the patroller will play $\boldsymbol{\pi}_i$ and hence

$$V_{\boldsymbol{\pi}_{\text{ADec}},\bullet}(Q,T,m) \geq p_i V_{\boldsymbol{\pi}_i,\bullet}(Q_i,T,m) = \frac{1}{\sum_{r=1}^{R} \frac{1}{V_{\boldsymbol{\pi}_r,\bullet}(Q_r,T,m)}}.$$

Therefore we achieve the lower bound on $V(Q,T,m)$. □

We note the requirement that the performance of each arbitrary strategy be non-zero. This is not an issue in lemma 2.3.14 as an optimal strategy must have a non-zero performance (as a lower bound above zero is guaranteed by the choose and wait patroller strategy). In addition to the lower bound in lemma 3.3.2 we briefly discuss when the use of decomposition provides a weak bound. If the subgraphs are not disjoint, then it is possible to improve the decomposition by removing the overlapping nodes from all but one subgraph. This can only improve the performance of strategies which can be picked in the new subgraph game as such nodes can be skipped and hence we should only consider disjoint decompositions. Furthermore decompositions which have some subgraph Q_i where $V_{\boldsymbol{\pi}_i,\bullet}(Q_i,T,m) = 1$, may be ineffective. In such a case if the pure patrol visits every node at intervals less than m time units apart, then while it captures all pure attacks, it has the potential to visit more nodes (outside the subgraph) and still do so. While this fact is hard to use explicitly, as moving nodes between subgraphs may cause the bound to decrease, it is an idea we should keep in mind when using decomposition to obtain a lower bound. We use our contribution to the decomposition in chapter 4, section 4.4, in order to obtain an optimal solution for a graph with a highly generalised star structure.

3.3.2 Simplification and expansion

In this section we formalise the process of repeated node identification, define its inverse and state explicitly the resultant embedded strategies which generate the bounds achieved by these processes. While the ideas of repeated node identication and its inverse have been used in [16] and [107] to get the value of certain patrolling games, the underlying strategies are not always clearly stated. Therefore, for clarity of strategy implementation, we focus on the generated strategies after discussing the formal operations.

Recall that node-identification is a graphical operation which performs a merging of two parent nodes, into a child node which retains its parents adjacencies. Consider a graph $Q = (N, E)$ which undergoes node identification to $Q^- = (N^-, E^-)$, then by lemma 2.3.2 we know $V(Q^-, T, m) \geq V(Q, T, m)$. An upper bound on $V(Q, T, m)$ can be generated by looking at an attacker strategy $\boldsymbol{\phi}^- \in \Phi(Q^-, T, m)$ to get $V(Q, T, m) \leq V_{\bullet, \boldsymbol{\phi}^-}(Q^-, T, m)$. For the attack strategy in the game $G(Q, T, m)$ which generates this same bound, we need to embed the strategy $\boldsymbol{\phi}^- \in \Phi(Q^-, T, m)$ to find a strategy $\boldsymbol{\phi} \in \Phi(Q, T, m)$. To embed the attacker strategy, we must place all pure attacks at the child node at the parent nodes. Therefore the embedded attack strategy has an arbitrary distribution which shares the attack probabilities at the child node between the parents. We formally define the embedded attack strategy as follow.

Definition 3.3.3. Given a node-identification from a graph Q to Q^-, merging nodes $u, v \in N$ to node $w \in N^-$, we define the *embedded attacker strategy* $\boldsymbol{\phi} \in \Phi(Q, T, m)$ embedding $\boldsymbol{\phi}^- \in \Phi(Q^-, T, m)$ such that the probability of choosing the pure attack strategy $(j, \tau) \in \mathcal{A}(Q, T, m)$ is

$$\varphi_{j,\tau} = \begin{cases} \varphi^-_{j,\tau} & \text{if } j \in N \setminus \{u, v\}, \\ X_\tau \varphi^-_{w,\tau} & \text{if } j = u, \\ (1 - X_\tau)\varphi^-_{w,\tau} & \text{if } j = v, \end{cases}$$

for some arbitrary chosen $X_\tau \in [0, 1]$ for each $\tau \in \mathcal{T}$.

That is an embedded attacker strategy $\boldsymbol{\phi} \in \Phi(Q, T, m)$ embedding $\boldsymbol{\phi}^- \in \Phi(Q^-, T, m)$ uses the same distribution among pure attacks and arbitrarily splits the distribution at the node $w \in N^-$ to nodes $u, v \in N$ for each commencement time, according to $X_\tau \in [0, 1]$. It is the embedded attack strategies which generated the upper bound on $V(Q, T, m)$ by node-identification (lemma 2.3.2), which need not use optimal attacker strategies in the game $G(Q^-, T, m)$ for the embedding.

Lemma 3.3.4. *Given a graph Q in which (u, v) are node-identified to form the graph Q^- and an attacker strategy $\boldsymbol{\phi}^- \in \Phi(Q^-, T, m)$ embedded to form $\boldsymbol{\phi} \in \Phi(Q, T, m)$, then for any $m \geq 1$ and for any $T \geq m$ we have*

$$V(Q, T, m) \leq V_{\bullet, \boldsymbol{\phi}}(Q, T, m) \leq V_{\bullet, \boldsymbol{\phi}^-}(Q^-, T, m).$$

$$V(Q, T, m) \leq V(Q^-, T, m).$$

Lemma 3.3.4 follows by the same argument originally given for node-identification bounds.

Proof. For any $W \in \mathcal{W}(Q, T, m)$ form a walk $W' \in \mathcal{W}(Q^-, T, m)$ such that $W'(t) = \mathcal{N}^-(Q, u, v)(W(t))$ for all $t \in \mathcal{T}$. Let w be the child of u and v, then if $w \in W'(I)$ then either $u \in W(I)$ or $v \in W(I)$ for any attack interval $I = \{\tau, ..., \tau + m - 1\}$ (for any $\tau \in \mathcal{T}$). So

$$\begin{aligned}
P(W, \boldsymbol{\phi}) &= \sum_{j \in N \setminus \{u,v\}} \sum_{\tau=0}^{T-1} \varphi_{j,\tau} \mathbb{I}_{\{j \in W(I)\}} + \sum_{\tau=0}^{T-1} \varphi_{u,\tau} \mathbb{I}_{\{u \in W(I)\}} + \sum_{\tau=0}^{T-1} \varphi_{v,\tau} \mathbb{I}_{\{v \in W(I)\}} \\
&= \sum_{j \in N \setminus \{u,v\}} \sum_{\tau=0}^{T-1} \varphi^-_{j,\tau} \mathbb{I}_{\{j \in W(I)\}} + \sum_{\tau=0}^{T-1} X_t \varphi^-_{w,\tau} \mathbb{I}_{\{u \in W(I)\}} + \sum_{\tau=0}^{T-1} (1 - X_t) \varphi^-_{w,\tau} \mathbb{I}_{\{v \in W(I)\}} \\
&\leq \sum_{j \in N \setminus \{u,v\}} \sum_{\tau=0}^{T-1} \varphi_{j,\tau} \mathbb{I}_{\{j \in W'(I)\}} + \sum_{\tau=0}^{T-1} \varphi^-_{w,\tau} \mathbb{I}_{\{w \in W'(I)\}} \\
&= P(W', \boldsymbol{\phi}^-).
\end{aligned}$$

Hence we have $V_{\bullet,\boldsymbol{\phi}}(Q, T, m) \leq V_{\bullet,\boldsymbol{\phi}^-}(Q^-, T, m)$ and therefore the result. □

Having seen how node identification and embedded attack strategies generate upper bounds, we look at the reverse of node identification, which we call *node-splitting*. Node splitting takes a parent node and splits it into two resultant child nodes, which are adjacent to each other and who between them inherit the parent's adjacencies.

Definition 3.3.5. The graphical operator of *node-splitting* maps a (simple undirected) graph $Q = (N, E)$ onto Q^+ by splitting the node w, into two nodes u, v, written as $\mathcal{Q}^+(Q, w) = Q^+$. The resultant graph $Q^+ = (N^+, E^+)$ is such that

$$N^+ = (N \setminus \{w\}) \cup \{u, v\} \text{ and } E^+ = (E \setminus E') \cup Y',$$

where $E' = \{(w, i) \mid i \in N, (w, i) \in E\}$ are the incident edges to w for some choice of *edge transference* set $Y' \subset Y$ where

$$Y = \{(u, x) \mid (w, x) \in E\} \cup \{(v, x) \mid (w, x) \in E\} \cup \{u, v\}$$

where Y' must be chosen such that $(u, v) \in Y'$ and for each $x \in N$ if $(, x) \in E$ then either $(u, x) \in Y'$ or $(v, x) \in Y'$. Further we denote the node-splitting node and edge maps as $\mathcal{N}^+(Q, w, j)$ and $\mathcal{E}^+(Q, w, e)$ respectively, which map nodes $j \in N$ and edge $e \in E$ of Q to nodes and edges of Q^+.

We note that the choice of the edge transference set $Y' \subset Y$ in our definition is arbitrary and some alternative definitions require $Y' = Y$, which we call *full edge transference*, such that children each inherit all parent adjacencies. While we could have adopted this requirement, as it means that the resultant graph Q^+ will have the most options for walks (among all choices of Y'), it is not required for our results and therefore we keep it as general as possible. In fact a smart choice of the edge transference set can keep the set of edges manageable when performing repeated node-splitting (later known as expansion). While node-identification and node-splitting perform the reverse ideas it is easy to see that they are not inverse operations.

Performing node-splitting on a parent node followed by node-identification on the resultant child nodes will result in the same graph, i.e $\mathcal{Q}^{-}(\mathcal{Q}^{+}(Q,w),u,v) = Q$, in which u, v are distinct elements of $\mathcal{N}^{+}(w), \mathcal{N}^{+}(w)$. Therefore, we say that node-identification is left inverse for node-splitting. However performing node-identification followed by node-splitting may not result in the same graph (unless Y' is suitably chosen or $Y' = Y$) and so it is not true that node splitting is the left inverse for node-identification. Figure 3.3.1 shows an example of this, where nodes lose previous adjacencies, as the edge $(4,3)$ no longer exists, and may gain ones which were not previously present, but were shared by their sibling (the other child form the parent), such as the edge $(2,6)$.

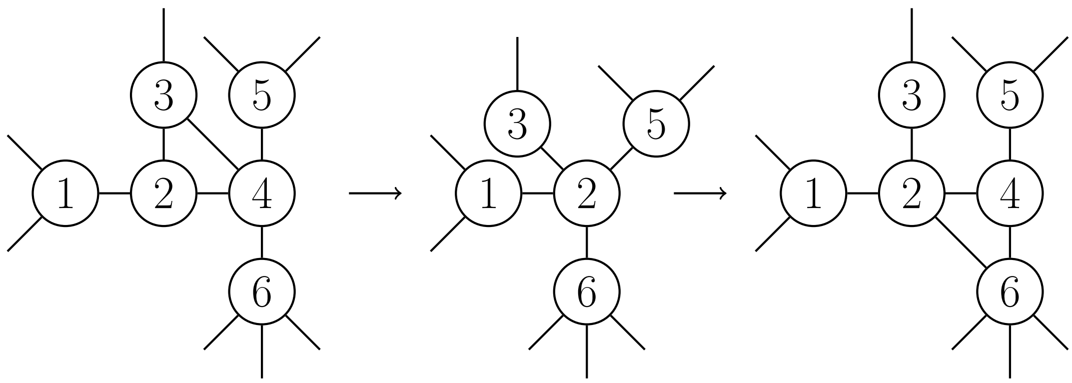


Figure 3.3.1: A graph, with nodes 2 and 4 undergoing node-identification followed by there resultant node 2 being node splitting back into nodes 2 and 4. Note that adjacencies are not necessarily recovered under node splitting.

As node-identification is the left inverse of node-splitting we know that $V(Q,T,m) \geq V(Q^{+},T,m)$, therefore it possible to consider the patroller strategy which generates this lower bound on $V(Q,T,m)$ by considering the embedding of any strategy $\boldsymbol{\pi}^{+} \in \Pi(Q^{+},T,m)$ to form a strategy $\boldsymbol{\pi} \in \Pi(Q,T,m)$.

Definition 3.3.6 (Embedded patroller strategy)**.** Given a node-splitting of the graph Q into $Q^{+} = \mathcal{Q}^{+}(Q,w)$, which splits node $w \in N$ into $u, v \in N^{+}$, for some choice of edge transference Y' we define the *embedded walk* $W \in \mathcal{W}(Q,T,m)$, embedding the walk $W^{+} \in \mathcal{W}(Q^{+},T,m)$, such that $W^{+}(t) = \mathcal{N}^{+}(Q,w)(W(t))$ for all $t \in \mathcal{J}$. An *embedded patroller strategy* strategy $\boldsymbol{\pi} \in \Pi(Q,T,m)$, embedding $\boldsymbol{\pi}^{+} \in \Pi(Q^{+},T,m)$, is such that

$$\pi_{\beta_1(W)} = \sum_{W^{+} \in \mathcal{W}(Q^{+},T,m)} \pi^{+}_{\beta_1^{+}(W^{+})} \mathbb{I}_{\{W^{+}(t) = \mathcal{N}^{+}(Q,w)(W(t)) \ \forall t \in \mathcal{J}\}},$$

where $\beta_1^{+} : \mathcal{W}(Q^{+},T,m) \to \{1, ..., |\mathcal{W}(Q^{+},T,m)|\}$ is the arbitrary bijection chosen for the ordering of the walks in the game $G(Q^{+},T,m)$.

That is an embedded walk $W \in \mathcal{W}(Q,T,m)$, embedding the walk $W^+ \in \mathcal{W}(Q^+,T,m)$, follows W^+ expect when it is at node u or v. When W^+ uses either the edge (u,x) or (v,x) for some $x \in N^+ \setminus \{u,v\}$, then W uses the edge (w,x) in Q. If the walk W^+ uses the edge (u,v) the walk W waits at w. In addition an embedded patroller strategy $\boldsymbol{\pi} \in \Pi(Q,T,m)$ embedding $\boldsymbol{\pi}^+ \in \Pi(Q^+,T,m)$ plays each walk $W \in \mathcal{W}(Q,T,m)$ with a probability equal to the sum of all walks $\boldsymbol{\pi}^+$ played which embed to W.

Lemma 3.3.7. *Given a graph Q which undergoes node-splitting at node w to form Q^+ and a patroller strategy $\boldsymbol{\pi}^+ \in \Pi(Q^+,T,m)$ embedded to form $\boldsymbol{\pi} \in \Pi(Q,T,m)$, then for any $m \geq 1$ and for any $T \geq m$ we have*

$$V(Q,T,m) \geq V_{\boldsymbol{\pi},\bullet}(Q,T,m) \geq V_{\boldsymbol{\pi}^+,\bullet}(Q^+,T,m).$$

Moreover,

$$V(Q,T,m) \geq V(Q^+,T,m).$$

The proof of lemma 3.3.7 follows similarly to the proof of lemma 3.3.4.

Proof. For any $(j,\tau) \in \mathcal{A}(Q,T,m)$ form an attack $(j',\tau) \in \mathcal{A}(Q^+,T,m)$ such that $j' = j$ if $j \neq w$ and $j' = u$ if $j = w$ where u is child of w. Then if $j' \in W^+(I)$ then $j \in W(I)$ where $W \in \mathcal{W}(Q,T,m)$ is the embedded walk of $W' \in \mathcal{W}(Q^+,T,m)$ and $I = \{\tau, ..., \tau+m-1\}$. So

$$\begin{aligned}
P(\boldsymbol{\pi},(j,\tau)) &= \sum_{W \in \mathcal{W}(Q,T,m)} \pi_{\beta_1(W)} \mathbb{I}_{\{j \in W(I)\}} \\
&= \sum_{W \in \mathcal{W}(Q,T,m)} \sum_{W^+ \in \mathcal{W}(Q^+,T,m)} \pi^+_{\beta_1^+(W^+)} \mathbb{I}_{\{W^+(t) = \mathcal{N}^+(Q,w)(W(t)) \text{ for all } t \in \mathcal{J}\}} \mathbb{I}_{\{j \in W(I)\}} \\
&\geq \sum_{W^+ \in \mathcal{W}(Q^+,T,m)} \pi^+_{\beta_1^+(W^+)} \mathbb{I}_{\{j \in W(I)\}} \\
&\geq \sum_{W^+ \in \mathcal{W}(Q^+,T,m)} \pi^+_{\beta_1^+(W^+)} \mathbb{I}_{\{j' \in W^+(I)\}} \\
&= P(\boldsymbol{\pi}^+,(j',\tau)).
\end{aligned}$$

Hence we have $V_{\boldsymbol{\pi},\bullet}(Q,T,m) \geq V_{\boldsymbol{\pi}^+,\bullet}(Q^+,T,m)$ and therefore the result. The moreover result holds by considering any optimal patroller strategy $\boldsymbol{\pi}^* \in \Pi(Q,T,m)$. □

We have seen the embedded strategies which generate bounds on the value of the game by merging or splitting nodes. As a single node-identification or node-splitting may not generate optimal bounds, we extend the operations to *simplification* and *expansion*, respectively, where a k-simplification operation performs k node-identifications and an l-expansion operation performs l node-splittings. Using the inequalities for node-identification and node-splitting (lemmas 3.3.4 and 3.3.7 respectively) we obtain the following bounds.

Corollary 3.3.8. *For any (simple undirected) graph Q which can be k-simplified to Q^{-k} and l-expanded to Q^{+l}, for all $k \geq 1$, for all $l \geq 1$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q^{+l}, T, m) \leq V(Q, T, m) \leq V(Q^{-k}, T, m).$$

Proof. From repeated application of the node-identification result in lemma 3.3.4 we get that

$$V(Q^{-k}, T, m) \geq V(Q^{-(k-1)}) \geq ... \geq V(Q, T, m).$$

Similarly, from repeated application of the node-splitting result in lemma 3.3.7 we get that

$$V(Q^{+l}, T, m) \leq V(Q^{+(l-1)}) \leq ... \leq V(Q, T, m).$$

□

As simplification and expansion are repeated node-identification and node-splitting respectively, it is possible to get embed strategies from the games $G(Q^{-k}, T, m)$ and $G(Q^{+l}, T, m)$ to get strategies for the game $G(Q, T, m)$, by repeated embedding of attacker and patroller strategies respectively. It is these repeatedly embedded strategies which generate the bounds in corollary 3.3.8, along with the optimal strategies for $G(Q^{+l}, T, m)$ and $G(Q^{+k}, T, m)$.

Definition 3.3.9. A k-embedded attack strategy is the repeated construction of embedded attack strategy $\boldsymbol{\phi}^{-(k-1)}$ from $\boldsymbol{\phi}^{-k}$ k times for some collection of choices for $X_{j,t}$ for $t \in \mathcal{T}$ for each embedding step $j = 1, ..., k$. Similarly an l-embedded patrol strategy is the repeated constructed of the embedded patrol strategy $\boldsymbol{\pi}^{+(l-1)}$ from $\boldsymbol{\pi}^{+l}$ l times.

Corollary 3.3.8 provides us with a clear reasoning that if a k-embedded attack strategy is optimal for the game $G(Q, T, m)$ then we know a j-embedded attack strategy that is optimal for the game $G(Q^{-j}, T, m)$ for $j = 0, 1, ..., k-1$. Note this is under the family of simplifications, i.e $Q^{-(j+1)}$ is simplified from Q^{-j}. Similarly knowing a l-embedded patrol strategy that is optimal in Q means we know a j-embedded patrol strategy that is optimal for the game $G(Q^{+j}, T, m)$ for $j = 0, 1, ..., l-1$. Essentially this is because all the intermediate simplifications or expansions bounds turn into equalities.

Thus, finding an optimal embedded attack strategy for a game on Q from some game on Q^{+k} for some k means all intermediate simplifications are solved by the intermediate embedded attack strategy and the embedded patrol strategy where Q^{+k} is expanded to Q. A similar statement holds when an optimal embedded patrol strategy is known in Q from some Q^{-l}. Thus using simplification and expansion to find optimal solutions, finds solutions for all patrolling games on graph intermediates. This helps to solve a set of graphical structures at the same time as solving a particular one. In particular one can see this idea in the solution to the complete bipartite graph. For the complete bipartite, $K_{a,b}$, with $a \leq b$, the value of the game does not depend on a, this is because node splitting is repeated on its nodes in order to get the optimal embedded strategy when $a = b$, by making the graph Hamiltonian.

So while finding an optimal embedded strategy provides us with solutions to all intermediate games, it can be incredibly hard to find such a simplification or expansion, each intermediate operation for simplification has $\binom{n}{2}$ choices and for expansion has n choices for the node to split along with $d(w)^2$ choices for the edge transfer when node w is chosen to be split, where $d(w)$ is the degree of node w.

As searching for the best k-simplification and l-expansion may be extremely time consuming, we may wish to reduce which nodes are considered. Consider a graph with leaf nodes and its adjacent nodes, called *penultimate* nodes, we show that it is not worth node identifying a leaf node with any other node than their adjacent penultimate nodes. Thus limiting the node-identification operation when in the search of optimality.

Lemma 3.3.10. *For the game $G(Q, T, m)$, for any graph Q with a leaf node i and its penultimate node p, for all $j \notin \{i, p\}$, for all $T \geq m$ and for all $m \geq 1$, we have*

$$V(Q^-(i,p), T, m) \leq V(Q^-(i,j), T, m),$$

where $Q^-(i,j) = \mathcal{Q}^-(Q, i, j)$.

Proof. For Q undergoing node-identification on i and j, forming $Q^-(i,j) = (N^-(i,j), E^-(i,j))$, which uses the relabelling of the child node w as j (isomorphic). Then we have $N^-(i,j) = N \setminus \{i\}$ and $E^-(i,j) = E \cup \{(p,j)\}$

Now consider Q undergoing node-identification on i and p, forming $Q^-(i,p) = (N^-(i,p), E^-(i,p))$, which uses the relabelling of the child node w as p (isomorphic). Then we have $N^-(i,p) = N \setminus \{i\}$ and $E^-(i,p) = E$.

Thus we can see that $Q^-(i,j)$ is the same graph as $Q^-(i,p)$ except it contains the edge (j,p). Hence as we know more edges can only increase the value (lemma 1 a) in [16]), we obtain that for all $m \geq 1$ and for all $T \geq m$ that

$$V(Q^-(i,p), T, m) \leq V(Q^-(i,j), T, m).$$

$\square$

In section 3.3.4 we look at using a Hamiltonian expansion for non-Hamiltonian graphs, along with the embedded Hamiltonian cycle, called a *full-node cycle*. In doing so we present a 'Hamiltonian like' bound.

3.3.3 Generalising the diametric attack

In this section we first correct lemma 2.3.24, stating the performance of a diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{di}}$ for any $T \geq m$ and hence the correct upper bound provided by playing such a strategy on the value $V(Q,T,m)$ for any $T \geq m$. In doing so we will note that $\boldsymbol{\phi}_{\text{di}}$ only provides the upper bound given in lemma 2.3.24 for certain game lengths T. We introduce the time-limited diametric attacker strategy $\boldsymbol{\phi}_{\text{tdi}}$, by limiting the distribution in commencement time for the diametric attacker strategy, which achieves the upper bound proposed by lemma 2.3.24 for a significantly greater range of game lengths. Further to this idea, we extend the notion of attacks at nodes which are spatially separated, to a generic collection of nodes, the implementation of which depends on the minimum distance between nodes. This notion of spatial separation extends the idea of node independence when nodes are not a distance at least m apart.

Recall the diametric attack strategy $\boldsymbol{\phi}_{\text{di}}$, as defined in [16], chooses to attack a pair of diametric nodes $(j,j') \in N^2$ equally for all commencement times. For $\boldsymbol{\phi}_{\text{di}}$ the set of nodes which are possibly attacked is $N_A = \{j,j'\}$. The diametric attack strategy is node-identical on $N_A = \{j,j'\}$ and non-decreasing on $N_A = \{j,j'\}$. Hence by using lemma 3.2.16 and corollary 3.2.17 we can evaluate the performance of $\boldsymbol{\phi}_{\text{di}}$. Thus giving the correct upper bound playing the strategy provides on the value of the game for all $T \geq m$.

Lemma 3.3.11. *For the game $G(Q,T,m)$ for any graph Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$\begin{aligned} V(Q,T,m) &\leq V_{\bullet,\boldsymbol{\phi}_{di}}(Q,T,m) \\ &= \max\left(\frac{1}{2}, \min\left(\frac{\gamma}{2(T-m+1)}, 1\right)\right), \end{aligned}$$

where

$$\gamma = m - \bar{d} + m(\alpha+1)_+ + (T-m+1-(\alpha+1)\bar{d})_+ + (T-m+1-(\alpha+2)\bar{d})_+,$$

in which $\alpha = \left\lfloor \frac{T-2m+1}{\bar{d}} \right\rfloor$, $\bar{d}$ is the diameter of Q and where $(x)_+ = \max(0,x)$ is the rectifier function. The upper bound is achieved by the attacker choosing a diametric attacker strategy for any diametric pair (j,j'). Moreover, for $T = m-1+(k+1)\bar{d}$ for all $k \in \mathbb{N}_0$, or as $T \to \infty$, we have

$$V(Q,T,m) \leq \max\left(\frac{1}{2}, \frac{m}{2\bar{d}}\right).$$

The proof of this lemma 3.3.11 follows by evaluating the performance of $\boldsymbol{\phi}_{\text{di}}$ using our work done on strategy reduction for the responding pure patroller walk.

Proof. We divide the proof into two cases, case 1: if $\bar{d} = d(j,j') \geq m$ and case 2: if $\bar{d} = d(j,j') < m$. Noting that the attack structure is node-symmetric and non-decreasing on $N_A = \{j,j'\}$ we can use corollary 3.2.17 to reduce the problem of finding the performance of $\boldsymbol{\phi}_{\text{di}}$ to

$$V_{\bullet,\boldsymbol{\phi}_{\text{di}}}(Q,T,m) = \max_{\omega \in \Omega_\uparrow} P(\omega, \boldsymbol{\phi}_{\text{di}})$$

Case 1: In the case of $\overline{d} \geq m$ we know that by lemma 3.2.16 that $\Omega_{\uparrow} = \{\omega_1, \omega_2\}$ in which $\omega_1 = ((j, T-1))$ and $\omega_2 = ((j', T-1))$. As these nodes are node-symmetric we can without loss of generality assume that $\omega = ((j, T-1))$ is the only walk to consider for the maximum. Therefore, the performance of $\boldsymbol{\phi}_{\text{di}}$ is

$$V_{\bullet,\boldsymbol{\phi}_{\text{di}}}(Q, T, m) = P(\omega_1, \boldsymbol{\phi}_{\text{di}}) = \sum_{t=0}^{T-m+1} \frac{1}{2(T-m+1)} = \frac{1}{2}.$$

Hence $V(Q, T, m) \leq \frac{1}{2}$ when $\overline{d} \geq m$.

Case 2: In the case of $\overline{d} < m$ we cannot simply use prior results to reduce the set of walks to a single element, however we can still assume without loss of generality that j can be the initial node. Then the only move, wait walks to consider start at j wait some time and then move to j' at which they do not wait before returning to j again not waiting and returning to j' repeating this alternating between j and j' for the entire time-horizon finishing by arriving at j or j' at time $T-1$. That is $\Omega_{\uparrow} = \{\omega(x) \mid x \in \{0, ..., T-1\}\}$ where $\omega(x)$ is such that

$$j_i(\omega(x)) = \begin{cases} j & \text{if } i \text{ is odd,} \\ j' & \text{if } i \text{ is even,} \end{cases}$$

and $\nu_i(\omega(x)) = x\mathbb{I}_{\{i=1\}}$ with the number of nodes visited $k = \left\lfloor \frac{T-1-x+\overline{d}}{\overline{d}} \right\rfloor$. Then the time of visits to nodes is given by

$$t_i(\omega(x)) = \begin{cases} 0 & \text{if } i = 1, \\ x + (i-1)\overline{d} & \text{otherwise,} \end{cases}$$

and

$$n_i(\omega(x)) = \begin{cases} 0 & \text{if } i = 1, \\ \max(x + \overline{d} - m + 1, 0) & \text{if } i = 2, \\ \max(x + (i-1)\overline{d} - m + 1, x + (i-3)\overline{d} + 1, 0) & \text{otherwise.} \end{cases}$$

Therefore the payoff for responding to $\boldsymbol{\phi}_{\text{di}}$ with $\omega(x)$ is

$$\begin{aligned} P(\omega(x), \boldsymbol{\phi}_{\text{di}}) &= \sum_{t=0}^{x} \frac{1}{2(T-m+1)} + \sum_{t=(x+\overline{d}-m+1,0)}^{\overline{d}+x} \frac{1}{2(T-m+1)} \\ &\quad + \sum_{i=2}^{k} \sum_{t=n_i(\omega(x))}^{x+(i-1)\overline{d}} \frac{1}{2(T-m+1)} \\ &= \min\left(\frac{\Delta}{2(T-m+1)}, 1\right), \end{aligned} \tag{3.14}$$

where

$$\begin{aligned} \Delta = x + 1 + \mathbb{I}_{\{x+\overline{d} \leq T-m\}} \min(m, x + \overline{d} + 1) + m(k-2)_+ \\ + (T - x - (k-1)\overline{d})_+ + (T - x - k\overline{d})_+. \end{aligned}$$

Thus to find the performance of ϕ_{di} we seek to maximize the payoff in equation (3.14) by choosing x. By considering x and $x+1$ it is clear to see that for $x \leq m - \overline{d} - 1$ the payoff is non-decreasing and for $x \geq m - \overline{d} - 1$ the payoff is non-increasing. Therefore, the best choice of x is $x^* = m - \overline{d} - 1$ and furthermore

$$\begin{aligned} V_{\bullet,\phi_{\text{di}}}(Q,T,m) &= P(\omega(x^*), \phi_{\text{di}}) \\ &= \min\left(\frac{\begin{bmatrix} m - \overline{d} + m(\alpha+1)_+ + (T - m + 1 - (\alpha+1)\overline{d})_+ \\ + (T - m + 1 - (\alpha+2)\overline{d})_+ \end{bmatrix}}{2(T-m+1)}, 1 \right). \end{aligned}$$

Hence

$$V(Q,T,m) \leq \frac{\gamma}{2(T-m+1)},$$

when $\overline{d} > m$.

The remaining part of the lemma for $T = m - 1 + (k+1)\overline{d}$ and as $T \to \infty$ follows by inspection of the above bound. □

Thus lemma 3.3.11 states that the upper bound proposed in lemma 2.3.24 (from [16]) provided by playing ϕ_{di} only holds for certain values of the game length T, or in the infinite time-horizon patrolling game. While the upper bound given in lemma 2.3.24 (from [16]) does hold for these certain game lengths, it does not hold for all game lengths $T \geq m$ and its implementation as an optimal attacker strategy in the solution to the patrolling game on the line graph, as in lemma 2.3.29 (from [107]), is consequently invalid. Figure 3.3.2 shows the discrepancy between the upper bound given by lemma 2.3.24 and our corrected lemma 3.3.11 for $m \geq \overline{d}$ as the game length T changes.

As mentioned in section 2.3.4, the solution to the game $G(L_n, T, m)$ relies on the bound of $V(Q,T,m) \leq \frac{m}{2\overline{d}}$ to hold for all $T \geq m$. Which as given in [107] is achieved by the diametric attacker strategy ϕ_{di}. However lemma 3.3.11 shows that the optimality is actually reduced to game lengths with $T = m - 1 + (k+1)\overline{d}$ for all $k \in \mathbb{N}_0$ or for $T = \infty$. This poses an issue with the patrolling game on the line graph when $m \in M_2^{L_n}$. We will soon see that it is possible to get a lesser restriction on the game length of $T \geq m + \overline{d} - 1$, by limiting the distribution of the diametric attacker strategy in the commencement time. To adapt the diametric attacker strategy we limit the distribution of commencement times to $\tau \in \{0, ..., \overline{d} - 1\}$, to create the time-limited diametric attacker strategy.

Definition 3.3.12. For the game $G(Q,T,m)$ ($Q = (N,E)$) a *time-limited diametric* attacker strategy using the diametric pair $(j,j') \in N^2$ is $\phi_{\text{tdi}} \in \Phi$ such that the probability of choosing the pure strategy $(j,\tau) \in \mathcal{A}$ is

$$\varphi_{i,\tau} = \begin{cases} \frac{1}{2\overline{d}} & \text{if } i \in \{j,j'\} \text{ and } \tau \in \{0,...,\overline{d}-1\}, \\ 0 & \text{otherwise.} \end{cases}$$

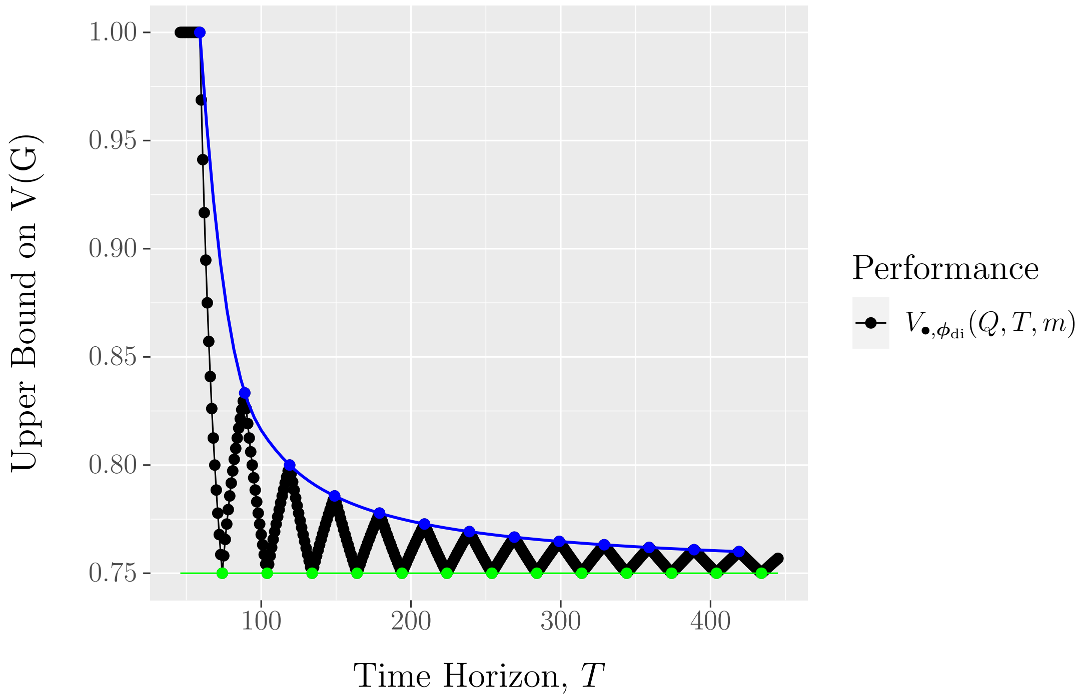


Figure 3.3.2: This graphs shows, in black, the upper bound on the value $V(Q, T, 45)$ provided by playing the diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{di}}$ for any graph Q with a diameter of $\overline{d} = 30$. Shown for game lengths $T \geq m = 45$. The green baseline shows the performance of $\boldsymbol{\phi}_{\mathrm{di}}$ as proposed by the upper bound in lemma 2.3.24 (from [16]). The green points show when the actual performance of the diametric attacker agrees with that proposed in lemma 2.3.24. The blue trend line shows that as the game length increases the upper bound tends to that given in lemma 2.3.24.

That is a time-limited diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{tdi}}$ chooses between its diametric pair of nodes uniformly and then chooses a commencement time from a limited set $\{0, ..., \overline{d}-1\}$. This limited set is the difference between $\boldsymbol{\phi}_{\mathrm{tdi}}$ and $\boldsymbol{\phi}_{\mathrm{di}}$ and arises from the fact that $\boldsymbol{\phi}_{\mathrm{di}}$ does provide the proposed bound in lemma 2.3.24 when $T = \overline{d}+m-1$. The time-limited diametric attacker strategy will provide the proposed upper bound with a slight limitation to the game length. We note that it is possible to limit the commencement set to $\{0, ..., k\overline{d}-1\}$ for some $k \in \mathbb{N}$, to still achieve the proposed bound, however such a choice only leads to a greater restrictions on game lengths for which the time-limited diametric attack is a valid strategy when $k \neq 1$.

Lemma 3.3.13. *For the game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$ and for all $T \geq m + \overline{d} - 1$ we have*

$$V(Q,T,m) \leq V_{\bullet,\boldsymbol{\phi}_{tdi}}(Q,T,m) = \max\left(\frac{1}{2}, \min\left(\frac{m}{2\overline{d}}, 1\right)\right),$$

where $\overline{d}$ is the diameter of the graph Q. The upper bound on $V(Q,T,m)$ is achieved by the attacker choosing a time-limited diametric attacker strategy $\boldsymbol{\phi}_{tdi}$, using any diametric pair (j, j').

The proof of lemma 3.3.13 follows the same idea as that for lemma 3.3.11, that is evaluating the performance of $\boldsymbol{\phi}_{\mathrm{tdi}}$. Finding $V_{\bullet,\boldsymbol{\phi}_{\mathrm{tdi}}}(Q,T,m)$ is much easier than $V_{\bullet,\boldsymbol{\phi}_{\mathrm{di}}}(Q,T,m)$ as $\boldsymbol{\phi}_{\mathrm{tdi}}$ uses a limited set to place non-zero probability pure attacks at in the commencement time.

Proof. We divide the proof into two cases, case 1: if $\overline{d} = d(j, j') \geq m$ and case 2: if $\overline{d} = d(j, j') < m$. Noting that the attack structure is node-symmetric and non-increasing on $N_A = \{j, j'\}$ we can use corollary 3.2.17 and lemma 3.2.14 to reduce the problem of finding the performance of $\boldsymbol{\phi}_{\mathrm{tdi}}$ to

$$V_{\bullet,\boldsymbol{\phi}_{\mathrm{tdi}}}(Q,T,m) = V_{\bullet,\boldsymbol{\phi}_{\mathrm{tdi}}}(Q,\overline{d}+m-1,m) = \max_{\omega \in \Omega_{\uparrow}} P(\omega, \boldsymbol{\phi}_{\mathrm{tdi}})$$

Case 1: In the case of $\overline{d} \geq m$ we know that by lemma 3.2.16 that $\Omega_{\uparrow} = \{\omega_1, \omega_2\}$ in which $\omega_1 = ((j, \overline{d}+m-1))$ and $\omega_2 = ((j', \overline{d}+m-1))$. As these nodes are node-symmetric we can without loss of generality assume that $\omega = ((j, \overline{d}+m-1))$ is the only walk to consider for the maximum. Therefore, the performance of $\boldsymbol{\phi}_{\mathrm{tdi}}$ is

$$V_{\bullet,\boldsymbol{\phi}_{\mathrm{tdi}}}(Q,\overline{d}+m-1,m) = P(\omega_1, \boldsymbol{\phi}_{\mathrm{tdi}}) = \sum_{t=0}^{\overline{d}-1} \frac{1}{2\overline{d}} = \frac{1}{2}.$$

Hence $V(Q,T,m) \leq \frac{1}{2}$ when $\overline{d} \geq m$.

Case 2: In the case of $\overline{d} < m$ we can again look at what elements are in $\Omega_{\uparrow}$. For the sequence of nodes it must go either $j, j', j, ...$ or $j', j, j', ...$ and as j and j' are node-symmetric we can without loss of generality assume that the sequence of nodes is $j, j', j, \ldots$. Therefore all move,wait walks in $\Omega_\uparrow$ start at j and then wait for some time before alternating between j' and j. That is $\Omega_\uparrow = \{\omega(x) \mid x \in \{0, \ldots, \bar{d}+m-1\}\}$ where $\omega(x)$ is such that

$$j_i(\omega(x)) = \begin{cases} j & \text{if } i \text{ is odd,} \\ j' & \text{if } i \text{ is even,} \end{cases}$$

and $\nu_i(\omega(x)) = x\mathbb{I}_{\{i=1\}}$ with the number of nodes visited $k = \left\lfloor \frac{2\bar{d}+m-2-x}{\bar{d}} \right\rfloor$. Then the time of visits to nodes is given by

$$t_i(\omega(x)) = \begin{cases} 0 & \text{if } i = 1, \\ x + (i-1)\bar{d} & \text{otherwise.} \end{cases}$$

and

$$n_i(\omega(x)) = \begin{cases} 0 & \text{if } i = 1, \\ \max(x+\bar{d}-m+1, 0) & \text{if } i = 2, \\ \max(x+(i-1)\bar{d}-m+1, x+(i-3)\bar{d}+1, 0) & \text{otherwise.} \end{cases}$$

Therefore the payoff for responding to $\boldsymbol{\phi}_{\mathrm{di}}$ with $\omega(x)$ is

$$\begin{aligned} P(\omega(x), \boldsymbol{\phi}_{\mathrm{di}}) &= \sum_{t=0}^{\min(x,\bar{d}-1)} \frac{1}{2\bar{d}} + \sum_{t=\max(x+\bar{d}-m+1,0)}^{\min(x+\bar{d},\bar{d}-1)} \frac{1}{2\bar{d}} + \sum_{i=3}^{k} \sum_{t=n_i(\omega(x))}^{\min(x+(i-1)\bar{d},\bar{d}-1)} \frac{1}{2\bar{d}} \\ &= \sum_{t=0}^{\min(x,\bar{d}-1)} \frac{1}{2\bar{d}} + \sum_{t=\max(x+\bar{d}-m+1,0)}^{\bar{d}-1} \frac{1}{2\bar{d}} \\ &\quad + \sum_{t=\max(x+2\bar{d}-m+1,x+1,0)}^{\bar{d}-1} \frac{1}{2\bar{d}}. \end{aligned} \tag{3.15}$$

Thus to find the performance of $\boldsymbol{\phi}_{\mathrm{tdi}}$ we seek to maximize the payoff in equation (3.15) by choosing x. It is clear from the equation that the choice of $x^* = \bar{d}-1$ maximizes it and therefore the performance of $\boldsymbol{\phi}_{\mathrm{tdi}}$ is

$$\begin{aligned} V_{\bullet,\boldsymbol{\phi}_{\mathrm{tdi}}}(Q, \bar{d}+m-1, m) &= P(\omega(x^*), \boldsymbol{\phi}_{\mathrm{tdi}}) \\ &= \sum_{t=0}^{\bar{d}-1} \frac{1}{2\bar{d}} + \sum_{t=2\bar{d}-m+1}^{\bar{d}+m-1} \frac{1}{2\bar{d}} + 0 \\ &= \frac{\bar{d}}{2\bar{d}} + \frac{\min(m-\bar{d}, \bar{d})}{2\bar{d}} = \min\left(\frac{m}{2\bar{d}}, 1\right). \end{aligned}$$

Hence $V(Q, T, m) \leq \min\left(\frac{m}{2\bar{d}}, 1\right)$ when $\bar{d} \geq m$. □

From lemma 3.3.13 it is clear that the proposed upper bound is true for all $T \geq m + \bar{d} - 1$. This can be seen in figure 3.3.3 where the performance of the time-limited diametric attacker strategy provides an upper bound which is strictly lower than the upper bound from the diametric bound.

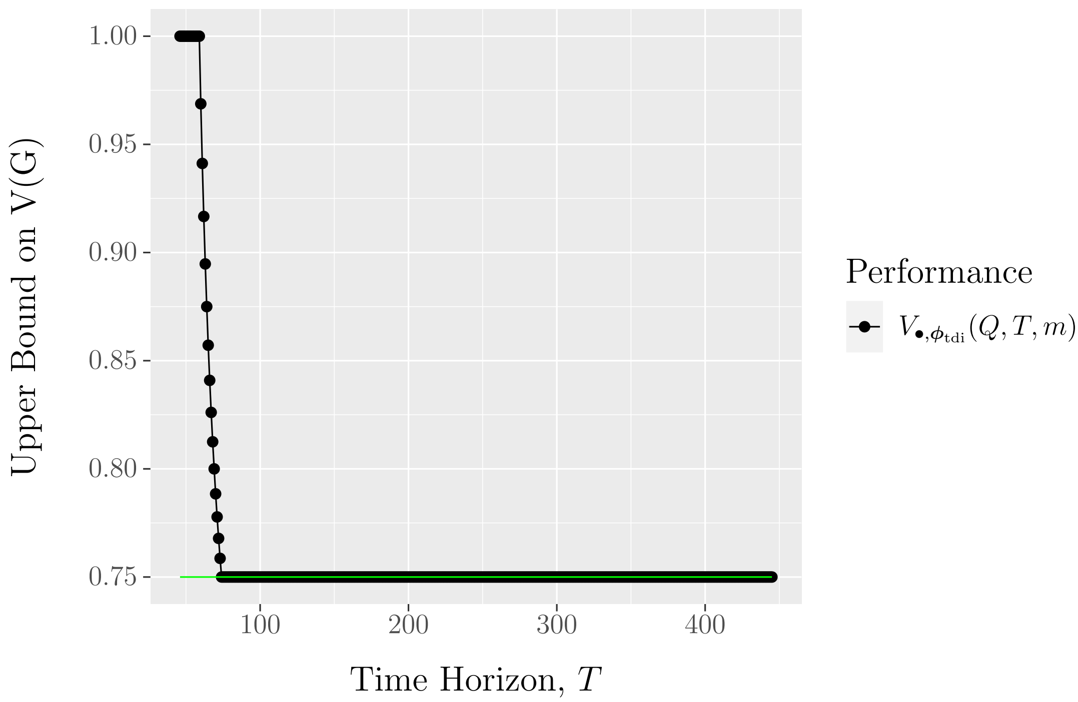


Figure 3.3.3: This graphs shows, in black, the upper bound on the value $V(Q, T, 45)$ of the game provided by the time-limited diametric attacker strategy $\boldsymbol{\phi}_{\mathrm{tdi}}$ for any graph with a diameter of $\overline{d} = 30$, shown for game lengths $T \geq m = 45$. The green baseline shows the upper bound as proposed in lemma 2.3.24 (from [16]) for a diametric attacker strategy.

While we are not able to prove the proposed bound is true for all $T \geq m$, we are able to show it for $T \geq m + \bar{d} - 1$. Hence we can edit lemma 2.3.29's statement in the region $M_2^{L_n}$ to become:

- for $T \geq m + n - 2$ and $m \in M_2^{L_n}$,
$$V(L_n, T, m) = \frac{m}{2(n-1)},$$
achieved by the random oscillation patroller strategy and the time-limited diametric attacker $\boldsymbol{\phi}_{\text{tdi}}$.

The proof of the time-limited diametric attack does not rely on the diameter of the graph but the distance between the two nodes that from the chosen diametric pair. Therefore, by considering two nodes at an arbitrary distance apart we arrive at the following corollary.

Corollary 3.3.14. *For a game $G(Q, T, m)$ for any graph Q, for any pair of nodes $(j, j') \in N^2$, for all $1 \leq m \leq 2d(j, j')$ and for all $T \geq m + d - 1$ we have*
$$V(Q, T, m) \leq \max\left(\frac{1}{2}, \frac{m}{2d(j, j')}\right).$$

Clearly using any pair (j, j') such that $d(j, j') < \bar{d}$ in corollary 3.3.14 provides a worse upper bound than lemma 3.3.13. However by allowing for general distance we can consider using a set of nodes, D, rather than just a pair. We choose D such all nodes attacked are exactly a distance d apart, i.e. $d(j, j') = d$ for all $j, j' \in D$. Then using the same principle behind the time-limited diametric attacker strategy, we can develop a much larger class of attacker strategies. As the nodes in D essentially form a polygon with sides of length d, we call such an attacker strategy a *d-polygonal* attacker strategy.

Definition 3.3.15. For the game $G(Q, T, m)$ and a set of nodes $D \subset N$ such that there exists a $d \in \mathbb{N}$ such that $d(j, j') = d$ for all $j, j' \in D$, the *$d-$polygonal* attack strategy $\boldsymbol{\phi}_{\text{poly}}$, using set D, is such that the probability of playing the pure strategy $(j, \tau) \in \mathcal{A}$ is
$$\varphi_{j,\tau} = \begin{cases} \frac{1}{|D|d} & \text{for } j \in D \text{ and } \tau \in \{0, ..., d-1\}, \\ 0 & \text{otherwise.} \end{cases}$$

A d-polygonal attack strategy using D for some d gives rise to a larger class of bounds than the time-limited diametric bound, which is a $\bar{d}$-polygonal attack strategy using $\{j, j'\}$ where $d(j, j') = \bar{d}$.

Lemma 3.3.16. *For the game $G(Q, T, m)$ for any graph Q, for all $1 \leq m \leq |D|d$ and for all $T \geq m + d - 1$ we have*
$$V(Q, T, m) \leq V_{\bullet, \boldsymbol{\phi}_{poly}}(Q, T, m) = \max\left(\frac{1}{|D|}, \min\left(\frac{m}{|D|d}, 1\right)\right),$$
achieved by the d-polygonal attacker strategy $\boldsymbol{\phi}_{poly}$, using the set D.

The proof lemma 3.3.16 follows by evaluating the performance of the d-polygonal attacker strategy $\boldsymbol{\phi}_{\text{poly}}$. As this is extremely similar to the proof of lemma 3.3.13 we leave the proof to appendix A.2.

From lemma 3.3.16, we can see that for a chosen distance d, the best upper bound uses the set D^*, where D^* is the set of maximal cardinality, containing nodes a distance of exactly d apart. Henceforth we will call the d-polygonal attack strategy that uses a maximal set D^* *optimized*. While a d-polygonal attacker strategy requires the set of nodes to be exactly d apart. It is easy to relax this assumption to assume that the nodes are *at least* a distance d apart from one another.

Definition 3.3.17. For the game $G(Q, T, m)$ and a set of nodes $D \subset N$ such that there exists a $d \in \mathbb{N}$ such that $d(j, j') \geq d$ for all $j, j' \in D$, the $d-$*exterior-polygonal* attack strategy $\boldsymbol{\phi}_{\text{epoly}}$, using set D, is such that the probability of playing the pure strategy $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{|D|d} & \text{for } j \in D \text{ and } \tau \in \{0, ..., d-1\}, \\ 0 & \text{otherwise.} \end{cases}$$

Exterior-polygonal attack strategies allow us to form an attacker strategy given a subset of nodes $N_1 \subset N$. In particular we can form a exterior-polygonal attacker strategy using $D = N_1$ with a distance of

$$d = \min_{(j,j') s.t j,j' \in N, j \neq j'} d(j, j').$$

Using an exterior-polygonal attacker strategy gives a bound similar to that for the polygonal attacker strategy (lemma 3.3.16).

Lemma 3.3.18. *For the game $G(Q, T, m)$ for any graph Q, for all $1 \leq m \leq |D|d$ and for all $T \geq m + d - 1$ we have*

$$V(Q, T, m) \leq \max\left(\frac{1}{|D|}, \frac{m}{|D|d}\right),$$

achieved by the d-exterior-polygonal attacker strategy $\boldsymbol{\phi}_{epoly}$, using the set D.

It is intuitive that increasing the distance from d to d' can do no harm to the attacker strategy, as the patroller must travel further than in a d-polygonal attacker strategy on the same number of nodes. We prove lemma 3.3.18 formally by using simplification to create a graph in which the nodes in D are exactly a distance of d apart.

Proof. Consider $\mathcal{Q}^{-l}$ a simplification mapping which maps the graph Q to Q^{-l} (with node mapping $\mathcal{N}^{-l}$), for some $l \in \mathbb{N}$, such that for all $j, j' \in D$ the distance in Q' has $d(\mathcal{N}^{-l}(j), \mathcal{N}^{-l}(j')) = d$. Then we have

$$V(Q, T, m) \leq V(Q^{-l}, T, m) \leq \max\left(\frac{1}{|D|}, \frac{m}{|D|d}\right),$$

with the upper bound on the game $G(Q^{-l}, T, m)$ being given by the performance of a d-polygonal attacker strategy using the set $\mathcal{N}^{-l}(D)$. This d-polygonal attacker strategy can be embedded to create the d-exterior polygonal attacker strategy, using D, for use in the game $G(Q, T, m)$. □

Note that such a simplification map, in the above proof, is always possible, as repeated node identification of the attacked node and the nearest node along the shortest path to another node of length greater than d, eventually yields a shortest path of length exactly length d.

As with a d-polygonal attacker strategy, it is clear that for a given distance d, using D_d^* a maximal cardinality set of nodes at least a distance d apart provides the best upper bound. We call the d-exterior polygonal attacker strategy on set D_d^*, the *optimized* d-exterior polygonal attacker strategy. Furthermore it is now clear that the bound provided by lemma 3.3.18 depends on the distance d for the optimized d-exterior polygonal attacker strategy. For each $d \in \{1, .., \overline{d} - 1\}$ finding D_d^* provides us with a bound. For $d \geq m$, the upper bound becomes,

$$V(Q, T, m) \leq \frac{1}{|D_d^*|}.$$

Hence it is clear that for $d \geq m$, the best upper bound is provided by the largest cardinality among the sets $D_m^*, D_{m+1}^*, \ldots$. Clearly D_m^* is the largest cardinality set and by definition $D_m^* = \mathcal{L}_{Q,T,m}$. On the other hand for $d < m$, then the upper bound becomes,

$$V(Q, T, m) \leq \frac{m}{d|D_d^*|}.$$

In this case we can not reduce the best choice of $d \in \{1, ...m - 1\}$ as the cardinality of the set D_d^* is multiplied by d. We summarize this optimization of the exterior polygonal attack strategy in the following theorem. The exterior polygonal attacker strategies can be thought as a generalisation of both the independent attacker strategy and diametric attacker strategy.

Theorem 3.3.19. *For the game $G(Q, T, m)$ for any graph Q, for all $m \geq 1$ and for all $T \geq m + \overline{d} - 1$ we have*

$$V(Q, T, m) \leq \min\left(\frac{1}{\mathcal{L}_{Q,T,m}}, \min_{d \leq m} \frac{m}{|D_d^*|d}, \right),$$

where D_d^* is a maximal cardinality set of nodes in which each pair is at least a distance d apart.

The proof of theorem 3.3.19 is simple as it combines the best result from lemma 3.3.18 and then uses lemma 2.3.21. The exact bound provided by theorem 3.3.19 is dependent on both the attack length m and the best exterior polygonal distance d (or equivalently subset of nodes).

The optimized d-exterior polygonal attacker strategy becomes an already known attacker strategies for particular choices of d, including the position-uniform attacker strategy (when $d = 1$) and the bipartite attacker strategy (when $d = 2$), the independence attacker strategy (when $d = m$). Theorem 3.3.19 provides an improvement over lemma 3.3.13. To emphasize the use of exterior polygonal attack strategies (for a not currently identified attacker strategy), we look at applying it to the cyclic graph, C_n. For the game $G(C_n, T, m)$ for all $n \geq 1$, for all $m \geq 1$ and for all $T \geq m$, an optimal attacker strategy is the position-uniform attacker strategy $\boldsymbol{\phi}_{\text{pu}}$ (or equivalently an optimized 1-polygonal attacker strategy). However it is possible to get various other optimal attacker strategies for different distances. For C_n, we know $\overline{d} = \left\lfloor \frac{n}{2} \right\rfloor$ and hence for the exterior polygonal attack we can consider distances $d = 1, 2, ..., \left\lfloor \frac{n}{2} \right\rfloor$.

For each of these distances such that $m \geq d$, the maximization of the product $d|D_d^*|$ is required. For C_n and given d, maximizing $|D_d^*|$ is easy as we can simply divide the nodes between this distance and rounding down, so $|D_d^*| = \left\lfloor \frac{n}{d} \right\rfloor$. Hence to maximize $d|D_d^*|$, we need to choose d such that $n = kd$ for some $k \in \mathbb{N}$ and all of these generate the bound $\frac{m}{n}$. Therefore, we get alternative optimal attacker strategies for C_n where $m \geq d$. Also note that in these cases the attacks will be exactly d away from each other and so it is just an optimized d-polygonal attacker strategy.

Corollary 3.3.20. *For the game $G(C_n, T, m)$, where $n = kd$ for some $k \in \mathbb{N}$ and some $d = 1, ..., \left\lfloor \frac{n}{2} \right\rfloor$, for all $T \geq m + d - 1$ and for all $m \geq d$, the optimized d-polygonal attack strategy is optimal.*

We illustrate a simple analysis of all exterior-polygonal attacks ($d = 1, 2, 3, 4$) for the game $G(C_9, T, m)$ in example 3.3.21.

Example 3.3.21. For the game $G(C_9, T, m)$ for some $T \geq m + 2$ and some $m \geq 3$, we look at the optimized d-exterior-polygonal attacker strategies using $d = 1, 2, 3, 4$. The choice of nodes such attacker strategies make are shown in figure 3.3.4, such that nodes in D_d are filled grey. In addition we state the upper bound provided by lemma 3.3.18. By comparing the upper bounds for $d = 1, 2, 3, 4$ we get that either $d = 1$ or $d = 3$ provide the best bounds and in particular for the 3(-exterior)-polygonal attacker strategy the set of nodes used is $D = \{1, 4, 7\}$. In addition we note that equivalent attacker strategies can be made by using the sets $D_2 = \{2, 5, 8\}$ or $D_3 = \{3, 6, 9\}$.

Another simple analysis can be conducted for the game on the line graph $G(L_n, T, m)$. By theorem 3.3.19 a d-exterior-polygonal attack provides an upper bound of

$$V(L_n, T, m) \leq \max\left(\frac{1}{1 + \left\lfloor \frac{n-1}{d} \right\rfloor}, \frac{m}{d + d\left\lfloor \frac{n-1}{d} \right\rfloor}\right) \ \forall m \geq 1\ \forall T \geq m + n - 2.$$

For this upper bound if we consider $m \geq d$, it is clear that $d = \overline{d} = n - 1$ provides the best upper bound by the exterior polygonal attack strategy which is equivalent to the time-limited diametric attack strategy. Therefore no alternative exterior-polygonal attackers strategy exists.

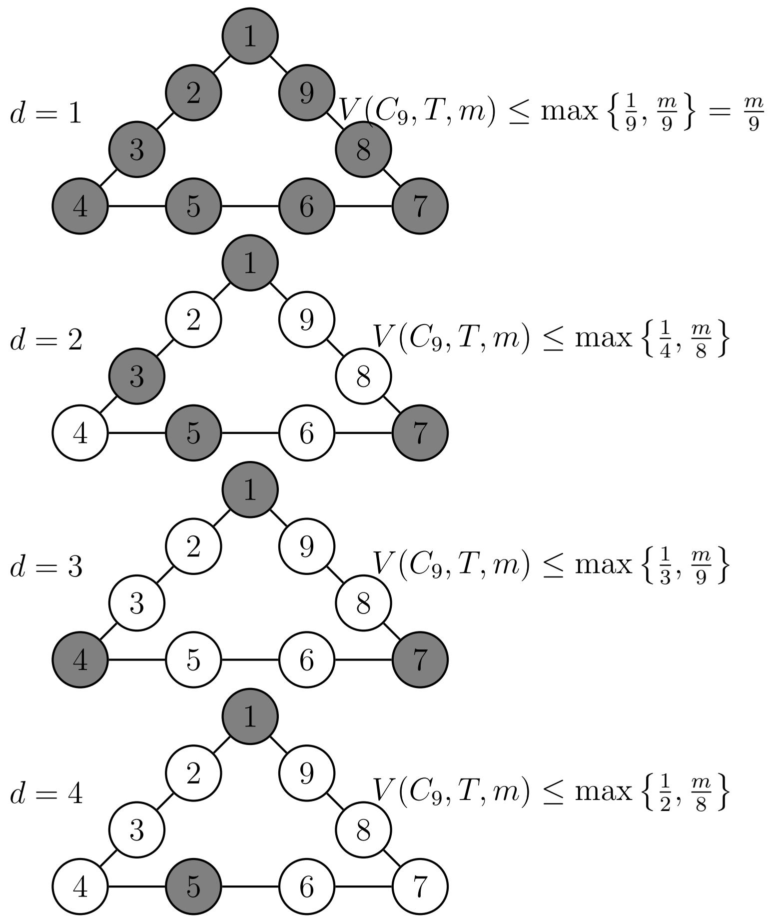


Figure 3.3.4: The cyclic graph on nine vertices C_9, with the four possible exterior-polygonal attacks at polygonal distances $d = 1, 2, 3, 4$, with attacked nodes coloured grey. The upper bound provided by theorem 3.3.19 is stated beside each strategy.

3.3.4 Embedding Hamiltonian cycles

In this section we obtain a 'Hamiltonian like' lower bound, similar to that in lemma 2.3.7 albeit for non-Hamiltonian graphs $Q \notin \mathcal{H}$. This is done by taking a closed walk which contains every node, but which, unlike a Hamiltonian cycle, may repeat some nodes in order to do so. Note a closed walk always ends were it starts but this is not considered a repeat of that node and that by definition for $Q \notin \mathcal{H}$ we know that such a closed walk must have repeated nodes.

Definition 3.3.22. For a graph $Q = (N, E)$ a *full-node* cycle is a walk $W : \{0, ..., L\} \to N$ of *length* $L \in \mathbb{N}$ such that $W(L) = W(0)$ and for all $j \in N$ there exists some $t \in \{0, ..., L-1\}$ such that $W(t) = j$.

That is a full-node cycle is a closed walk that contains each node of the graph at least once and the *length* of such a full-node cycle is the number of nodes (not necessarily distinct) in the W omitting the last node. For convenience we omit the final node $W(L)$ when writing the full-node cycle as a vector $W \in N^L$. A full-node cycle of length $|N|$ is a Hamiltonian cycle, as no node will be repeated.This is the reason we refer to a full-node cycle as a cycle rather than a closed walk. For the graph L_4 an example of a full-node cycle is $W_{\text{FNC}} = (1, 1, 2, 3, 4, 3, 4, 3, 2)$, which is of length 9. In W_{FNC} the nodes 1,2 and 4 are repeated once and the node 3 is repeated twice for a total repetition of 5 nodes. Note that for W_{FNC} we omit the final node $(W(7) = 1)$ in the vector form as it is not consider a repeated node.

For any graph $Q = (N, E)$ any full-node cycle W is of length $|N| + r$ for some $r \geq 0$, where r is the number of nodes which are repeated. We can then consider an r-expansion of Q, by node-splitting each node according to the number of times it is repeated in W, to form $Q^{+r} = (N^{+r}, E^{+r})$. Such an expansion is called a *Hamiltonian expansion* as the graph $Q^{+r} = (N^{+r}, E^{+r}) \in \mathcal{H}$. For this expansion, we will make the following choice: for each parent node, the first child node is adjacent to only the node prior to the parent in the full-node cycle and the other child node; the second child node is adjacent to the node subsequent to the parent in the full-node cycle and the other child node; with all remaining adjacencies can be distributed between the children arbitrarily. While formally this arbitrary distribution of remaining adjacencies is needed, the patroller does not need to make use of them as the graph Q^{+r} is Hamiltonian and so we can make the decision to ignore these remaining adjacencies. This decision to ignore the remaining adjacencies makes figures much easier to follow, as the graph Q^{+r} can be seen as cycle $C_{|N|+r}$ under some relabelling (isomorphism).

Using corollary 3.3.8 we know that for all $m \geq 1$ and for all $T \geq m$ that $V(Q^{+r}, T, m) \leq V(Q, T, m)$, with the lower bound on the game $G(Q, T, m)$ generated by the embedding the optimal strategy in the game $G(Q^{+r}, T, m)$. As previously mentioned Q^{+r} is Hamiltonian and so $F = \mathcal{N}^{+r}(W)$ is a Hamiltonian cycle (where W is the full-node cycle used for the expansion and $\mathcal{N}^{+r}$ is the node mapping of the expansion). In addition

$$V(Q^{+r}, T, m) = \frac{m}{|N| + r},$$

where $\boldsymbol{\pi}_{\mathrm{rH}} \in \Pi(Q^{+r}, T, m)$, using Hamiltonian cycle F, is optimal. Note that as

$$V(C_{|N|+r}, T, m) = \frac{m}{|N|+r}$$

and $\boldsymbol{\pi}_{\mathrm{rH}} \in \Pi(C_{|N|+r}, T, m)$, using Hamiltonian cycle F is optimal, we have no need for the reaming adjacencies previously mentioned in the construction of Q^{+r} and hence can think of Hamiltonian expansions as cyclic graphs. Embedding $\boldsymbol{\pi}_{\mathrm{rH}} \in \Pi(Q^{+r}, T, m)$, using Hamiltonian cycle F, into the game $G(Q, T, m)$ will therefore give us a strategy $\boldsymbol{\pi} \in \Pi(Q, T, m)$ such that

$$V_{\boldsymbol{\pi},\bullet}(Q, T, m) \geq \frac{m}{|N|+r}$$

Definition 3.3.23. For a graph Q with a full-node cycle W we define the *random full-node cycle* patroller strategy $\boldsymbol{\pi}_{RFNC}(W)$, using full-node cycle W, as the embedding of $\boldsymbol{\pi}_{\mathrm{rH}} \in \Pi(C_{|N|+r}, T, m)$, using Hamiltonian cycle $\mathcal{N}^{+r}(W)$ (where $\mathcal{N}^{+r}$ is the node mapping of the Hamiltonian expansion from Q to $C_{|N|+r}$).

Thus we have that

$$V(Q, T, m) \geq V_{\boldsymbol{\pi}_{RFNC}(W),\bullet}(Q, T, m) \geq \frac{m}{|N|+r}, \tag{3.16}$$

for any full-node cycle W of length $|N| + r$ for some $r \geq 0$. While infinitely many full-node cycles exist for a graph, as nodes may be repeated, it is clear from equation 3.16 that the best lower bound has the minimal number of repetitions r. Therefore, it is clear that we should use a full-node cycle with the minimal length.

Definition 3.3.24. For a graph Q we say a full node cycle W is a *minimal full-node cycle* if the length of W is the minimal length of all full-node cycles for Q. We let W^Q_{MFNC} denote a full-node cycle. Likewise we call a random full-node cycle patroller strategy, using a minimal full-node cycle W^Q_{MFNC}, a *random minimal full-node cycle*, letting $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}} = \boldsymbol{\pi}_{RFNC}(W^Q_{\mathrm{MFNC}})$.

Note that similar to the Hamiltonian cycle, in order to achieve a lower bound, we need only consider one minimal full-node cycle as all give rise to the same bounds. We remark that the random full-node cycle does not start at nodes uniformly and follow the cycle, as in the random Hamiltonian, but starts at a position along the full-node cycle uniformly. This is an important distinction as subsequent movement may change depending on the selected position in the full-node cycle. We now present example 3.3.25 which shows the use of Hamiltonian expansion, and random full-node cycles.

Example 3.3.25. For the graph Q as seen in figure 3.3.5, we present two Hamiltonian expansions Q^{+3} and Q^{+2} also seen in the figure. The Hamiltonian expansions of Q into Q^{+3} and Q^{+2} arise from expanding the full-node cycles $W_1 = (1, 2, 1, 3, 4, 5, 6, 4, 3)$ and $W_2 = (1, 2, 3, 4, 5, 6, 4, 3)$ respectively. We remark that due to our decision about adjacency inheritance, $Q^{+3} \equiv C_9$ and $Q^{+2} \equiv C_8$ under a relabelling (isomorphism). By using the patroller strategies $\boldsymbol{\pi}_{\mathrm{RNFC}}(W_1)$ and

$\boldsymbol{\pi}_{\text{RNFC}}(W_2)$ we get that for all $m \geq 1$ and for all $T \geq m$ that $V(Q,T,m) \geq \frac{m}{9}$ and $V(Q,T,m) \geq \frac{m}{8}$ respectively. Hence it is clear that using W_2 gives a better lower bound in the random full-node cycle than W_3. To implement $\boldsymbol{\pi}_{\text{RNFC}}(W_2)$ the patroller should choose to either:

- Start at nodes $1,2,5,6$ each with probability $\frac{1}{8}$ and follow the full-node cycle.
- Start at nodes 3 with probability $\frac{1}{4}$ and then with equal probability choose to follow the full-node cycle starting at the third or eighth position (follow either $(3,4,5,6,4,3,1,2,3)$ or $(3,1,2,3,4,5,6,4,3)$ both with probability $\frac{1}{2}$).
- Start at nodes 4 with probability $\frac{1}{4}$ and then with equal probability choose to follow the full-node cycle starting at the fourth or seventh position (follow either $(4,5,6,4,3,1,2,3,4)$ or $(4,3,1,2,3,4,5,6,4)$ both with probability $\frac{1}{2}$).

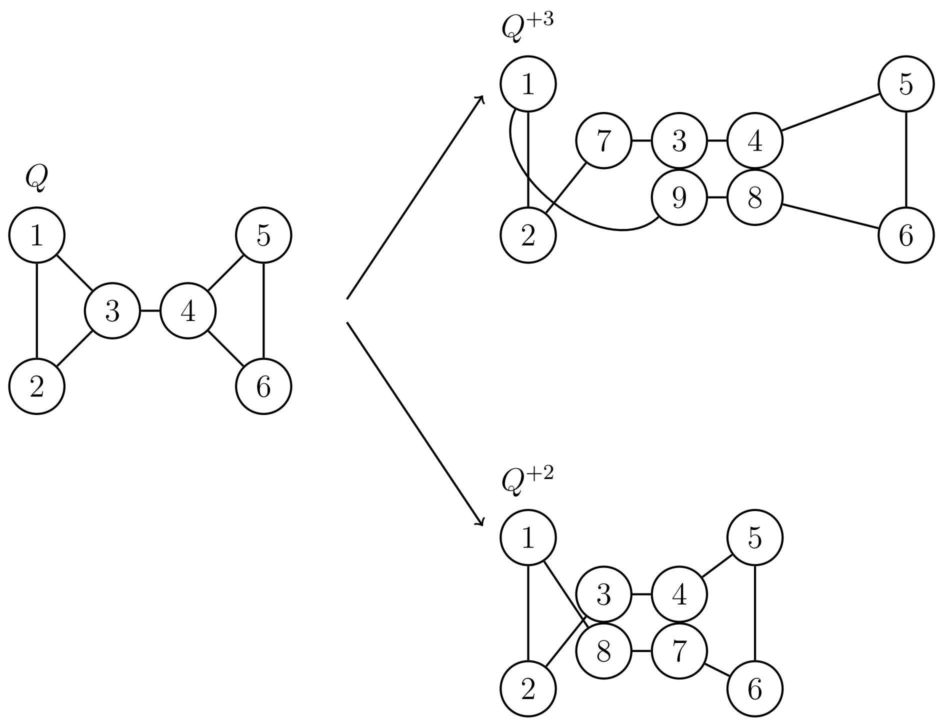


Figure 3.3.5: Graphs Q and two different Hamiltonian expansions Q^{+3} and Q^{+2} used in example 3.3.25.

Having seen how to achieve the bound as in equation (3.16) for any random full-node cycle we arrive at the following theorem for any random minimal full-node cycle.

Theorem 3.3.26. *For the game $G(Q,T,m)$, for any graph Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q,T,m) \geq \frac{m}{L} = \frac{m}{|N| + r^*},$$

where L is the length of W^Q_{MFNC} and r^* is the number of nodes repeated in W^Q_{MFNC} repetitions (as $L = |N| + r^*$). The lower bound is achieved by a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}^Q_{RMFNC}$. Furthermore if $m \geq L$

$$V(Q, T, m) = 1,$$

achieved by the (random) minimal full-node cycle patroller strategy.

In theorem 3.3.26 we note that the second part follows as any minimal full-node cycle must be intercepting and hence $\mathcal{C}_{Q,T,m} = 1$. While this idea of Hamiltonian expansion has been used to solve the complete bipartite and line graph we have provided a general bound for any graph. We acknowledge that the optimal patroller strategy for the line graph when $m \in M_2^{L_n}$ (see lemma 2.3.29), called the random oscillation in [107], is a random minimal full-node cycle $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ using $W^{L_n}_{\text{MFNC}} = (1, 2, ..., n, n-1, ..., 2)$. Later, in section 3.5 we use theorem 3.3.26 to find solve the patrolling game on complete k-partite graphs for $k \geq 3$. In chapter 4 we also see that the random minimal full-node cycle is optimal for an extension of the line graph for a certain region of attack lengths.

It is easy to see, that because the full-node cycle visits nodes with different frequencies, those visited with a higher frequency have a higher chance of catching the attacker at than those visited with lower frequencies. Furthermore the timings between repeated visits is important. In the following section we will look at this weakness of minimal full-node cycles by looking at how well $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ performs at catching an attacker at each node in the graph Q. In addition in chapter 5, section 5.3, we conjecture that $\frac{m}{|N|+r^*}$ is in fact the value for all trees when $m \geq \frac{1}{2}(|N| + r^*)$, and that any random minimal full-node cycle is optimal. This conjecture is supported by intuitive reasoning and numerical evidence.

3.3.5 Weakness of the random minimal full-node cycle

In this section, we define the performance of a patroller strategy at a node and use this performance to identify the weakness of the random minimal full-node cycle strategy.

Definition 3.3.27. For a game $G(Q, T, m)$ ($Q = (N, E)$) the *performance* of a patroller strategy $\boldsymbol{\pi}$ at node $j \in N$ is defined by

$$V_{\boldsymbol{\pi},\bullet,j}(Q, T, m) = \min_{\tau \in \mathcal{T}} P(\boldsymbol{\pi}, (j, \tau)).$$

Considering the performance of $\boldsymbol{\pi}$ at each node allows us to split the problem of finding the overall performance of the strategy by finding the minimum performance amongst all nodes. That is we have

$$V_{\boldsymbol{\pi},\bullet}(Q, T, m) = \min_{j \in N} V_{\boldsymbol{\pi},\bullet,j}(Q, T, m).$$

Node	1	2	3	4
Performance	$\frac{13}{14}$	$\frac{12}{14}$	$\frac{12}{14}$	$\frac{5}{14}$

Table 3.1: Table of performance at nodes for $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_{\text{RFNC}}(W_{\text{FNC}})$, using the random full-node cycle $W_{\text{FNC}} = (1,2,3,2,2,3,1,1,3,3,1,2,1,4)$, for the game $G(K_4, 40, 5)$.

When the patroller strategy is a random full-node cycle strategy $\boldsymbol{\pi}_{\text{RFNC}}(W)$, using full-node cycle W, the performance at node $j \in N$ can be easily calculated by considering when the node is repeated (if at all) during W. Let $W = (i_1, ..., i_{|N|+r})$ then W repeats node j at indices i_{a_k} for $k \in \{1, ..., l\}$ such that $i_{a_1}, ..., i_{a_l} = j$ for some $1 \leq a_1 \leq a_2 \leq ... \leq a_l \leq |N| + r$ for some $l \in \{1, ..., |N| + r\}$. Then the performance at node j is given by

$$V_{\boldsymbol{\pi}_{\text{RFNC}},\bullet,j}(Q,T,m) = \frac{\sum_{p=1}^{l} \min(m, a_p - a_{p-1 \bmod l} \mod |N| + r)}{n + r}. \tag{3.17}$$

Example 3.3.28 illustrates the calculation the performance of a random full-node cycle strategy at a node.

Example 3.3.28. Consider for the game $G(K_4, 40, 5)$, played on the complete graph with 4 nodes K_4, the full-node cycle

$$W_{\text{FNC}} = (1,2,3,2,2,3,1,1,3,3,1,2,1,4)$$

being used to form a random full-node cycle strategy $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_{\text{RFNC}}(W_{\text{FNC}})$. To calculate the performance at node 2 $V_{\boldsymbol{\pi}_1,\bullet,2}(K_4, 40, 5)$ we need to look at the time the full-node cycle is at node 2. As $W(1), W(3), W(4), W(11) = 2$ we have that $a_1 = 2$, $a_2 = 4$, $a_3 = 5$ and $a_4 = 12$ and by equation (3.17) performance of $\boldsymbol{\pi}_1$ at node 2 is

$$V_{\boldsymbol{\pi}_1,\bullet,2}(K_4, 40, 5) = \frac{(2 - 12 \mod 14) + (4-2) + (5-4) + \min(5, 12-5)}{14} = \frac{12}{14}.$$

The performance of $\boldsymbol{\pi}_1$ at all four nodes is shown in table 3.1. Taking the minimum of these gives us the lower bound on the value which is achieved by the patroller using the strategy $\boldsymbol{\pi}_1$, hence $V(K_4, 40, 5) \geq V_{\boldsymbol{\pi}_1,\bullet}(K_4, 40, 5) = \frac{5}{14}$.

It is clear that the strategy given in example 3.3.28 is not optimal, as the value of the game is known to be 1 (for example by using $W_{\text{MFNC}}^{K_4} = (1,2,3,4)$ a Hamiltonian cycle we achieve a performance at each node of 1). However it does illuminate the fact that we should aim for a full-node cycle which repeats each node uniformly and furthermore attempts to do so every m units of time.

3.4 Patroller strategy improvement

In this section, we consider improving the performance of a baseline patroller strategy $\boldsymbol{\pi}_0 \in \Pi(Q,T,m)$, for the patrolling game $G(Q,T,m)$ $(Q = (N,E))$, where the current lower bound is given by

$$V(Q,T,m) \geq V_{\boldsymbol{\pi}_0,\bullet}(Q,T,m) = \min_{j\in N} V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m). \tag{3.18}$$

Recall that $V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m)$ is the performance of the baseline patroller strategy at node $j \in N$. As the minimal performing node, say j', under the baseline patroller strategy determines the lower bound provided by choosing said strategy, we may seek to improve the strategies performance in order to raise the overall performance of the strategy. To do so we can use additional strategies, which have a greater performance at the worst performing node. A simple choice would be to play with probability p a patroller strategy which just waits at the worst node for all time, which has a performance of 1 at this worst node and 0 for all others. For the best choice of p we should choose p such that

$$(1-p) \min_{j\in N\setminus\{j'\}} V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) \geq (1-p)V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) + p.$$

To formalise this improvement idea, we consider the creation of a *hybrid patroller strategy*, $\boldsymbol{\pi}_{\text{Hybrid}}$. This strategy is a hybrid of the baseline patroller strategy $\boldsymbol{\pi}_0$ and l chosen patroller strategies, $\boldsymbol{\pi}_i$ for $i = 1,...,l$, which aim to improve the baseline patroller strategy. In the hybrid strategy, $\boldsymbol{\pi}_{\text{Hybrid}}$, the patroller chooses to use $\boldsymbol{\pi}_i$ with probability p_i for $i = 1,...,l$. We can write the probability of playing the baseline strategy as

$$p_0 = 1 - \sum_{i=1}^{l} p_i$$

due to the fact that

$$\sum_{i=0}^{l} p_i = 1,$$

thus eliminating one choice variable. Therefore, the choice of variables p_i for $i = 1,...,l$ determines the exact hybrid patroller strategy which is being played. We say that a hybrid patroller strategy $\boldsymbol{\pi}_{\text{Hybrid}}$, for some choice of p_i for $i = 1,...,l$, is a *strict improvement* over the baseline strategy if

$$\min_{j\in N} V_{\boldsymbol{\pi}_{\text{Hybrid}},\bullet,j}(Q,T,m) > \min_{j\in N} V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m).$$

That is the lower bound provided by the hybrid strategy is strictly better than that provided by the baseline strategy. In this case we say the baseline is improved by the patroller strategies $\boldsymbol{\pi}_i$ for $i = 1,...,l$. We note that the choice of p_i for $i = 1,...,l$ affects whether the hybrid strategy is a strict improvement. Therefore we develop a program to find the best choice of p_i for $i = 1,...,l$ and hence determine if strict improvement using $\boldsymbol{\pi}_i$ for $i = 1,...,l$ is possible.

We use the following program called the *patrol improvement program* (PIP, equation 3.19), to determine the best improvement using $\boldsymbol{\pi}_i$ for $i = 1, ..., l$. We note that it is possible to convert the PIP into a linear program, by incorporating the constraint, which can be easier for computational implementation.

$$\begin{aligned} \text{maximize} \quad & \min_{j \in N} \sum_{i=0}^{l} V_{\boldsymbol{\pi}_i, \bullet, j}(Q, T, m) p_i \\ \text{subject to} \quad & \sum_{i=0}^{l} p_i = 1, \\ & p_i \in [0,1], \text{ for } i = 0, ..., l. \end{aligned} \tag{3.19}$$

The optimal solution of the PIP gives us this best choice of p_i for $i = 1, .., l$ and hence the best hybrid strategy that uses $\boldsymbol{\pi}_i$ for $i = 0, ..., l$. Hence the optimal value of the PIP gives the performance of the best hybrid strategy using $\boldsymbol{\pi}_i$ for $i = 1, ..., l$, and thus determines if using these improvement strategies leads to a strict improvement.

While the PIP can find the best choice of probabilities p_i for a given set of patroller strategies $\boldsymbol{\pi}_i$ for $i = 1, .., l$, the bound it generates depends on these patroller strategies. Including more strategies is always possible, at the cost of computational time. The PIP is a powerful tool when such strategies are carefully selected to improve low performing nodes of baseline strategy. In addition to saving computation time, a careful choice allows for analytic bounds to be found. We provide example 3.4.1 to showcase the power of the PIP.

Example 3.4.1. For the game $G(Q, 5, 3)$, on the graph Q as seen in figure 3.4.1, we consider the initial patroller strategy $\boldsymbol{\pi}_0 = \boldsymbol{\pi}^{Q}_{\text{RMFNC}}$, using the minimal full-node cycle $W^{Q}_{\text{MFNC}} = (1, 2, 4, 3, 4, 5, 6, 5, 4, 2)$. The performance of $\boldsymbol{\pi}_0$ at nodes $j \in \{1, ..., 6\}$ is, by equation (3.17),

$$V_{\boldsymbol{\pi}_0, \bullet, j}(Q, 5, 3) = \begin{cases} 0.3 & \text{for } j = 1, 3, 6, \\ 0.5 & \text{for } j = 2, 5, \\ 1 & \text{for } j = 4. \end{cases} \tag{3.20}$$

Figure 3.4.2 illustrates equation (3.20). It is clear that the worst performing nodes are $1, 3$ and 6, so some strategies that perform better at these nodes would be good candidates for improvement.

Consider improving $\boldsymbol{\pi}_0$ by three pure intercepting patroller strategies, $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, $\boldsymbol{\pi}_3$, which simply wait at 1, 3 and 6 for all time, respectively. Let

$$B(i) = \begin{cases} 1 & \text{if } i = 1, \\ 3 & \text{if } i = 2, \\ 6 & \text{if } i = 3, \end{cases}$$

then $\boldsymbol{\pi}_i$ catches all pure attacks $(B(i), \tau)$ for all $\tau \in \mathcal{T}$ and no other pure attacks. So $V_{\boldsymbol{\pi}_i, \bullet, j}(Q, 5, 3) = \mathbb{I}_{\{j = B(i)\}}$ for $i \in \{1, 2, 3\}$ and by using the PIP with the reduction that $p_0 = 1 - p_1 - p_2 - p_3$ we need to solve

$$\begin{aligned} \text{maximize} \quad & \min\Big(f(\boldsymbol{p}) + p_1, g(\boldsymbol{p}), f(\boldsymbol{p}) + p_2, \\ & \qquad h(\boldsymbol{p}), g(\boldsymbol{p}), f(\boldsymbol{p}) + p_3 \Big) \\ \text{s.t} \quad & p_j \in [0,1] \text{ for } j = 1,2,3, \\ & p_1 + p_2 + p_3 \leq 1, \end{aligned} \tag{3.21}$$

in which $\boldsymbol{p} = (p_1, p_2, p_3)$, $f(\boldsymbol{p}) = 0.3(1 - p_1 - p_2 - p_3)$, $g(\boldsymbol{p}) = 0.5(1 - p_1 - p_2 - p_3)$ and $h(\boldsymbol{p}) = (1 - p_1 - p_2 - p_3)$. Solving equation (3.21) we immediately get that $p_1 = p_2 = p_3$ and along with knowing that $h(\boldsymbol{p}) \geq g(\boldsymbol{p})$ for all choices of $\boldsymbol{p}$, we only need to consider solving $f(\boldsymbol{p}) + p_1 = g(\boldsymbol{p})$ giving $p_1 = \frac{1}{8}$. So $\boldsymbol{p} = (\frac{1}{8}, \frac{1}{8}, \frac{1}{8})$ and hence for this hybrid strategy with $p_0 = \frac{5}{8}$ and $p_i = \frac{1}{8}$ for $i \in \{1,2,3\}$ we get a lower bound of $\frac{5}{16}$ (optimal value of equation (3.21)). So using the PIP, along with the chosen strategies we can achieve a strict improvement over the baseline as

$$V_{\boldsymbol{\pi}_{\text{Hybrid}},\bullet}(Q,5,3) = \frac{5}{16} > \frac{3}{10} = V_{\boldsymbol{\pi}_0,\bullet}(Q,5,3).$$

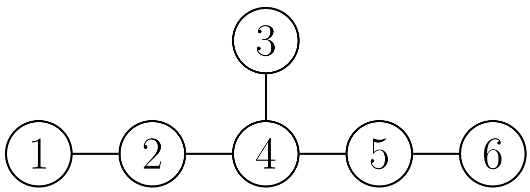


Figure 3.4.1: Graph Q used in example 3.4.1

As previously mentioned, careful choice of the strategies $\boldsymbol{\pi}_i$ for $i = 1, ..., l$ for some $l \in \mathbb{N}$, help to determine if it possible to get a hybrid strategy which is strict improvement. While the choice of patroller strategies available in Π is infinite, we can determine a sufficient condition on when strict improvement is possible over a baseline by considering using pure patrollers which are intercepting.

Lemma 3.4.2. *For the game $G(Q,T,m)$ it is possible to get a hybrid patroller strategy which is a strict improvement over the baseline patroller strategy $\boldsymbol{\pi}_0 \in \Pi(Q,T,m)$ if there exists some $N_I \subset N$ such that:*

- *for all $j, j' \in N_I$,$V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) = V_{\boldsymbol{\pi}_0,\bullet,j'}(Q,T,m)$,*
- *for all $j \in N_I$ and for all $j' \notin N_I$, $V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) < V_{\boldsymbol{\pi}_0,\bullet,j'}(Q,T,m)$ and*
- *for all $j \in N_I$, $V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) < \frac{1}{|N_I|}$.*

The conditions in lemma 3.4.2 correspond to knowing all nodes in the set are currently the minimal performing nodes and that this performance is low enough that it can be improved by choosing $|N_I|$ strategies which for each $j \in N_I$ wait at node j for the entire time-horizon. The construction of such a hybrid strategy and following the PIP provides the proof of the lemma.

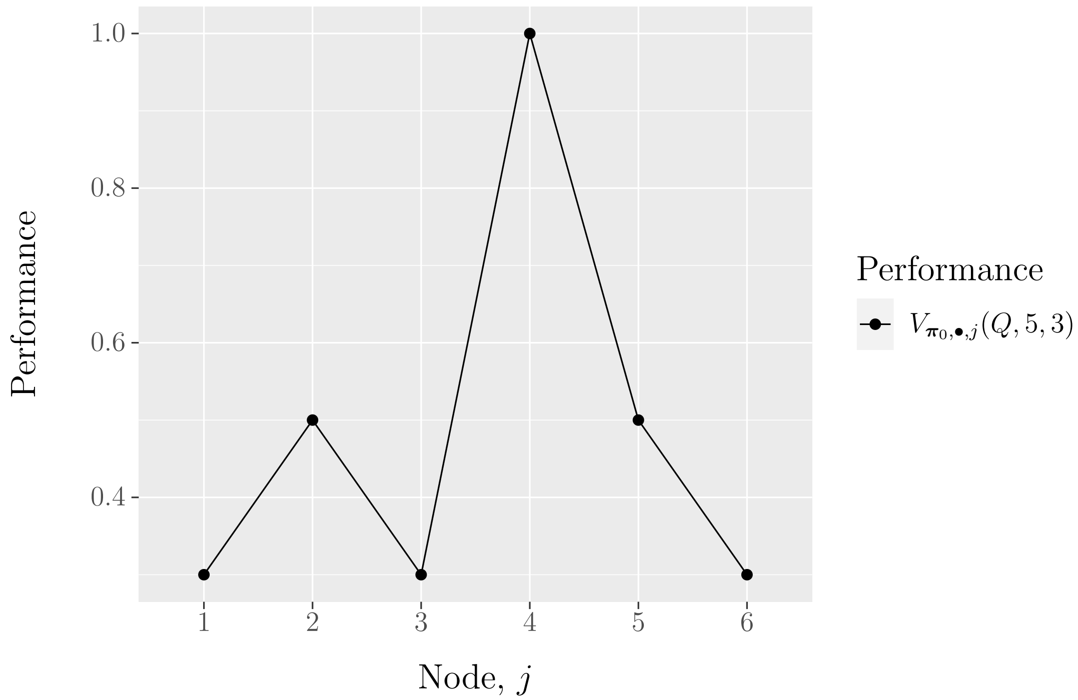


Figure 3.4.2: Performance of $\boldsymbol{\pi}_0$ at nodes as in equation (3.20).

Proof. Let $B : \{1, ..., |N_I|\} \to N_I$ be an bijection, such that $B(i)$ for $i \in \{1, ..., |N_I|\}$ maps to distinct node of N_I. Then let $\boldsymbol{\pi}_i$ be the patroller strategy $W_i(t) = B(i)$ for all $t \in \mathcal{J}$ for each $i \in \{1, ..., |N_I|\}$, then

$$V_{\boldsymbol{\pi}_i,\bullet,j}(Q,T,m) = \mathbb{I}_{\{j=B(i)\}}.$$

Consider improving the baseline $\boldsymbol{\pi}_0$ with $\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_{|N_I|}$ then from the PIP we immediately know that $p_i = p \leq \frac{1}{|N_I|}$ for all $i \in \{1, ..., |N_I|\}$, so we can reduce the PIP to focus on a single node $j \in N_I$ for which we need to solve

$$(1 - |N_I|p) \min_{i \in N \setminus N_I} V_{\boldsymbol{\pi}_0,\bullet,i}(Q,T,m) = (1 - |N_I|p) V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) + p.$$

By the conditions imposed in the lemma it is clear that this reduced PIP has a non-zero optimal solution, namely

$$p = \frac{\min_{i \in N \setminus N_I} V_{\boldsymbol{\pi}_0,\bullet,i}(Q,T,m) - V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m)}{|N_I| \left(\min_{i \in N \setminus N_I} V_{\boldsymbol{\pi}_0,\bullet,i}(Q,T,m) - V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) \right) + 1},$$

and an optimal value of

$$\frac{\min_{i \in N \setminus N_I} V_{\boldsymbol{\pi}_0,\bullet,i}(Q,T,m) - V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) + 1}{|N_I| \left(\min_{i \in N \setminus N_I} V_{\boldsymbol{\pi}_0,\bullet,i}(Q,T,m) - V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m) \right) + 1} > V_{\boldsymbol{\pi}_0,\bullet,j}(Q,T,m).$$

Hence as the performance at node j was minimal among all nodes in the strategy $\boldsymbol{\pi}_0$, we have constructed a hybrid strategy which is a strict improvement over $\boldsymbol{\pi}_0$. $\square$

Following from the proof of lemma 3.4.2, we note that we do not necessarily require the use of $|N_I|$ improvement strategies. We can instead use the minimal number of intercepting patrols which between them contain all nodes in N_I. Therefore it is possible the conditions in the lemma can be relaxed. Furthermore, other alternative strategies which have a higher performances at the nodes in N_I could be used instead of intercepting patrols. However, as we do not require the use of such sufficient conditions in the remainder of this thesis we leave such sufficient conditions for future work.

3.5 k-partite graphs

In this section we showcase the power of our contributions to techniques and strategies shown throughout this chapter. We find solutions to patrolling games on some bipartite graphs which are not complete. Further we provide the value of the game for any complete k-partite graphs, discussing when certain edges from such a complete graph can be removed without changing the found value.

By lemma 2.3.28 we know that for any complete bipartite graph $K_{a,b} \in \mathcal{KP}_2$, with $|A| = a \leq b = |B|$, the value of the game is given by

$$V(K_{a,b}, T, m) = \frac{m}{2b},$$

with the lower bound derived from the complete bipartite graph $K_{b,b}$ which is Hamiltonian. Therefore the strategy used in the game $G(K_{b,b}, T, m)$ is embedded into the game $G(K_{a,b}, T, m)$ to form a random minimal full-node cycle strategy $\boldsymbol{\pi}^{K_{a,b}}_{\mathrm{RMFNC}}$. The optimal attacker strategy created in [16] for use in any bipartite graphs (not necessarily complete) was such that the probability of choosing $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2b} & \text{if } j \in B, \tau \in \{0, 1\}, \\ 0 & \text{otherwise.} \end{cases}$$

We remark that this optimal attacker strategy, created just for bipartite graphs, is equivalent to a 2-polygonal attack $\boldsymbol{\phi}_{\mathrm{poly}}$, using B. An intuitive reason that such a strategy is optimal is that the set A is seen as a stepping stone, that the patroller must pass through to get between nodes in B.

The upper bound given by choosing such an attacker strategy was given in [16] as $V(Q, T, m) \leq \frac{m}{2b}$ for any bipartite graph $Q \in \mathcal{P}_2$ with sets A, B such that $|A| = a \leq b = |B|$. While only giving value to complete bipartite graphs, the work done in [16], does not explore the idea for optimality in non-complete bipartite graphs. This can be done by considering the length of a minimal full-node cycle, as if Q has a minimal full-node cycle W^Q_{MFNC} of length $2b$ we get a lower bound of $V(Q, T, m) \geq \frac{m}{2b}$ and hence the value of the game. Therefore as a corollary of theorem 3.3.26 we have the following.

Corollary 3.5.1. *For the game $G(Q, T, m)$ where Q is any bipartite graph (with A, B such that $|A| = a \leq b = |B|$) which has a minimal full-node cycle of length $2b$, for all $m \geq 2$ and for all $T \geq m + 1$ we have*

$$V(Q, T, m) = \frac{m}{2b},$$

achieved by choosing a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}^{Q}_{RMFNC}$ and 2-polygonal attacker strategy $\boldsymbol{\phi}_{poly}$, using B.

The proof of corollary 3.5.1 follows immediately from theorem 3.3.26 for the lower bound and lemma 3.3.16 for the upper bound.

While corollary 3.5.1 solves some non-complete bipartite graphs it is not applicable to all that require some repetition of nodes in the larger set B during a minimal full-node cycle. For example consider L_n for $n \geq 5$, then the minimal full-node cycle repeats the following number of nodes in B,

$$\begin{cases} \frac{n-3}{2} & \text{if } n \text{ odd}, \\ \frac{n-2}{2} & \text{if } n \text{ even}, \end{cases}$$

meaning it is impossible to apply corollary 3.5.1 and get the value. We provide an example of a graph matching the condition for the length of the minimal full-node cycle in example 3.5.2.

Example 3.5.2. Consider the graph Q in figure 3.5.1 we note that $Q \in \mathcal{P}_2 \setminus \mathcal{KP}_2$ as many edges are missing compared to if it was the complete version $K_{3,6}$. For example, (b_1, a_3) is absent from Q. Regardless of these missing edges the value of the game remains unchanged in comparison to that of the game on $K_{3,6}$ as

$$W^{Q}_{\text{MFNC}} = (b_1, a_1, b_2, a_1, b_3, a_2, b_4, a_3, b_5, a_3, b_6, a_1)$$

is a minimal full node cycle of length $12 = 2 \times 6$ satisfying the conditions of corollary 3.5.1. Thus for all $m \geq 2$ and for all $T \geq m + 1$ we have

$$V(Q, T, m) = \frac{m}{12}.$$

However if we remove the edge (b_6, a_3) from Q to form Q' we can no longer say that the value remains unchanged when compared to the game on $K_{3,6}$, as the length of a minimal full-node cycle in Q' increases to $14 \neq 2 \times 6$.

In addition to being able to repeatedly remove some edges from $Q \in \mathcal{KP}_2$, as long as it does not effect the length of the minimal full-node cycle, we can consider the addition of edges to a Q. Any edge (j, j') with $j, j' \in A$ may be added without affecting the value of the game. This is because adding edges can only increase the value (see equation (2.18)), but the upper bound still holds by lemma 3.3.18 for the 2-polygonal attack strategy $\boldsymbol{\phi}_{\text{poly}}$, using B.

We now turn our attention to solving the game $G(Q, T, m)$, where $Q = Gr(n_1, n_2)$ is the *grid graph* with parameters $n_2, n_1 \in \mathbb{N}$ such that $n_1 \geq n_2 \geq 2$. That is we

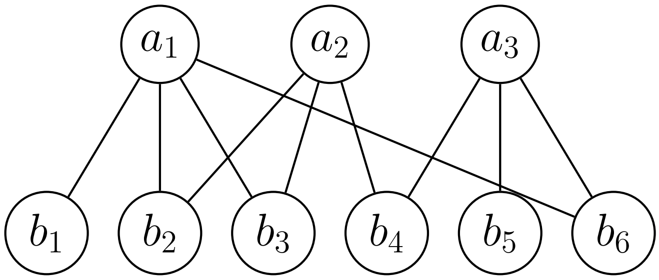


Figure 3.5.1: An incomplete bipartite Q using in example 3.5.2.

look at the patrolling game played on a grid of $n_1 \times n_2$ nodes each adjacent to its orthogonal neighbouring grid points. The node set of $Gr(n_1, n_2)$ is given by

$$N = \{(i, j) | 1 \leq i \leq n_1, 1 \leq j \leq n_2\}$$

and the edge set is given by

$$E = \{((i_1, j_1), (i_2, j_2)) | 1 \leq i_1 \leq n - 1, 1 \leq j_1 \leq n - 1, i_2 = i_1 + 1 \text{ or } j_2 = j_1 + 1\}.$$

We assume that $n_1, n_2 \geq 2$ so that the game is not simply a line graph which has already been solved and that $n_1 \geq n_2$ by noticing that the grid can be transposed if $n_2 > n_1$. For any $n_1 \geq n_2 \geq 2$ the grid graph $Gr(n_1, n_2)$ is bipartite, with sets $A = \{(i, j) \in N \mid i + j = 1 \mod 2\}$ and $B = \{(i, j) \in N \mid i + j = 0 \mod 2\}$ and so

$$\begin{cases} |A| = |B| = \frac{n_1 n_2}{2} & \text{if } n_1 \text{ or } n_2 \text{ are even,} \\ |A| = \frac{n_1 n_2 - 1}{2}, |B| = \frac{n_1 n_2 + 1}{2} & \text{if } n_1 \text{ and } n_2 \text{ are odd.} \end{cases} \tag{3.22}$$

Furthermore the grid graph $Gr(n_1, n_2)$ is only a complete bipartite graph if $n_1 = n_2 = 2$ and in this case $Gr(2, 2) \equiv K_{2,2}$ under a relabelling (isomorphism). By lemma 3.3.16, the 2-polygonal attacker strategy $\boldsymbol{\phi}_{\text{poly}}$, using B gives us, for any $n_1 \geq n_2 \geq 2$, for all $m \geq 1$ and for all $T \geq m$, an upper bound of

$$V(Gr(n_1, n_2), T, m) \leq \begin{cases} \frac{m}{n_1 n_2} & \text{if } n_1 \text{ or } n_2 \text{ are even,} \\ \frac{m}{n_1 n_2 + 1} & \text{if } n_1 \text{ and } n_2 \text{ are odd.} \end{cases} \tag{3.23}$$

For an equal lower bound we look to constructing a minimal full-node cycle. A minimal full-node cycle $W_{\text{MFNC}}^{Gr(n_1, n_2)}$ can be constructed in the following way:

1. Starting at node $(1, 1)$, go across the row to $(1, n_2)$.

1a. If n_1 and n_2 are both odd: Zig-zag left and right while going down to (n_1, n_2), going down to $(2, n_2)$, across to $(2, n_2 - 1)$, down to $(3, n_2 - 1)$, across to $(3, n_2)$... going to the node (n_1, n_2). Go left to $(n_1, n_2 - 1)$.

2. Zig-Zag up and down rows until $(1, 1)$ is reached by going left (to $(n_1, n_2 - 2)$ if n_1, n_2 are both odd or $(n_1, n_2 - 1)$ otherwise) , then up the column till the second row (to $(2, n_2 - 2)$ if n_1, n_2 are both odd or $(2, n_2 - 1)$ otherwise), then left (to $(2, n_2 - 3)$ if n_1, n_2 are both odd or $(2, n_2 - 2)$ otherwise), then down the column till the final row (to $(n_1, n_2 - 3)$ if n_1, n_2 are both odd or $(n_1, n_2 - 2)$ otherwise), repeating until $(1, 1)$ is reached.

An example of this construction of such for $Gr(5, 7)$ and $G(6, 7)$ can be seen in figure 3.5.2. Notice that the initial zig-zagging at the bottom of the grid graph is not required unless both n_1 and n_2 are odd (in which step $1a$. needs to be followed), and in this case one edge (and one node) is repeated. This zig-zagging is done to remove a row, and revert the construction to the other case, at the cost of one repeat.

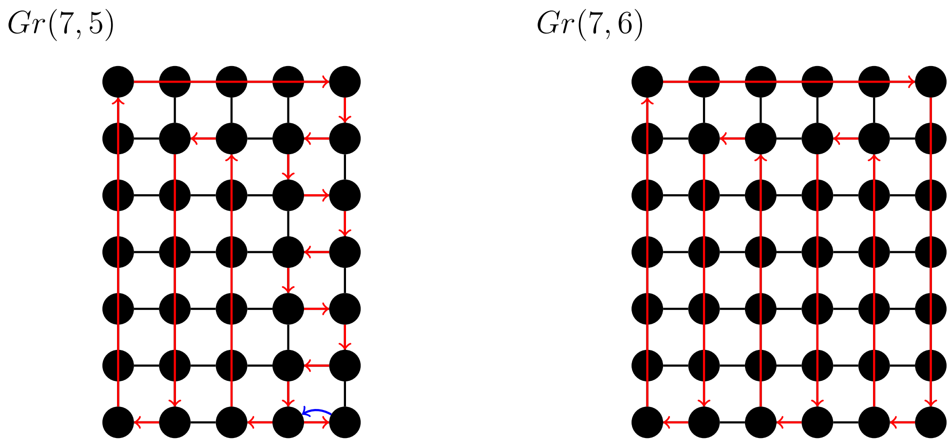


Figure 3.5.2: Two grid graphs, $Gr(7, 5)$ and $Gr(7, 6)$, each showing a constructed minimal full-node cycle in red, aside form the repeated edge for $Gr(7, 5)$ which is highlighted in blue.

From the construction of such a minimal full-node cycle we see that $Gr(n_1, n_2)$ is Hamiltonian when n_1 or n_2 are both even, as the length of the minimal full-node cycle is n_1n_2. On the other hand if either n_1 or n_2 is odd, then the length of a minimal full-node cycle is n_1n_2+1, due to the single repetition (in our construction this repetition is of node $(n_1, n_2 - 1)$). Thus we obtain the following lower bound by theorem 3.3.26: for all $n_1 \geq n_2 \geq 2$, for all $m \geq 2$ and for all $T \geq m$ we have

$$V(Gr(n_1, n_2), T, m) \geq \begin{cases} \frac{m}{n_1n_2} & \text{if } n_1 \text{ or } n_2 \text{ are even,} \\ \frac{m}{n_1n_2+1} & \text{if } n_1 \text{ and } n_2 \text{ are odd.} \end{cases} \tag{3.24}$$

Equations (3.23) and (3.24) therefore prove the following lemma.

Corollary 3.5.3. *For the game $G(Gr(n_1, n_2), T, m)$, for any $n_1, n_2 \in \mathbb{N} \setminus \{1\}$, for all $m \geq 2$ and for all $T \geq m + 1$, we have*

$$V(Gr(n_1, n_2), T, m) = \begin{cases} \frac{m}{n_1n_2} & \textit{if } n_1 \textit{ or } n_2 \textit{ are even,} \\ \frac{m}{n_1n_2+1} & \textit{if } n_1 \textit{ and } n_2 \textit{ are odd,} \end{cases} \tag{3.25}$$

achieved by the random minimal full-node patroller strategy $\boldsymbol{\pi}_{RMFNC}^{Gr(n_1,n_2)}$ and a 2-polygonal attack strategy $\boldsymbol{\phi}_{poly}$, using $B = \{(i, j) \in N \mid i + j = 0 \mod 2\}$.

Having seen that for a bipartite graph, the value is the same as the complete bipartite graph, on the same half-sets, if there exists a full-node cycle of length $2b$, we now move on to look at complete k-partite graphs for $k \geq 3$.

Definition 3.5.4. A graph $Q = (N, E)$ is called *k-partite* if there exists sets $A_1, ..., A_k$ such that the following criteria are met

- $\bigcup_{i=1}^{k} A_i = N$,
- $A_i \cap A_{i'} = \emptyset$ for all $i, i' \in \{1, ..., k\}$ such that $i \neq i'$ and
- $(j, j') \notin E$ for all $j, j' \in A_i$ for all $i \in \{1, ..., k\}$.

The set of all k-partite graphs is denoted by $\mathcal{P}_k$. Furthermore we say Q is a complete k-partite if in addition to the above criteria we also have that $(j, j') \in E$ for all $j \in A_i, j' \in A_{i'}$ for all $i, i' \in \{1, ..., k\}$ such that $i \neq i'$. In this case we say the graph is $Q = K_{a_1,...,a_k}$ where $a_1 = |A_1| \leq ... \leq a_k = |A_k|$. The set of all complete k-partite graphs is denoted by $\mathcal{KP}_k$

We first present results for complete tripartite (3-partite) graphs, followed by analogous results on complete k-partite graphs for $k \geq 3$. Let $K_{a,b,c}$ be the complete tripartite graph with node sets A, B and C with $1 \leq |A| = a \leq |B| = b \leq |C| = c$. Unlike bipartite graphs which are never Hamiltonian, it is possible for tripartite graphs to be Hamiltonian, as the graph does not contain only even length cycles. When $K_{a,b,c}$ is Hamiltonian the value of the game is immediately known by lemma 2.3.27. To decide if a complete tripartite graph is Hamiltonian we can use Ore's theorem ([106]) which states that such a graph is Hamiltonian if $a + b \geq c$. However this does not give us a correspondence, we can easily seek the inverse of this statement to get such a correspondence.

Lemma 3.5.5. *A complete tripartite graph, $K_{a,b,c}$ $(1 \leq a \leq b \leq c)$ is Hamiltonian if and only if $a + b \geq c$.*

Lemma 3.5.5 was developed after having found no such correspondence lemma between complete tripartite graphs and the size of the sets. However, as one can see from the following proof of the lemma, it is relatively easy using Ore's theorem. While we assumed this result trivial and thus not found in our initial literature review we later found the result in [87], which finds such a Hamiltonian cycle if $a + b > c$.

Proof. We begin by formally showing that if $a + b \geq c$ then the graph is Hamiltonian using Ore's theorem, which states that "For a graph $Q = (N, E)$, if $d(i) + d(j) \geq |N|$ for all i, j such that $(i, j) \notin E$ then Q is Hamiltonian", where $d(v)$ is the degree of node $v \in N$. So we have the three inequalities to check:

$$2(b + c) \geq |N| = a + b + c,$$
$$2(a + c) \geq |N| = a + b + c,$$
$$2(a + b) \geq |N| = a + b + c.$$

The first two conditions hold as $1 \leq a \leq b \leq c$, hence given that $a+b \geq c$ all conditions hold and the complete tripartite graph is Hamiltonian.

Next we show that if the graph is Hamiltonian then $a+b \geq c$. If the graph is Hamiltonian then it must exhibit a Hamiltonian cycle, which will be of the following form $H = (*_1, j_1, *_2, j_2, , ..., *_c, j_c, \sim_1, \sim_2, , ..., \sim_{|A|+|B|-|C|})$ where $*_i$ for $i \in \{1, ..., |C|\}$ and $\sim_i$ for $i \in \{|1|, ..., |A|+|B|-|C|\}$ are some unique listings of nodes in A or B and $j_i \in C$ for $i \in \{1, ..., |C|\}$. The number of nodes from A and B used in the Hamiltonian are exactly $|A|+|B| = a+b$, as no nodes are repeated. So for the listings of $*_i$ and $\sim_i$, there must be exactly $a+b$ nodes between them, clearly the total number of listing of $*$ and $\sim$ are greater than c by the construction of H and hence $a+b \geq c$. □

The proof also gives us an easily implemented Hamiltonian cycle for any $K_{a,b,c}$ $(1 \leq a \leq b \leq c)$ such that $a+b \geq c$, that being

$$H = (a_1, c_1, ..., a_{|A|}, c_{|A|}, b_1, c_{|A|+1}, ..., b_{|C|-|A|}, c_{|C|}, b_{|C|-|A|+1}, ..., b_{|B|}).$$

Lemma 3.5.5 implies that a complete tripartite graph being Hamiltonian is equivalent to the largest node set having not more than half of the total number of nodes. In the case of $a+b \geq c$ we know that $K_{a,b,c} \in \mathcal{H}$ and so by lemma 2.3.27 the value of the game for all $m \geq 1$ and for all $T \geq m$ is

$$V(K_{a,b,c}, T, m) = \frac{m}{a+b+c}. \tag{3.26}$$

In contrast, if $a+b < c$ then $K_{a,b,c}$ is non-Hamiltonian, however a minimal full-node cycle can always be constructed. To construct a minimal full-node cycle, we can simply repeat nodes in A or B as necessary to visit all nodes in C. Doing this leads to a full-node cycle W^Q_{MFNC} of length $a+b+c+(c-a-b) = 2c$, as $c-a-b$ repeated nodes are necessary. Therefore, by theorem 3.3.26, we get that for all $m \geq 1$ and $T \geq m$ that

$$V(K_{a,b,c}, T, m) \geq \frac{m}{2c}, \tag{3.27}$$

achieved by $\boldsymbol{\pi}^{K_{a,b,c}}_{\text{RMFNC}}$. Lemma 3.3.18 gives for all $m \geq 2$ and $T \geq m+1$ an upper bound of

$$V(K_{a,b,c}, T, m) \leq \frac{m}{2c}, \tag{3.28}$$

achieved by the 2-polygonal attacker strategy $\boldsymbol{\phi}_{\text{poly}}$, using C. The upper and lower bounds in equations (3.27) and (3.28) are equal but require $T \geq m+1$. In the case of m even we can reduce this to $T \geq m$ by choosing $\boldsymbol{\phi}_{\text{u},C}$, a uniform attacker strategy on C, such that the probability of choosing $(j,\tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{c} & \text{if } j \in C, \tau = 0, \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that for even m any pure patroller can only visit $\frac{m}{2}$ nodes in C and hence choosing $\boldsymbol{\phi}_{\mathrm{u},C}$ provides an upper bound of $V(Q,T,m) \leq \frac{m}{2} \times \frac{1}{c} = \frac{m}{2c}$ for any $m \geq 2$ and $T \geq m$. Therefore the value of the game $G(K_{a,b,c},T,m)$ is known for any complete tripartite graph $K_{a,b,c}$, these results are collected in the following lemma.

Lemma 3.5.6. *For the game $G(K_{a,b,c},T,m)$ (with $1 \leq a \leq b \leq c$)*

- *if $a+b \geq c$ then for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(K_{a,b,c},T,m) = \frac{m}{a+b+c},$$

achieved by a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{K_{a,b,c}}$ and the position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$.

- *if $a+b < c$ then for all $m \geq 1$ and for all $T \geq m + \mathbb{I}_{\{m \text{ odd}\}}$ we have*

$$V(K_{a,b,c},T,m) = \frac{m}{2c},$$

achieved by a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{K_{a,b,c}}$ and a 2-polygonal attacker strategy using C if m is odd and a uniform attack $\boldsymbol{\phi}_{u,C}$ if m is even.

Similarly to a complete bipartite $K_{a,b}$ for which stated that the inclusion of edges (j,j') for $j,j' \in A$ do not affect the value of a game, we obtain that for a complete tripartite $K_{a,b,c}$ that inclusion of edges (j,j') such that $j,j' \in A$ or $j,j' \in B$ do not affect the value of a game, because the upper bound still holds. Furthermore edges can be removed if they do not effect the length of the minimal full-node cycle so a full-node cycle remains of length $a+b+c$ if $a+b \geq c$ or $2c$ if $a+b < c$.

All complete k-partite graphs admit the same optimal strategies of the random minimal full-node cycle and either a position uniform attacker strategy or a 2-polygonal attacker strategy using the largest set A_k. We present the analogous results in the following theorem.

Theorem 3.5.7. *For the game $G(K_{a_1,\dots,a_k},T,m)$(with $1 \leq a_1 \leq \dots \leq a_k$)*

- *if $\sum_{i=1}^{k-1} a_i \geq a_k$ then for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(K_{a_1,\dots,a_k},T,m) = \frac{m}{\sum_{i=1}^{k} a_i},$$

achieved by choosing a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{K_{a_1,\dots,a_k}}$ and the position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$.

- *if* $\sum_{i=1}^{k-1} a_i < a_k$ *then for all* $m \geq 1$ *and for all* $T \geq m + \mathbb{I}_{\{m \ odd\}}$ *we have*

$$V(K_{a_1,\dots,a_k}, T, m) = \frac{m}{2a_k},$$

achieved by choosing a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{K_{a_1,\dots,a_k}}$ *and choosing the* 2*-polygonal attack strategy using* A_k *if* m *is odd and the uniform attacker strategy* $\boldsymbol{\phi}_{u,A_k}$ *if* m *is even.*

Furthermore,

- *the addition of an edge* (j, j') *such that* $j, j' \in A_i$ *for some* $i \in \{1, \dots, k-1\}$ *into the graph* $K_{a_1,\dots,a_k}$ *does not affect the above result on the value of the game.*
- *the removal of an edge* $(j, j') \in E$ *such that in either case the length of a minimal full-node cycle remains the same does not affect the above result on the value of the game.*

As the proof of theorem 3.5.7 is analogous to that of the complete tripartite, which has been explained previously, we leave it to appendix A.1. Of particular note from the proof is the construction of minimal full-node cycle for the game, which is constructed by alternating between a nodes in A_k and a node in $N \setminus A_k$ using distinct nodes where possible. This means it is easy to see that the removal of all edges such that (i, i') for $i \in A_j$ and $i' \in A_{j'}$ with $j \neq j'$ and $j \neq k \neq j'$, does not affect the length of a minimal full-node. Essentially, if all such edges are removed the complete bipartite K_{p,a_k} where $p = \sum_{i=1}^{k-1} a_i$ is formed, which omits the same optimal solutions when the patrolling game is played on the graphs.

3.6 Concluding comments

In this chapter we started in section 3.2 by providing our contributions to strategy reduction techniques which included; reducing the set of attacker strategies to one which ensures at there is a non-zero probability of a pure attack with a commencement time $\tau = 0$; reducing the pure walk response set for the patroller given an arbitrary attacker strategy, which included reducing the walks to move, wait form in the set Ω''' by theorem 3.2.13. In doing such a reduction to searching through $\omega \in \Omega'''$ in order to find the maximum payoff against an arbitrary attacker strategy $\boldsymbol{\phi}$ finds $V_{\bullet,\boldsymbol{\phi}}(Q, T, m)$, the performance of $\boldsymbol{\phi}$, and hence finds an upper bound on the value of the game as $V(Q, T, m) \leq V_{\bullet,\boldsymbol{\phi}}(Q, T, m)$. More reductions to Ω''' where shown when the attacker strategy $\boldsymbol{\phi}$ is no longer arbitrary but has some properties, making the process of finding the performance much easier in such cases.

Next, in section 3.3, we produced patroller and attacker strategies which can be chosen to generate lower and upper bounds respectively on $V(Q,T,m)$. In particular we extended the idea of game decomposition into subgraph patrolling games without known values (or optimal patroller strategies) and the idea of node-identification. We then correctly stated the upper bound for the diametric attacker strategy $\boldsymbol{\phi}_{\text{di}}$, correcting the originally stated lemma 2.3.24 (lemma 9 from [16]) with our lemma 3.3.11. Further to this we developed the time-limited attacker strategy $\boldsymbol{\phi}_{\text{tdi}}$ which provides the bound as in the originally stated lemma 2.3.24, the development of which is essential as the bound (and therefore strategy) is used in the solution to the line graph. Hence $\boldsymbol{\phi}_{\text{tdi}}$ must replace previous results which used $\boldsymbol{\phi}_{\text{di}}$ and in doing so must implement a condition on the game length that $T \geq m + \bar{d} - 1$. Finally we developed a 'Hamiltonian like lower' bound by the use of a random minimal full-node cycle $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$, using the minimal full-node cycle W^{Q}_{MFNC}, in theorem 3.3.26 giving us the lower bound

$$V(Q,T,m) \leq \frac{m}{L}$$

where L is the length of W^{Q}_{MFNC}. This lower bound proves extremely important in finding the solution to the elongated star graph in chapter 4 and we conjecture in chapter 5 that for certain conditions on the patrolling game that such a bound is equal to the value of the game. We also provide some analysis of the weakness of using a random minimal full-node cycle $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ by looking at performances at each node, allowing us to locate spatial weaknesses.

We then, in section 3.4, provided the Patrol Improvement Program(PIP) which can maximize the improvement the of a patroller strategy provided some other patroller strategies are carefully chosen. In addition we provided some sufficient conditions to show when improvement was possible and hence when a patroller strategy is not optimal.

To conclude this chapter we used our contributions to techniques and strategies to extend the previous solutions to patrolling games on complete bipartite graphs $K_{a,b}$ to some non-complete bipartite graphs $Q \in \mathcal{P}_2 \setminus \mathcal{KP}_2$ when the length of a minimal full-node cycle on the graph Q remains the same as in the complete bipartite graph. This was then extended to complete k-partite graphs and some non-complete k-partite graphs with an analogous condition on the minimal full-node cycle.

Chapter 4

Patrolling games on extended star graphs

4.1 Chapter introduction

In this chapter we apply our contributions to techniques developed in chapter 3, specifically our results for a random minimal full-node cycle strategy $\boldsymbol{\pi}^{Q}_{\mathrm{RMFNC}}$ in theorem 3.3.26 and our work on the reduction of the best response space to a given attacker strategy as in sections 3.2.2 and 3.2.3, to patrolling games on various extended star graphs. We will look at the following extensions to star graphs, *elongated star* graphs, *generalised star* graphs, *dual star* graphs and *linked-generalised star* graphs.

In section 4.2 we will study the patrolling game on the elongated star graph, denoted S_n^k, which is formed by taking the star graph and performing k node splitting operations on node 1 so that it is at a distance of $k+1$ from the centre. The elongated star graph, S_n^k, merges the structure of a star graph, S_{n-1}, and a line graph, L_{k+2}. Extending the star graph in such a way allows for the patrolling game scenario in which there is a border with a small station at one end, which contains several rooms. As with the solution to the line graph, for the solution on the elongated star graph, we consider the decomposition of the set of possible attack lengths into six regions. These attack length regions depend on both the star and line parameters, n and k respectively. The first region of $m = 1$ is solved for all graphs by lemma 2.3.26 so requires no work. In section 4.2.2 we find that the (random) full-node cycle is optimal for another region. In subsection 4.2.3 we also see that the random full-node cycle is optimal, by finding an optimal attacker strategy covering another region. In section 4.2.4 we see that the region when $m = 2$ is solved by the covering patroller strategy and the independent attacker strategy. For the remaining regions no currently known strategy provides optimality for either player. In section 4.2.5 we look at why the random minimal full-node cycle is not optimal in the remaining regions. In sections 4.2.6 and 4.2.8 we apply the PIP from chapter 3, section 3.4, to find improvements by selecting some additional strategies which perform better at weakly performing nodes. In sections 4.2.7 and 4.2.9 we will see that such improvements to the random minimal full-node cycle strategy are optimal by finding and adapting attacker strategies for the game. However, this is only done when $\rho \equiv m - 2k - 2 \mod 4 \in \{0, 2\}$ and for $\rho \in \{1, 3\}$ we find near optimal attacker strategies. The strategies for each ρ are discussed in section 4.2.10 along with there respective upper bounds such strategies achieve. The work on the elongated star graph is summarized by theorem 4.2.24 in section 4.2.11.

In section 4.3 we follow the work on the elongated star graph, by studying the patrolling game on the generalised star graph, denoted $S_n^{\boldsymbol{k}}$, where each node i in the star graph undergoes k_i node-splittings to be a distance $k_i + 1$ from the centre. Extending S_n^k to $S_n^{\boldsymbol{k}}$ allows us to model the scenario of a central location, with any number of arbitrary length borders which need to be patrolled. The patrolling game on a generalised star graph is one model of the cow path problem (as in [96]), with variable arc lengths in the patrolling game set-up. We are able to achieve optimal strategies in multiple regions of the attack length, remarking that analogous attacker strategies are optimal when the random full-node cycle is optimal. We do not produce a full solution for all attack length regions, due to the complexity of finding bespoke attacker strategies when the random full-node cycle is not optimal. We do however see that the PIP can still be used in order to improve the random full-node cycle strategy.

In section 4.4 we look at linking disconnected star graphs. In particular we start with section 4.4.1, which considers the dual-star graph S_{n_1,n_2}, which connects two disconnected star graphs, S_{n_1} and S_{n_2}, by an edge between their centres. We use decomposition, along with a bespoke attacker strategy to find the optimal solution to patrolling games on the dual-star graph. This is followed by section 4.4.2 in which we provide a partial solution for the patrolling game on a p-linked generalised star graph, $(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)$. $(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)$ is constructed by connecting each of the centres of $S_{n_i}^{\boldsymbol{k}_l}$ by the graph Q_c. In finding this partial solution we further show the idea of superfluous edges (first seen in section 2.3.4 with $V(K_n, T, m) = V(C_n, T, m)$), which do not effect the value of the game when removed.

In chapter 2 we saw that a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{Q}$ is optimal for all complete k-partite graphs for $k \geq 2$, as well as for the line graphs L_n when the attack length $m \geq n - 1$. Throughout this chapter we will see further evidence that $\boldsymbol{\pi}_{\text{RMFNC}}^{Q}$ is the optimal solution for certain regions of the attack length m when the graph Q is a variety of extensions to the star graph. We draw your attention to such results as in chapter 5 we will make conjecture 5.3.2, stating that for all $m \geq |N| - 1$ the random full-node cycle $\boldsymbol{\pi}_{\text{RMFNC}}^{Q}$ is optimal for any tree. Work done in this chapter will provide additional empirical evidence that leads us to believe the conjecture.

4.2 Elongated star graphs

4.2.1 Introduction to the elongated star graph

The elongated star graph S_n^k, for some $n \in \mathbb{N}$ and $k \in \mathbb{N}$, is formed by k node-splittings on node 1, resulting in node 1 being a distance of $k+1$ from the centre. Equivalently S_n^k can be formed by taking the star S_{n-1} and the line L_{k+2} and node identifying one end of the line graph (node $k+2$) to the centre of the star graph (node c). This node identification adds a branch of length $k+1$ to the star graph, the remaining $n-1$ branches being of the usual length, 1. We formally define the elongated star graph and its node labelling below.

Definition 4.2.1. The *elongated star* graph is a graph $S_n^k = (N, E)$ such that

$$N = \{1, 2, ..., k+1\} \cup \{c, *_1, ..., *_{n-1}\}$$

and

$$E = \{(1,2), (2,3), ..., (k, k+1), (k+1, c)\} \cup \{(c, *_1), (c, *_2), ..., (c, *_{n-1})\}.$$

We denote the class of elongated star graphs by $\mathcal{SE}$.

Figure 4.2.1 shows the elongated star graph S_4^5, which can be formed by node-splitting a leaf node 1 five times or by node-identifying 7 from L_7 and c from S_3.

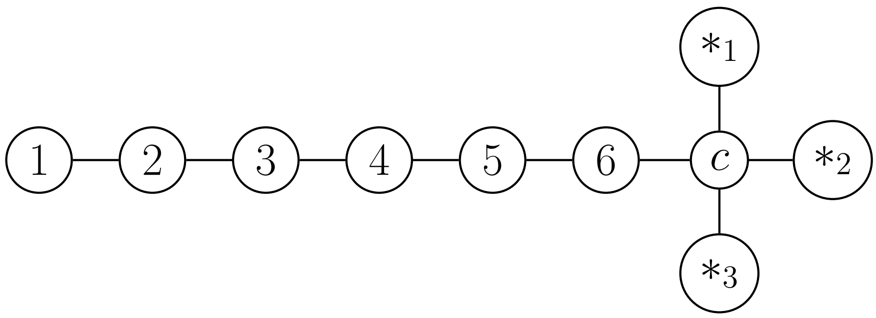


Figure 4.2.1: The elongated star graph $S_4^5 \in \mathcal{SE}$.

We say that S_n^k has n branches, which all start at centre c and end at 1 or $*_i$ for $i = 1, ..., n$, which have branch lengths $d(c, 1) = k+1$ or $d(c, *_i) = 1$ respectively. We refer to n as the number of branches and k as the *branch elongation* parameter. We call nodes 1 and $*_i$ for $i = 1, ..., n$ branch ends.

For S_n^k, we note that if the number of branches n is one or two then the elongated star graph is equivalent to a line graph: $S_1^k \equiv L_{k+2}$ and $S_2^k \equiv L_{k+3}$. Similarly, we note that if the branch elongation k is zero, then the elongated star is equivalent to a star graph: $S_n^0 \equiv S_n$. So for the purposes of our study of the patrolling game on the elongated star graph we assume $n \geq 3$ and $k \geq 1$. We do however note that $n = 1, 2$ and $k = 0$ can be used to compare strategies to those used for the line and star graphs respectively.

As our aim is find the value of the patrolling game $G(S_n^k, T, m)$ for all graphic parameters $n \geq 3$, $k \geq 1$ and all game parameters $m \geq 1$, $T \geq m$, we decompose the set of possible attack lengths $m \in \mathbb{N}$ into six regions, dependent on the number of branches n and the branch elongation k. We consider the following regions for the attack length m:

- $M_0^{S_n^k} = \{m : m = 1\}$,
- $M_1^{S_n^k} = \{m : m \geq 2(n+k)\}$,
- $M_2^{S_n^k} = \{m : 2(k+1) \leq m \leq 2(n+k)\}$,
- $M_3^{S_n^k} = \{m : m = 2\}$,
- $M_4^{S_n^k} = \{m : m > 2n, m < 2(k+1)\}$,
- $M_5^{S_n^k} = \{m : 3 \leq m \leq 2n, m < 2(k+1)\}$.

Note a similarity to that of the decomposition of the set of attack lengths as done for the line graph (see section 2.3.4) with a similar chosen indexing scheme. In our aim to solve the patrolling game $G(S_n^k, T, m)$ we will place some requirements on the game length T dependent on the attacker strategy and how they distribute throughout the commencement time.

We remark that the solution for $m = 1$ ($m \in M_0^{S_n^k}$) is already known as lemma 2.3.26 gives the solution to the game $G(Q, T, 1)$ for all graphs Q and all game lengths $T \geq 1$. Thus, for all $n \geq 3$, for all $k \geq 1$, and for all $T \geq 1$ we have

$$V(S_n^k, T, 1) = \frac{1}{n+k+1},$$

where optimal strategies are the choose and wait patroller strategy $\boldsymbol{\pi}_{\mathrm{cw}}$ and the position-uniform attacker strategy $\boldsymbol{\phi}_{\mathrm{pu}}$.

In section 4.2.2 we provide the solution for the game $G(S_n^k, T, m)$ when $m \in M_1^{S_n^k}$, followed by the solution for the region $m \in M_2^{S_n^k}$ in section 4.2.3. While the game has the same optimal patroller strategy for $m \in M_1^{S_n^k}$ and $m \in M_2^{S_n^k}$, an attacker strategy is developed for the game in which $m \in M_2^{S_n^k}$. In the developed attacker strategy the attacker weights their probability of choosing a branch end such that it is proportional to the distance from the centre. This attacker strategy, called *time-centred* attacker strategy has a higher concentration of potential attacks at the elongated branch end (node 1) compared to other branch ends (nodes $*_i$ for $i = 1, ..., n-1$). This work is followed by the solution to the game when $m \in M_3^{S_n^k}$ ($m = 2$) in section 4.2.4, which uses the independent and covering strategies.

For the final two regions, $M_4^{S_n^k}$ and $M_5^{S_n^k}$, an improvement for the lower bound provided by the patrollers random minimal full-node cycle strategy can be found. The weakness of such a strategy is discussed in section 4.2.5. The PIP from chapter 3, section 3.4, is used in section 4.2.6 for the region $M_4^{S_n^k}$ to find an improvement for the random full-node cycle strategy, leading to a greater lower bound for the game. The lower bound is dependent on the attack length m and therefore prompts a further decomposition of the region $M_4^{S_n^k}$ into two regions $M_{4,0}^{S_n^k}$ and $M_{4,1}^{S_n^k}$. In section 4.2.7, we provide a partial solution for $M_{4,0}^{S_n^k}$, when $n = 3$, by providing an optimal attacker strategy. In addition we provide an example to show that the improvement found in subsection 4.2.6 can be optimal when $n \geq 4$. In section 4.2.8 we again find an improvement over the random full-node cycle strategy when $m \in M_5^{S_n^k}$. Again the improved lower bound found is dependent on m and prompts the further decomposition of $M_5^{S_n^k}$ into two regions $M_{5,0}^{S_n^k}$ and $M_{5,1}^{S_n^k}$. In section 4.2.9, we provide an attacker strategy which is optimal for the region $M_{5,0}^{S_n^k}$. In section 4.2.10 solutions for the region $M_{5,1}^{S_n^k}$ are given by considering simplification. Optimal solutions are given for $\rho \equiv m - 2k - 2 \mod 4 \in \{0, 2\}$ and near optimal solutions are given for $\rho \in \{1, 3\}$, with each attacker strategy requiring small manipulations arising from the case of $\rho = 0$. In order to develop these strategies, it is worth noting the symmetry between the *star nodes*, $*_i$ for all $i \in \{1, ..., n-1\}$. We know that these must be attacked and patrolled in the same way as each other, due to the isomorphism between a relabelling of these star nodes. This helps us more easily consider strategies and how they are improved and adjusted.

We now present details of our results for the game $G(S_n^k, T, m)$ in the various attack length regions, as detailed above.

4.2.2 Solution when $m \in M_1^{S_n^k}$

We begin our solution of the game $G(S_n^k, T, m)$ for $m \in M_1^{S_n^k}$, where

$$M_1^{S_n^k} = \{m : m \geq 2(n+k)\},$$

by finding a minimal full-node cycle for the elongated star graph. It is easy to identify one that starts at node 1, moves to c, and then visits each star node $*_i$ for $i = 1, ..., n-1$, and then returns to 1. That is we define a minimal full-node cycle as

$$W_{\text{MFNC}}^{S_n^k} = (1, 2, ..., k+1, c, *_1, c, *_2, ..., *_{n-1}, c, k+1, ..., 2). \tag{4.1}$$

$W_{\text{MFNC}}^{S_n^k}$ is of length $k+1+2(n-1)+k+1 = 2(n+k)$. Recall that a minimal full-node cycle patroller strategies repeats the cycle as required to fill the time-horizon. By theorem 3.3.26 we can immediately state a lower bound on value of game $G(S_n^k, T, m)$ for $m \in M_1^{S_n^k}$, that being $V(S_n^k, T, m) \geq 1$ and hence along with the trivial upper bound, given by equation (2.11), of $V(S_n^k, T, m) \leq 1$ (for any attacker strategy) we arrive at the value of the game $V(S_n^k, T, m) = 1$. Moreover, the pure strategy $W_{\text{MFNC}}^{S_n^k}$ guarantees a win for the patroller and the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ is not required to achieve the value of the game. Therefore we arrive at the following lemma.

Lemma 4.2.2. *For the game $G(S_n^k, T, m)$ for all $n \geq 3$, for all $k \geq 1$, for all $m \in M_1^{S_n^k}$ and for all $T \geq m$ we have*

$$V(S_n^k, T, m) = 1,$$

achieved by a minimal full-node cycle patroller strategy $W_{MFNC}^{S_n^k}$ and any attacker strategy.

We note that there are multiple minimal full-node cycles which will achieve the value in lemma 4.2.2 and that $W^{S_n^k}_{\text{MFNC}}$ (as in equation (4.1)) is just one such minimal full-node cycle.

4.2.3 Solution when $m \in M_2^{S_n^k}$

We follow the solution when $m \in M_1^{S_n^k}$, by remarking that the minimal full-node cycle $W^{S_n^k}_{\text{MFNC}}$ does not provide optimality when $m \in M_2^{S_n^k}$, where

$$M_2^{S_n^k} = \{m : 2(k+1) \leq m \leq 2(n+k)\},$$

as it can no longer guarantee the capture of all pure attacks. However, we can still use theorem 3.3.26 and the random minimal full-node cycle $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$, using $W^{S_n^k}_{\text{MFNC}}$, in order to provide a lower bound. By theorem 3.3.26, for all $n \geq 3$, for all $k \geq 1$, for all $m \geq 1$ and for all $T \geq m$, we have

$$V(S_n^k, T, m) \geq \frac{m}{2(n+k)}. \tag{4.2}$$

Recall that the randomness in $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ comes from choosing a place within $W^{S_n^k}_{\text{MFNC}}$ uniformly, before following the full-node cycle. An alternative approach to understanding this strategy is to see $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ as the random Hamiltonian $\boldsymbol{\pi}_{\text{rH}}$ in the game $G(C_{2(n+k)}, T, m)$ embedded into the game $G(S_n^k, T, m)$, where S_n^k is expanded to $C_{2(n+k)}$ by repeated node-splitting operations. Figure 4.2.2 illustrates an example of S_3^2 expanded to C_{10}. The nodes repeated in the minimal full-node cycle $W^{S_3^2}_{\text{MFNC}} = (1, 2, 3, c, *_1, c, *_2, c, 3, 2)$ are $2, 3$ and c, which undergo node-splitting. Note that node c is repeated twice in the $W^{S_3^2}_{\text{MFNC}}$ and as a result is node-split twice. We note the unusual labelling of the graph C_{10} is done in order to show the node-splitting of; node 2 into 2_1 and 2_2; node 3 into 3_1 and 3_2; and node c into c_1, c_2 and c_3.

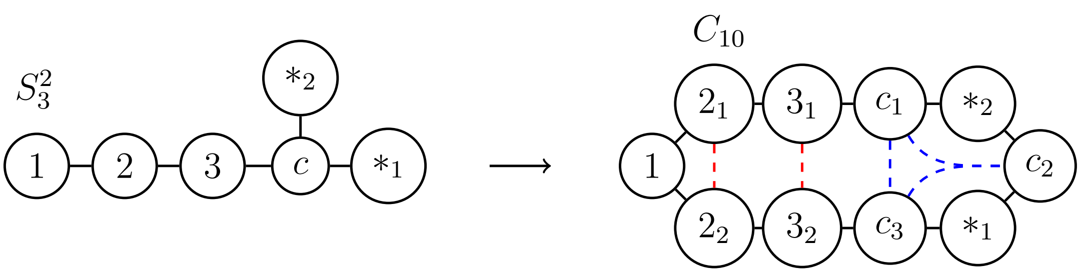


Figure 4.2.2: A Hamiltonian expansion of S_3^2 to C_{10}. Achieved by node-splitting nodes 2,3 once and node c twice, with appropriate edge inheritances chosen. Node 2 is node-split into nodes 2_1 and 2_2, similarly node 3 is node-split into nodes 3_1 and 3_2 and node c is node-split into c_1, c_2 and c_3. The dashed lines help to highlight this correspondence.

To show that the random minimal full-node cycle patroller strategy is optimal, we seek a tight upper bound on the value of the game with the lower bound given in equation (4.2). A simple, but naive, approach for the attacker would be to use the time-limited diametric attacker strategy $\boldsymbol{\pi}_{\text{tdi}}$, as in the solution to the line graph. However for S_n^k, the diameter is $\bar{d} = k + 2$ and thus, by lemma 3.3.13, we have an upper bound of

$$V(S_n^k, T, m) \leq \max\left(\frac{1}{2}, \frac{m}{2(k+2)}\right).$$

So $\boldsymbol{\phi}_{\text{tdi}}$ does not perform optimally (unless $k = 0$). This can be clearly seen in figure 4.2.3, which has a great degree of sub-optimality occurring between the strategies. While from this it is not immediately clear which players strategies are sub-optimal, we later see in lemma 4.2.7 that $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ is optimal for $m \in M_2^{S_n^k}$ and hence we seek to find an attacker strategy which generates a better upper bound. The time-limited diametric attacker strategy seems naive as it only places pure attacks at nodes $*_1$ (without loss of generality) and 1 and therefore does not utilise the symmetry between the star nodes $*_1, ..., *_{n-1}$, which must be attacked with the same probability for the same time in an optimal strategy.

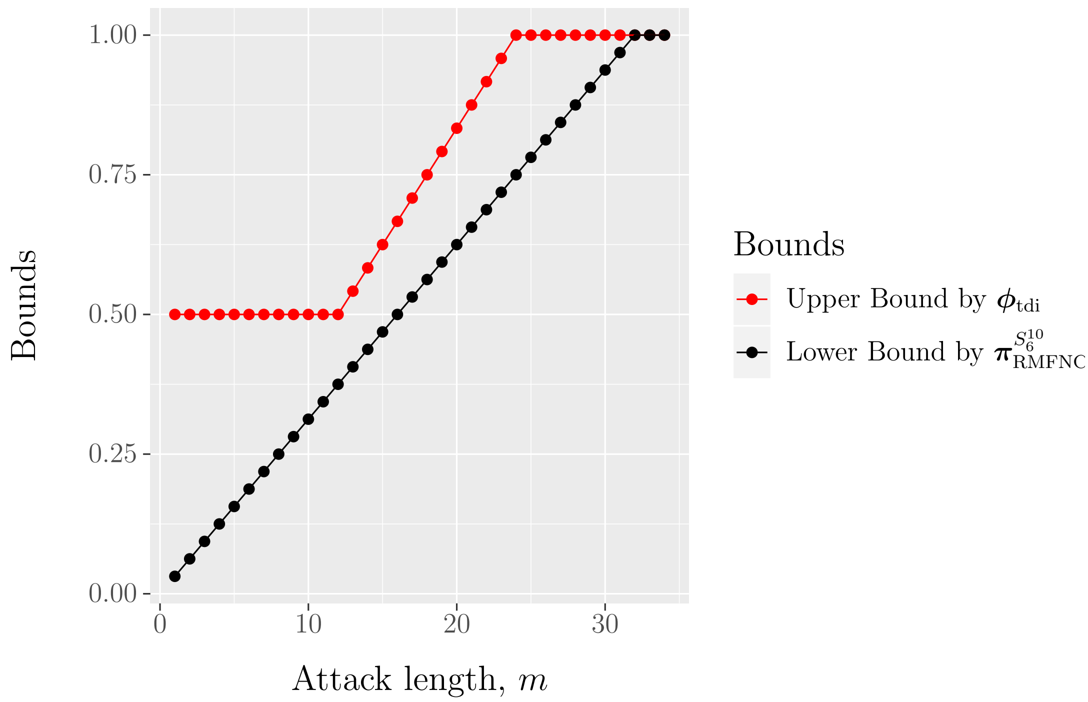


Figure 4.2.3: This graphs shows for the game $G(S_6^{10}, T, m)$, the lower bound from the random minimal full-node cycle strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_6^{10}}$ in black and the upper bound from the time-limited diametric attacker $\boldsymbol{\phi}_{\text{tdi}}$ in red, for $m = 1, ..., 2(n + k + 1)$ and assuming $T \geq m + k + 1$.

Another approach for the attacker is to use the branch ends (nodes 1 and $*_i$ for all $i \in \{1, ..., n-1\}$) proportional to that node's distance from the centre c. That is, have the attacker choose node 1, $k+1$ times more often than any star node $*_i$ for $i = 1, ..., n-1$.

Definition 4.2.3. The *weighted* attacker strategy $\boldsymbol{\phi}_{\mathrm{W}}$ is such that the probability of choosing to play the pure attack $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{n+k} & \text{if } j = *_i \text{ for some } i \in \{1, ..., n-1\} \text{ and } \tau = 0, \\ \frac{k+1}{n+k} & \text{if } j = 1 \text{ and } \tau = 0, \\ 0 & \text{otherwise.} \end{cases}$$

That is for $\boldsymbol{\phi}_{\mathrm{W}}$ the attacker chooses node 1 with probability $\frac{k+1}{n+k}$ and node $*_i$ with probability $\frac{1}{n+k}$ for $i = 1, ..., n-1$ and commences their attack at time 0. Note that in $\boldsymbol{\phi}_{\mathrm{W}}$ the only potential attacks commence at time 0, which we choose in order to make $\boldsymbol{\phi}_{\mathrm{W}}$ a feasible strategy for the most amount of game lengths T as possible. It is however possible to perform time-shifting (as in section 3.2.1) to have all the pure attacks commencement at time t such that $t + m \leq T$.

An example of the weighted attacker strategy $\boldsymbol{\phi}_{\mathrm{W}}$ used in a game $G(S_n^3, 14, 8)$, on the graph S_n^3 for any $n \geq 3$, can be seen in figure 4.2.4. The figure shows $\boldsymbol{\phi}_{\mathrm{W}}$ as its equivalent space-time agent matrix $\boldsymbol{S}_{\mathrm{W}}^{\mathrm{A}}$ with $*$ representing all star nodes $*_i$ for $i \in \{1, ..., n-1\}$, which are attacked identically. Recall that a space-time agent matrix shows how many agents (and therefore the proportional probability) that an attacker is at a space-time point and for the purpose of the figure the space-time places at which pure attacks may commence are filled grey.

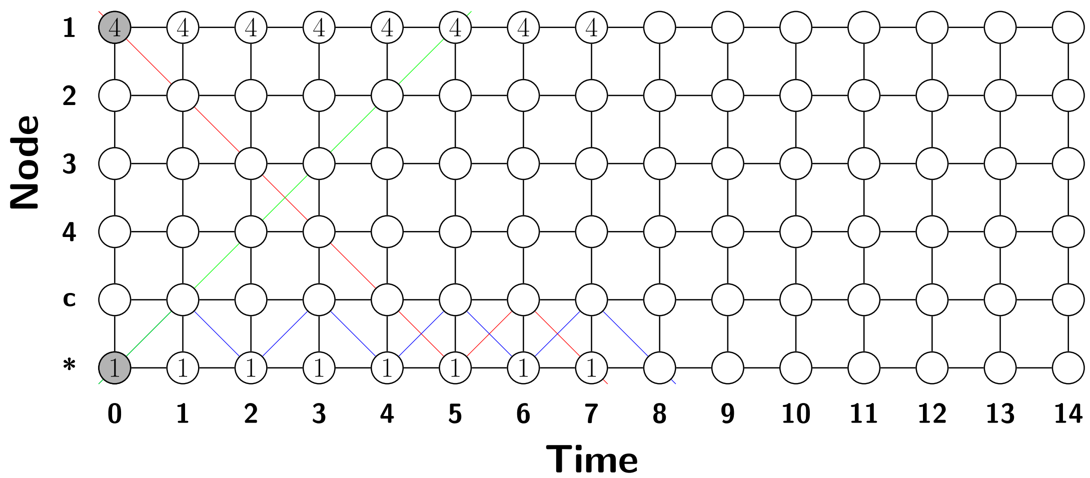


Figure 4.2.4: Space-time agent matrix $\boldsymbol{S}_{\mathrm{W}}^{\mathrm{A}}$ for the weighted attacker strategy $\boldsymbol{\phi}_{\mathrm{W}}$ for the game $G(S_n^3, 14, 8)$ for any $n \geq 3$. Three example pure patrollers are shown in red, green and blue. Catching 6, 5 and 4 of the attackers agents respectively, out of $n + 3$ total agents.

Using our contribution to how a pure patroller best responds to attacker strategies (sections 3.2.2 and 3.2.3), we are able to find the performance, and therefore upper bound, by the attacker using $\boldsymbol{\phi}_{\mathrm{W}}$.

Lemma 4.2.4. *For the game $G(S_n^k, T, m)$ for all $n \geq 3$, for all $k \geq 1$, for all*

$m \geq 1$ and for $T \geq m$ we have

$$V(S_n^k, T, m) \leq \begin{cases} \frac{k+1}{n+k} & \text{for } 1 \leq m \leq k+2, \\ \frac{k+2+\left\lfloor \frac{m-k-3}{2} \right\rfloor}{n+k} & \text{for } k+3 \leq m \leq k+2(n-1), \\ 1 & \text{for } m > k+2(n-1), \end{cases} \tag{4.3}$$

achieved by the attacker using the weighted attacker strategy $\boldsymbol{\phi}_W$.

The proof of lemma 4.2.4 follows by evaluating $V_{\bullet,\boldsymbol{\phi}_\mathrm{W}}(S_n^k, T, m)$, the performance of $\boldsymbol{\phi}_\mathrm{W}$, by relying on the work done in section 3.2.2.

Proof. We aim to calculate $V_{\bullet,\boldsymbol{\phi}_\mathrm{W}}(S_n^k, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet,\boldsymbol{\phi}_\mathrm{W}}(S_n^k, T, m) = V_{\bullet,\boldsymbol{\phi}_\mathrm{W}}(S_n^k, m, m) = \max_{W \in \mathcal{W}(S_n^k, m, m)} P(W, \boldsymbol{\phi}_\mathrm{W}),$$

so we only need to consider pure walks for m units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet,\boldsymbol{\phi}_\mathrm{W}}(S_n^k, m, m) = \max_{\omega \in \Omega'''(S_n^k, m, m)} P(\omega, \boldsymbol{\phi}_\mathrm{W}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_n^k, m, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ such that the following three conditions are met

- $j_i \in N_A \setminus \{j_1, ..., j_{i-1}\}$ for all $i \in \{1, ..., q\}$, where $N_A = \{1\} \cup \{*_l \mid l \in \{1, ..., n-1\}\}$,
- $\nu_i = 0$ for all $i \in \{1, ..., q\}$ and
- $\nu_1 + \sum_{i=1}^{q-1} (d_{N_A}(j_i, j_{i+1}) + \nu_{i+1}) \equiv t_q + \nu q = m - 1$.

That is a move-wait walk such that nodes belong to those at which non-zero probability pure attacks are placed, with no waiting and that the arrival at the final node plus the final waiting match the end of the time-horizon.

In addition as the star nodes $*_i$ for $i \in \{1, ..., n-1\}$ are symmetric and are such that $\varphi_{*_i,\tau} = \varphi_{*_{i'},\tau}$ for all $\tau \in \mathcal{T}$ and for all $i, i' \in \{1, ..., n-1\}$, we can without loss of generality assume the order of the visit to the star nodes in increasing index order. That is $*_i$ is the i^{th} star node visited by any move-wait walk ω, with the node 1 also possibly being visited. We now consider two possible cases depending if ω has $j_1 = 1$ or $j_1 = *_1$.

For the first case consider ω_1 such that $j_1 = 1$, then $j_i = *_{i-1}$ for $i \in \{2, ..., n\}$ and so the payoff is

$$P(\omega_1, \boldsymbol{\phi}_{\mathrm{W}}) = \frac{k+1}{n+k} + \sum_{i=1}^{n-1} \frac{\mathbb{I}_{\{k+2i \leq m-1\}}}{n+k}$$
$$= \begin{cases} \frac{k+1}{n+k} & \text{for } 1 \leq m \leq k+2, \\ \frac{k+1+1+\left\lfloor \frac{m-1-(k+2)}{2} \right\rfloor}{n+k} & \text{for } k+3 \leq m \leq k+2(n-1), \\ 1 & \text{for } m > k+2(n-1). \end{cases} \tag{4.4}$$

For the second case consider ω_2 such that $j_{r+1} = 1$ for some $r \in \{1, ..., q-1\}$ then $j_i = *_i$ for $i \in \{1, ..., r\}$ and $j_i = *_{i-1}$ for $i \in \{r+2, ..., q\}$ and so the payoff is

$$P(\omega_2, \boldsymbol{\phi}_{\mathrm{W}}) = \sum_{i=1}^{r} \frac{\mathbb{I}_{\{2(i-1) \leq m-1\}}}{n+k} + \frac{(k+1)\mathbb{I}_{\{k+2r \leq m-1\}}}{n+k} + \sum_{i=r+1}^{n-1} \frac{\mathbb{I}_{2(k+i) \leq m-1}}{n+k}$$
$$= \begin{cases} \frac{\min(1+\left\lfloor \frac{m-1}{2} \right\rfloor, r)}{n+k} & \text{for } 1 \leq m \leq 2r+k, \\ \frac{r+k+1}{n+k} & \text{for } 2r+k+1 \leq m \leq 2(k+r+1), \\ \gamma & \text{for } 2(k+r+1)+1 \leq m \leq 2(k+n-1), \\ 1 & \text{for } m > 2(k+n-1). \end{cases} \tag{4.5}$$

where

$$\gamma = \frac{r+k+1+\min(1+\left\lfloor \frac{m-1-2(k+r+1)}{2} \right\rfloor, n-1-r)}{n+k}.$$

From equations (4.4) and (4.5) it is clear that $P(\omega_1, \boldsymbol{\phi}_{\mathrm{W}}) \geq P(\omega_2, \boldsymbol{\phi}_{\mathrm{W}})$ for any $r \in \{1, ..., q-1\}$ and hence

$$V_{\bullet, \boldsymbol{\phi}_{\mathrm{W}}}(S_n^k, m, m) = P(\omega_1, \boldsymbol{\phi}_{\mathrm{W}}).$$

Therefore

$$V(S_n^k, T, m) \leq V_{\bullet, \boldsymbol{\phi}_{\mathrm{W}}}(S_n^k, T, m) = \begin{cases} \frac{k+1}{n+k} & \text{for } 1 \leq m \leq k+2, \\ \frac{k+2+\left\lfloor \frac{m-k-3}{2} \right\rfloor}{n+k} & \text{for } k+3 \leq m \leq k+2(n-1), \\ 1 & \text{for } m > k+2(n-1). \end{cases}$$

□

We provide figure 4.2.5 to showcase the sub-optimality of the performance of the weighted attacker strategy $\boldsymbol{\phi}_{\mathrm{W}}$. From the figure we see that when the attack length is such that $k+3 \leq m \leq k+2(n-1)$, the upper bound provided by the weighted attacker strategy and the lower bound by the random minimal full-node cycle patroller strategy are near optimal. It is easy to see from lemmas 4.2.4 and 3.3.13, that $\boldsymbol{\phi}_{\mathrm{W}}$ provides a better upper bound on the value of the game than $\boldsymbol{\phi}_{\mathrm{tdi}}$. Therefore while the weighted attacker strategy is not optimal, the idea of

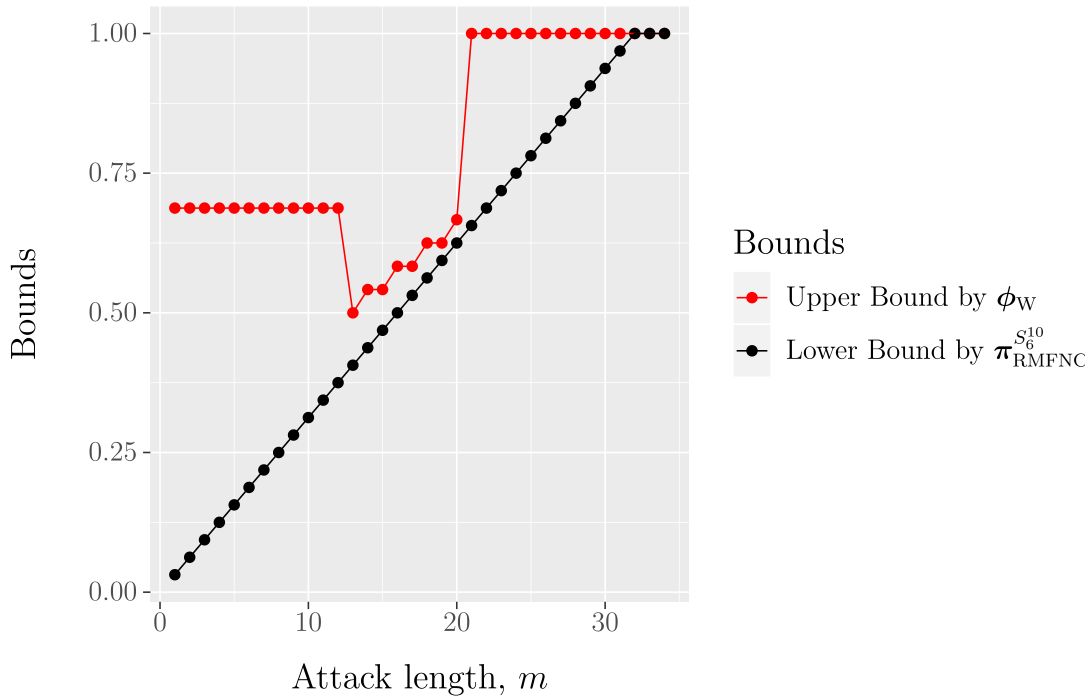


Figure 4.2.5: This graphs shows for the game $G(S_6^{10}, T, m)$, the lower bound from the random minimal full-node cycle strategy $\boldsymbol{\pi}^{S_6^{10}}_{\mathrm{RMFNC}}$ in black and the upper bound from the weighted attacker strategy $\boldsymbol{\phi}_{\mathrm{W}}$ in red, for $m = 1, ..., 2(n + k + 1)$ and for $T \geq m$.

compensating a pure patroller for the distance by having a higher weighting at node 1, is useful.

The weighted attacker strategy $\boldsymbol{\phi}_{\mathrm{W}}$ does not spread out attacks in commencement times, meaning it does not require the best response pure patroller to wait at nodes for any period of time. One way to spread out the weighting may be to equally distribute the probability of choosing the pure attack $(1, 0)$ among the pure attacks $(1, \tau)$ for $\tau = 0, ..., k - 1$. However such a spread, while providing a better bound than the weighted attacker strategy, does not provide a tight upper bound with the lower bound of equation (4.2). The reason that $\boldsymbol{\phi}_{\mathrm{W}}$ does not give a tight upper bound is because it is not symmetric in the commencement time, as the pure attacks played with a non-zero probability all commence at time 0. We achieve an attacker strategy which does provide a tight upper bound by distributing the probability for being at node 1 amongst the commencement time such that each commencement time has a probability equal to that of choosing to attack a star node. Therefore, we could suggest using a strategy such that the probability of playing $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{n+k} & \text{if } j = *_i \text{ for some } i \in \{1, ..., n-1\} \text{ and } \tau = 0, \\ \frac{1}{n+k} & \text{if } j = 1 \text{ and } \tau \in \{0, ..., k\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

However the attacker strategy with probabilities as given in equation (4.6) is not symmetric as pure attacks at node 1 are skewed to start at earlier times. This issue can be fixed by centring the attacker strategy and in doing so we develop the *time-centred* attacker strategy.

Definition 4.2.5. The *time-centred* attacker strategy $\boldsymbol{\phi}_{\text{tc}}$ is such that the probability of playing the pure attack $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2(n+k)} & \text{if } j = *_i \text{ for some } i \in \{1, ..., n-1\} \text{ and } \tau \in \{k, k+1\} \\ \frac{1}{2(n+k)} & \text{if } j = 1 \text{ and } \tau \in \{0, 1, ..., 2k+1\}, \\ 0 & \text{otherwise.} \end{cases}$$

That is in the time-centred attacker strategy the attacker chooses; node 1 with probability $\frac{k+1}{n+k}$ and then, given that they chose node 1, chooses a commencement time $\tau \in \{0, ..., 2k+1\}$ with equal probability; and node $*_i$ with probability $\frac{1}{n+k}$ for each $i \in \{1, ..., n-1\}$ and then, given that they chose node $*_i$, chooses a commencement time $\tau \in \{k, k+1\}$ with equal probability. An example of the time-centred attacker strategy for use on a game with the graph S_n^3, for any $n \geq 3$, can be seen in figure 4.2.6. The strategy is shown as its space-time agent matrix $\boldsymbol{S}_{tc}^A$ in which $2(n+k)$ agents choose a space-time point to commence their attack. In comparison to figure 4.2.4, for $\boldsymbol{\phi}_W$, we see that the time-centred attacker strategy $\boldsymbol{\phi}_{\text{tc}}$, equally distributes its agents who are attacking node 1. In addition, agents at a star nodes $*_i$ for $i \in \{1, ..., n-1\}$, are centred so that attack intervals are symmetric in the time-horizon. Note that the number of agents in $\boldsymbol{S}_{tc}^A$ is double the number in $\boldsymbol{S}_W^A$. While this does not affect probability, it is required for symmetry when m is even. Intuitively not doing so makes the best pure patroller response move between star nodes before moving to finish at node 1, resulting in possibly seeing one more agent than needed for a tight upper bound.

Looking at the three pure patrollers (in red, green and blue) shown in figure 4.2.6, we see that they are each able to catch 8 out of the $2n+6$ potential agents. Note that this number of agents who are caught at 8 for any $n \geq 3$. In addition note that all ‘sensible’ pure patrollers, who move between unseen nodes in which they would arrive in time to see at least one potential agent (arrive before max $\mathcal{T}(j)$ for the node j) see the same number of potential agents. However if $n = 3$ instead of $n \geq 4$, the blue pure patroller is no longer sensible, as they will have already seen all agents at nodes $*_1$, $*_2$ and $*_3$. To correct this the blue pure patroller should move to node 1 at time $t = 7$.

The cost of distributing in commencement time in $\boldsymbol{\phi}_{\text{tc}}$, compared to $\boldsymbol{\phi}_{\text{W}}$, is that the game length T needs to be greater in order to for $\boldsymbol{\phi}_{\text{tc}} \in \Phi(S_n^k, T, m)$. Namely we require $T \geq m + 2k + 1$ for $\boldsymbol{\phi}_{\text{tc}}$ to be feasible in the game $G(S_n^k, T, m)$, whereas $\boldsymbol{\phi}_{\text{W}}$ only required $T \geq m$.

Lemma 4.2.6. *For the game $G(S_n^k, T, m)$ for all $n \geq 3$, for all $k \geq 1$, for all $m \geq 1$ and for all $T \geq 2k + m + 1$ we have*

$$V(S_n^k, T, m) \leq \max\left(\frac{k+1}{n+k}, \frac{m}{2(n+k)}\right),$$

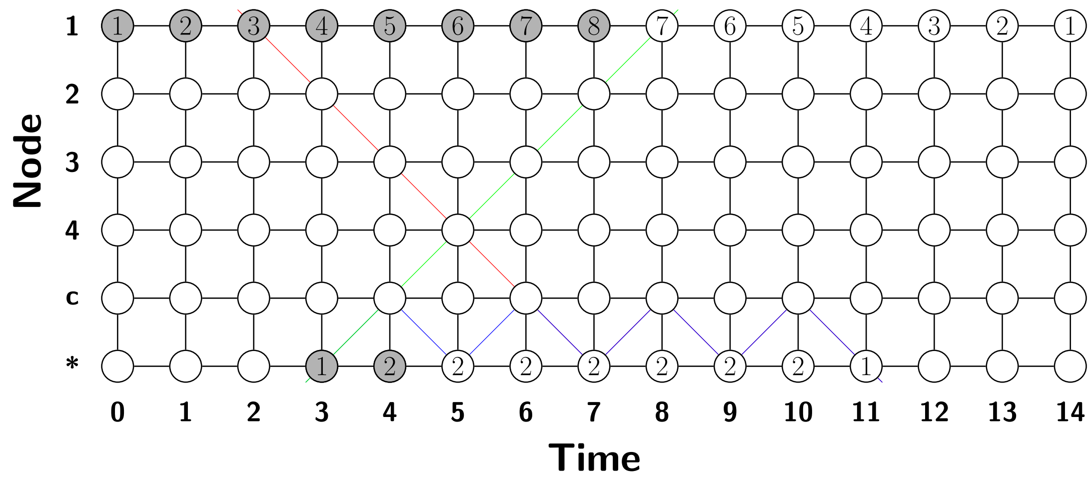


Figure 4.2.6: Space-time agent matrix $\boldsymbol{S}^A_{\mathrm{tc}}$ for the time-centred attacker strategy $\boldsymbol{\phi}_{\mathrm{tc}}$ for the game $G(S^3_n, 14, 8)$ for any $n \geq 4$. Three example pure patrollers are drawn in red, green and blue, all catching 8 agents out of the $2n+6$.

achieved by the attacker using the time-centred attacker strategy $\boldsymbol{\phi}_{tc}$.

The proof of lemma 4.2.6 follows by evaluating $V_{\bullet,\boldsymbol{\phi}_{\mathrm{tc}}}(S^k_n, T, m)$, the performance of $\boldsymbol{\phi}_{\mathrm{tc}}$, by relying on the work done in section 3.2.2. However, unlike the proof of lemma 4.2.4, when we have a distribution in the commencement time, we need to consider waiting at the attacked nodes $N_A = \{1, *_1, ..., *_{n-1}\}$ as there is more than one pure attack with non-zero probability at each node in N_A.

Proof. We aim to calculate $V_{\bullet,\boldsymbol{\phi}_{\mathrm{tc}}}(S^k_n, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet,\boldsymbol{\phi}_{\mathrm{tc}}}(S^k_n, T, m) = V_{\bullet,\boldsymbol{\phi}_{\mathrm{tc}}}(S^k_n, 2k+1+m, m) = \max_{W \in \mathcal{W}(S^k_n, 2k+1+m, m)} P(W, \boldsymbol{\phi}_{\mathrm{tc}}),$$

so we only need to consider pure walks for $2k+1+m$ units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet,\boldsymbol{\phi}_{\mathrm{tc}}}(S^k_n, 2k+1+m, m) = \max_{\omega \in \Omega'''(S^k_n, 2k+1+m, m)} P(\omega, \boldsymbol{\phi}_{\mathrm{tc}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S^k_n, 2k+1+m, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ such that the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$, where $N_A = \{1\} \cup \{*_l \mid l \in \{1, ..., n-1\}\}$,

- $\nu_1 \in \{k, k+1\}$ if $j_1 \in \{*_l \mid l \in \{1, ..., n-1\}\}$, $\nu_1 \in \{0, ..., 2k+1\}$ if $j_1 = 1$, $\nu_i = 0$ for all $i \in \{2, ..., q\}$ and

- $\nu_1 + \sum_{i=1}^{q-1} (d(j_i, j_{i+1}, N_A) + \nu_{i+1}) \equiv t_q + \nu_q = 2k + m.$

That is a move-wait walk such that nodes belong to those which have a non-zero probability of catching the attacker at if travelled to, with no waiting aside from at the initial node and that the arrival at the final node plus the final waiting match the end of the time-horizon.

For any such walk $\omega \in \Omega'''(S_n^{\boldsymbol{k}}, 2k+m+1, m)$ the payoff is given by

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{\text{tc}}) &= \sum_{i=1}^{q} \sum_{n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_i,t} \\ &= \sum_{t=\mathbb{I}_{\{j_1 \neq 1\}}k}^{\nu_1} \varphi_{j_1,t} + \sum_{i=2}^{q} \sum_{t=n_i(\omega)}^{(k+1)+\mathbb{I}_{\{j_i=1\}}k} \varphi_{j_i,t}. \\ &= \sum_{t=\mathbb{I}_{\{j_1 \neq 1\}}k}^{\min(\nu_1,(k+1)+\mathbb{I}_{\{j_i=1\}}k)} \frac{1}{2(n+k)} + \sum_{i=2}^{q} \sum_{t=n'_i(\omega)}^{((k+1)+\mathbb{I}_{\{j_i=1\}}k} \frac{1}{2(n+k)} \\ &\leq \sum_{t=\mathbb{I}_{\{j_1 \neq 1\}}k}^{\min(\nu_1,(k+1)+\mathbb{I}_{\{j_i=1\}}k)} \frac{1}{2(n+k)} + \sum_{i=2}^{q} \sum_{t=n''_i(\omega)}^{(k+1)+\mathbb{I}_{\{j_i=1\}}k} \frac{1}{2(n+k)}. \end{aligned} \tag{4.7}$$

Where $n_i(\omega) = \max(0, l_i(\omega)+1, t_i(\omega)-m+1)$, $n'_i(\omega) = \max(\mathbb{I}_{\{j_i=1\}}k, l_i(\omega)+1, t_i(\omega)-m+1)$ and $n''_i(\omega) = \max(\mathbb{I}_{\{j_i=1\}}k, t_i(\omega)-m+1)$ in equation (4.7). Essentially the inequality follows by ignoring when a node was last visited.

For any $i' \in \{2, ..., q\}$ such that $n''_{i'}(\omega) = t_{i'}(\omega) - m + 1$ we have for all $i \in \{i'+1, ..., q\}$ that $(k+1) + \mathbb{I}_{\{j_i=1\}}k > n''_i(\omega)$ as $t_{i'+1}(\omega) = t_{i'}(\omega) + 2 + \mathbb{I}_{\{j_{i'}=1 \text{ or } j_{i'+1}=1\}}k >$

$(k+1) + \mathbb{I}_{\{j_{i'+1}=1\}}k$. Equation (4.7) therefore becomes

$$\begin{aligned}
P(\omega, \boldsymbol{\phi}_{\mathrm{tc}}) \leq & \sum_{t=\mathbb{I}_{\{j_1\neq 1\}}k}^{\min(\nu_1,(k+1)+\mathbb{I}_{\{j_i=1\}}k)} \frac{1}{2(n+k)} + \sum_{i=2}^{q} \sum_{t=n''_i(\omega)}^{(k+1)+\mathbb{I}_{\{j_i=1\}}k} \frac{1}{2(n+k)} \\
= & \frac{\min(\nu_1+1-\mathbb{I}_{\{j_1\neq 1\}}k, (k+2)+\mathbb{I}_{\{j_1=1\}}k - \mathbb{I}_{\{j_1\neq 1\}}k)}{2(n+k)} + \frac{\sum_{i=2}^{i'-1} 2(1+\mathbb{I}_{\{j_i=1\}}k)}{2(n+k)} \\
& + \frac{\max((k+2)+\mathbb{I}_{\{j_{i'}=1\}}k - t_{i'}(\omega)+m-1, 0)}{2(n+k)} \\
= & \frac{\min(\nu_1+1-\mathbb{I}_{\{j_1\neq 1\}}k, 2(1+\mathbb{I}_{\{j_1=1\}}k))}{2(n+k)} + \frac{\sum_{i=2}^{i'-1} 2(1+\mathbb{I}_{\{j_i=1\}}k)}{2(n+k)} \\
& + \frac{\max((k+2)+\mathbb{I}_{\{j_{i'}=1\}}k - t_{i'}(\omega)+m-1, 0)}{2(n+k)} \\
= & \frac{\min(\nu_1+1-\mathbb{I}_{\{j_1\neq 1\}}k, 2(1+\mathbb{I}_{\{j_1=1\}}k)) + \max(\mathbb{I}_{\{j_1\neq 1\}k-\nu_1+m-1}, 0)}{2(n+k)}
\end{aligned} \tag{4.8}$$

From equation (4.8) it is clear that in order to maximize the payoff for the walk ω it should have $\nu_1 = (k+1) + \mathbb{I}_{\{j_1=1\}}k$ and hence we get that

$$P(\omega, \boldsymbol{\phi}_{\mathrm{tc}}) \leq \frac{\max(2(\mathbb{I}_{\{j_1=1\}}k+1), m)}{2(n+k)}.$$

So it is best for ω to have $j_1 = 1$ and therefore we have

$$V_{\bullet,\boldsymbol{\phi}_{\mathrm{tc}}}(S_n^k, 2k+1+m, m) \leq \max\left(\frac{k+1}{n+k}, \frac{m}{2(n+k)}\right). \tag{4.9}$$

The upper bound on the performance of $\boldsymbol{\phi}_{\mathrm{tc}}$, as in equation (4.9), gives

$$V(S_n^k, T, m) \leq \max\left(\frac{k+1}{n+k}, \frac{m}{2(n+k)}\right).$$

□

From the proof of lemma 4.2.6, we see that there are numerous pure patrollers which are the best response to $\boldsymbol{\phi}_{\mathrm{tc}}$, essentially any pure patrol that catches, on average, one agent per unit of time between times 0 and $m-1$ inclusive is a best response to $\boldsymbol{\phi}_{\mathrm{tc}}$. Together with the lower bound in equation (4.2), lemma 4.2.6 yields the following lemma. The game length T needs to be such that $\boldsymbol{\phi}_{\mathrm{tc}}$ is a feasible attacker strategy, thus $T \geq 2k+m+1$.

Lemma 4.2.7. *For the game $G(S_n^k, T, m)$, for all $n \geq 3$, for all $k \geq 1$, for all $T \geq 2k+m+1$ and for all $m \in M_2^{S_n^k}$ we have*

$$V(Q, T, m) = \frac{m}{2(n+k)},$$

achieved by the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{S_n^k}$ *and the time-centred attacker strategy* $\boldsymbol{\phi}_{tc}$.

We are now left with finding the solution to the game $G(S_n^k, T, m)$ when $1 < m < 2(k+1)$ ($m \in M_3^{S_n^k} \cup M_4^{S_n^k} \cup M_5^{S_n^k}$). We next cover the case of $m = 2$ ($m \in M_3^{S_n^k}$), since this does not require bespoke strategies to be created for the patroller and attacker and we can directly rely on general strategies which have already been developed.

4.2.4 Solution when $m = 2$ ($M_3^{S_n^k}$)

The solution for $m = 2$ ($m \in M_3^{S_n^k}$) follows the same approach as in the the solution to the line graph which relies on the covering patroller strategy and independent attacker strategy and the respective lower and upper bounds given in lemmas 2.3.12 and 2.3.21. In turn these rely on knowing the covering number (and a minimal covering set) and the independence number (and a maximal independent set) and so in this section we aim to find these numbers by constructing such sets and using the lemmas above to arrive at the value of the game $G(S_n^k, T, 2)$.

For the game $G(S_n^k, T, 2)$, for some $n \geq 3$, $k \geq 1$ and $T \geq 2$, we can form a minimal covering set by construction. Recall that for $m = 2$, elements of a covering set are intercepting patrols which are equivalent to edges with the pure patroller oscillating back and forth between the two incident nodes of the edge. Therefore, we choose a set C containing the minimal number of edges such that every node in the graph S_n^k is incident to at least one edge in C.

To construct C take $Q_1 = S_n^k$ and select any edge which has an incident leaf node (node of degree 1), add this edge to C and then delete the edge and the two incident nodes to form the graph Q_2. Repeat this process on the graph Q_2 until for some $l \in \mathbb{N}$ we have that $Q_l = (N_l, E_l)$ is a graph such that the $E_l = \emptyset$. Then for every node $j \in N_l$ add a connected edge from the original graph $Q_1 = S_n^k$. Performing this process leads to

$$C = \{(c, *_i) | i = 1, .., n-1\} \cup \{(k+1, k), (k, k-1), ..., (2, 1)\}.$$

That is C is the set of edges connecting the centre to the star nodes and every alternating edges along the line section. Note if $k = 1$ then $2 \equiv c$. The cardinality of C gives us the covering number for the game $G(S_n^k, T, 2)$ and so

$$\mathcal{C}_{S_n^k, T, 2} = n + \left\lfloor \frac{k}{2} \right\rfloor.$$

Thus, by lemma 2.3.12, we obtain for all $n \geq 3$, for all $k \geq 1$ and for all $T \geq 2$ the lower bound

$$V(S_n^k, T, 2) \geq \frac{1}{n + \lfloor \frac{k}{2} \rfloor}. \tag{4.10}$$

Similarly, we can find the a maximal independent set for the game $G(S_n^k, T, 2)$ and hence find the independence number $\mathcal{L}_{S_n^k,T,2}$. Using the same construction idea as for the set C, we can get nodes a distance of $m = 2$ apart. Starting with all leaf nodes we include these, then delete them and their adjacencies. We can repeat this process, deleting nodes and adjacencies, to form a maximal independent set

$$L = \begin{cases} \{*_1, ..., *_{n-1}, k+1, k-1, ..., 1\} \text{ if } k \text{ is even,} \\ \{*_1, ..., *_{n-1}, k+1, k-1, ..., 2\} \text{ if } k \text{ is odd.} \end{cases}$$

That is L is the set of star nodes and every alternating node along the line section. The cardinality of L gives us the independence number for the game $G(S_n^k, T, 2)$ and so

$$\mathcal{L}_{S_n^k,T,2} = n + \left\lfloor \frac{k}{2} \right\rfloor.$$

Thus, by lemma 2.3.21, we obtain for all $n \geq 3$, for all $k \geq 1$, for all $T \geq 2$ the upper bound of

$$V(S_n^k, T, 2) \leq \frac{1}{n + \left\lfloor \frac{k}{2} \right\rfloor}. \tag{4.11}$$

From equations (4.10) and (4.11) we have tight bounds and hence the following lemma, which gives value of the game $G(S_n^k, T, 2)$.

Lemma 4.2.8. *For the game $G(S_n^k, T, 2)$ for all $n \geq 3$, for all $k \geq 1$ and for all $T \geq 2$ we have*

$$V(S_n^k, T, 2) = \frac{1}{n + \left\lfloor \frac{k}{2} \right\rfloor},$$

achieved by the covering patroller strategy $\boldsymbol{\pi}_{Cov}$ and the independent attacker strategy $\boldsymbol{\phi}_{Ind}$.

Lemma 4.2.8 is proven by knowing that the covering number and independence number are the same. We see in chapter 5, section 5.2, that we can achieve this for any game $G(Q, T, 2)$ where Q is a tree (and $T \geq 2$). However, these numbers are not explicit and are found by an algorithm which simultaneously generates a minimal covering set and a maximal independent set.

It is worth noting that when $m > 2$ the covering number is larger than the independence number as two nodes a distance of m away cannot have an intercepting patrol between them. Therefore covering and independent strategies are not both optimal for $m > 2$. We return soon to considering the remaining attack lengths, $2 < m < 2(k+1)$ ($m \in M_4^{S_n^k}$ and $m \in M_5^{S_n^k}$), in which the random minimal full-node cycle $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ is no longer optimal. Before doing so we will first analyse exactly why the $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ is no longer optimal for these game lengths. This analysis is done by examining the performance at node $j \in N$. We then apply the patrol improvement program(PIP) from chapter 3, section 3.4, in order to find the value of the game, and optimal strategies, when $2 < m < 2(k+1)$.

4.2.5 Weakness of the random minimal full-node cycle

In this section we find which nodes $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ performs weakly at, which depends on the attack length m and hence explain a further decomposition of the attack length region $2 < m < 2(k+1)$ for which we later find the value of the game $G(S_n^k, T, m)$.

For the random minimal full-node cycle strategy $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ we have, by equation (3.17), that its performance at node $j \in N$ for all $n \geq 3$, for all $k \geq 1$, for all $m \geq 1$ and for all $T \geq m$ is given by

$$V_{\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}},\bullet,j}(S_n^k, T, m) = \begin{cases} \frac{\min(m+2(j-1),2m)}{2(n+k)}, & \text{for } j \leq \frac{n+k}{2}+1, \\ \frac{\min(m+2(n+k+1-j),2m)}{2(n+k)}, & \text{for } j > \frac{n+k}{2}+1, \\ \frac{\min(m+2(n-1),nm)}{2(n+k)}, & \text{for } j = c, \\ \frac{m}{2(n+k)}, & \text{for } j \in \{*_1, ..., *_{n-1}\}. \end{cases} \tag{4.12}$$

It is clear from equation (4.12) that the lower bound from using the strategy $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ (equation 4.2) arises from the nodes which have minimal performance. Nodes 1 and $*_i$ for all $i \in \{1, ..., n-1\}$ have the minimal performance among all nodes as they are visited only once per repetition of $W^{S_n^k}_{\text{MFNC}}$. This can be seen in figures 4.2.7 and 4.2.8 which both have these nodes as the lowest performance. We have provided two figures as the features of the performance depends on whether $m \in M_4^{S_n^k}$ or $m \in M_5^{S_n^k}$. Notice that in figure 4.2.7, when $m \in M_5^{S_n^k}$, that the performance in the line section (nodes $1, ..., k+1$) is increasing and then constant. On the other hand, in figure 4.2.8, when $m \in M_4^{S_n^k}$, increases, then is constant and then decreases for nodes in the line section.

From equation (4.12) we can see that the performance in the line section is non-decreasing if,

$$\frac{\min(m+2n, 2m)}{2(n+k)} = \frac{m+n}{2(n+k)} \iff m > 2n.$$

This fact is the reason the remaining attack lengths for which the game $G(S_n^k, T, m)$ is not solved, $2 < m < 2(k+1)$, is divided into the regions $m \in M_4^{S_n^k}$ and $m \in M_5^{S_n^k}$. Knowing this we must look at two separate improvements we can now use the patrol improvement program(PIP) to find a patroller strategy that performs better than $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$. This will be done by the use of carefully chosen patroller strategies that perform better than $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ at nodes which it performs weakly at, which differ between the two regions $m \in M_4^{S_n^k}$ and $m \in M_5^{S_n^k}$.

4.2.6 PIP when $m \in M_4^{S_n^k}$

In this section we use the patrol improvement program (PIP) from chapter 3, section 3.4, to improve the random minimal full-node cycle patroller strategy

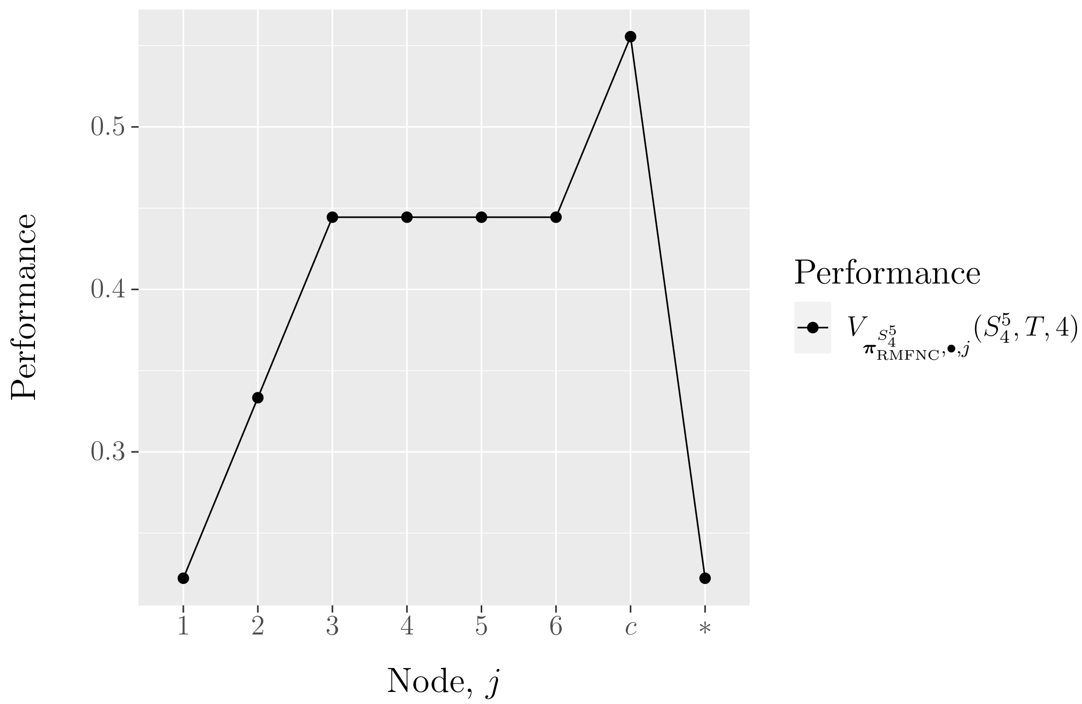


Figure 4.2.7: This graph shows the performance of the random minimal full-node cycle $\boldsymbol{\pi}^{S_4^5}_{\text{RMFNC}}$ at each node for the game $G(S_4^5, T, 4)$ for all $T \geq 4$. Note $m \in M_5^{S_n^k}$.

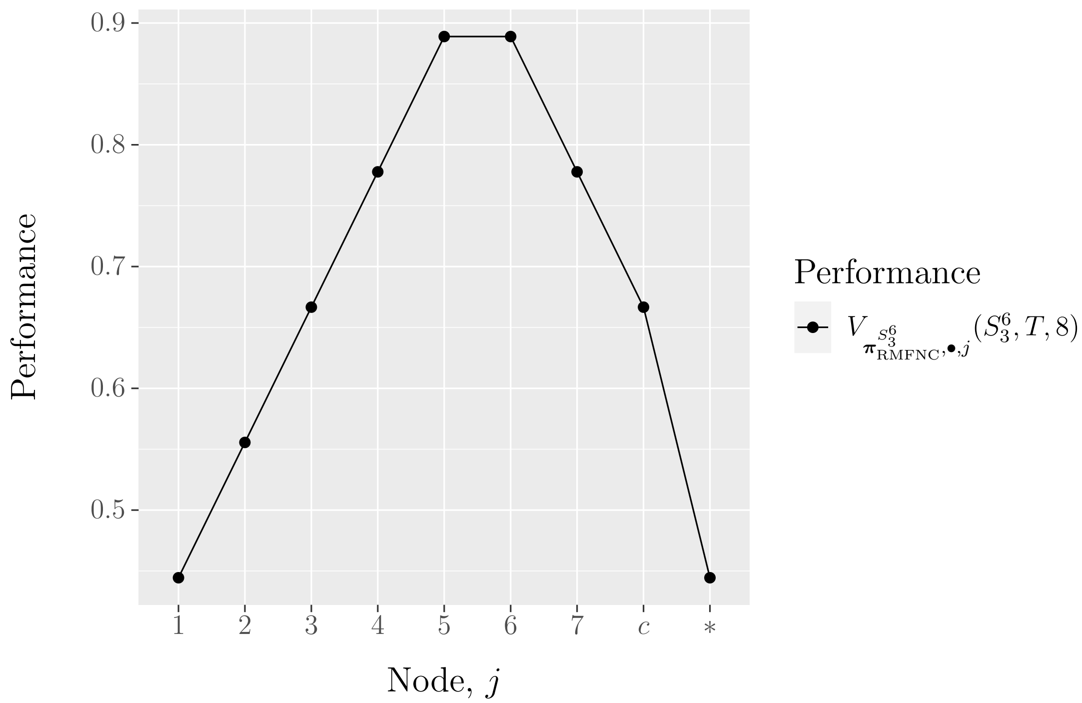


Figure 4.2.8: This graph shows the performance at each node of the random minimal full-node cycle $\boldsymbol{\pi}^{S_3^6}_{\text{RMFNC}}$ at each node for the game $G(S_3^6, T, 8)$ for all $T \geq 8$. Note $m \in M_4^{S_n^k}$.

$\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ for the game $G(S_n^k, T, m)$ when $m \in M_4^{S_n^k}$, where

$$M_4^{S_n^k} = \{m : m > 2n, m < 2(k+1)\}.$$

Recall, that when $m \in M_4^{S_n^k}$, we have $m > 2n$ so the performance of $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ at the nodes in the line section decreases towards the end (node $k+1$). Let $\boldsymbol{\pi}_0 = \boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ be the baseline strategy for the PIP and seek to find strategies which perform better at weakly performing nodes.

We can find a strategy that can be used for the weakly performing left nodes, $L = \{1, ..., \hat{m}+1\}$, where $\hat{m} = \left\lfloor \frac{m}{2} \right\rfloor$. This is done by forming the walk

$$W_{\text{L}} = (1, 2, ..., \hat{m}+1, \hat{m}, ..., 2),$$

which is repeated for the time-horizon. W_{L} is of length $2\hat{m}$ and as $2\hat{m} \leq m$ the walk is intercepting as it visits every node at most m time units apart. The pure strategy W_{L} will used as a candidate for improvement in the PIP, for notational convince we will denote the strategy in mixed form $\boldsymbol{\pi}_1 = W_{\text{L}}$. Then the performance at node $j \in N$ is

$$V_{\boldsymbol{\pi}_1, \bullet, j}(S_n^k, T, m) = \mathbb{I}_{\{j \in L\}}. \tag{4.13}$$

We can also find a strategy that can be used for the weakly performing right nodes, $R = \{k+n+1-\hat{m}, ..., k+1, c, *_1, ..., *_{n-1}\}$. We note that the performance of $\boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}}$ starts to decrease at node $k+1-(\left\lfloor \frac{m}{2} \right\rfloor - 2n) = k+n+1-\hat{m}$. This is done by forming the walk

$$W_{\text{R}} = (k+n-1-\hat{m}, ..., k+1, c, *_1, c, ..., *_n, c, k+1, ..., k+n-\hat{m}),$$

which is repeated for the time-horizon. W_{R} is of length $2\hat{m}$ and as $2\hat{m} \leq m$ (as $m \geq 2n$) the walk is intercepting as it visits every node at most m time units apart. The pure strategy W_{R} will used as a candidate for improvement in the PIP, for notational convince we will denote the strategy in mixed form $\boldsymbol{\pi}_2 = W_{\text{R}}$. Then the performance at node $j \in N$ is

$$V_{\boldsymbol{\pi}_1, \bullet, j}(S_n^k, T, m) = \mathbb{I}_{\{j \in R\}}. \tag{4.14}$$

Having got three patroller strategies we can form the simple hybrid strategy $\boldsymbol{\pi}_{\text{SimpHyb}}$ which plays $\boldsymbol{\pi}_i$ with probability p_i for $i = 0, 1, 2$ and use the PIP to determine the best probabilities which achieve the best improvement over the baseline strategy $\boldsymbol{\pi}_0 (= \boldsymbol{\pi}^{S_n^k}_{\text{RMFNC}})$. However before immediately using the PIP with our left and right improvements, we must first consider if any nodes are not contained within either improvement. That is if the set of middle nodes $M = N \setminus (L \cup R)$ is empty or not. Figure 4.2.9 shows an example of an elongated star graph game in which left and right improvements are used and contain the entire node set, so $M = \emptyset$. Figure 4.2.10 shows the related game in which the branch elongation k has been increased from 6 to 7. Note that in figure 4.2.10 the left and right improvements no longer contain the entire node set, so $M \neq \emptyset$.

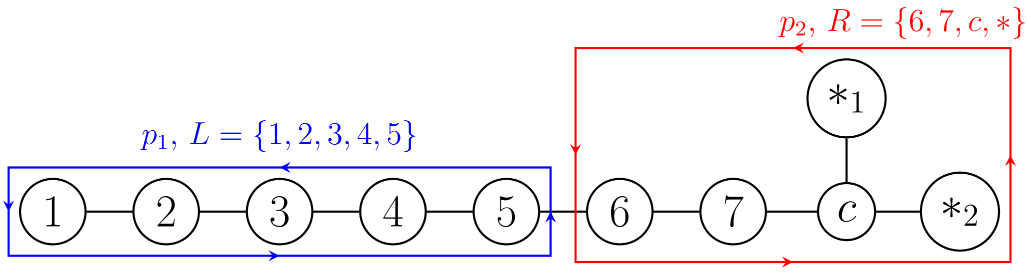


Figure 4.2.9: The game $G(S_3^6, 20, 8)$ with the left improvement, $\boldsymbol{\pi}_1$ played with probability p_1, shown in blue and the right improvement, $\boldsymbol{\pi}_2$ played with probability p_2, shown in red. The sets are $L = \{1, 2, 3, 4, 5\}$ and $R = \{6, 7, c, *_1, *_2\}$ ($M = \emptyset$).

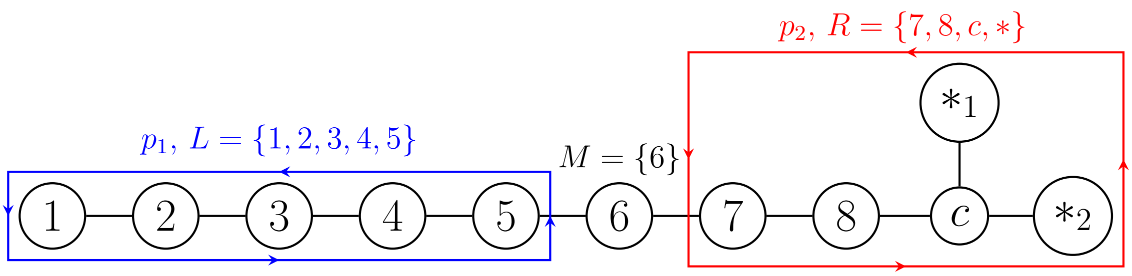


Figure 4.2.10: The game $G(S_3^7, 20, 8)$ with the left improvement, $\boldsymbol{\pi}_1$ played with probability p_1, shown in blue and the right improvement, $\boldsymbol{\pi}_2$ played with probability p_2, shown in red. The sets are $L = \{1, 2, 3, 4, 5\}$, $R = \{7, 8, c, *_1, *_2\}$ and $M = \{6\}$.

By considering the node $\hat{m}+1$ and $k+n+1-\hat{m}$, we have that

$$M = \emptyset \iff \hat{m} \geq \frac{n+k-1}{2}.$$

We use this condition to further decompose $M_4^{S_n^k}$ into

$$M_{4,0}^{S_n^k} = M_4^{S_n^k} \cap \left\{ m \mid \hat{m} \geq \frac{n+k-1}{2} \right\}$$

and

$$M_{4,1}^{S_n^k} = M_4^{S_n^k} \cap \left\{ m \mid \hat{m} < \frac{n+k-1}{2} \right\}.$$

The reason we must consider if $M = \emptyset$ or $M \neq \emptyset$ is that it affects PIP's optimization of the probabilities p_1 and p_2. The following lemma contains the improved lower bound, over the lower bound achieved by using $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ (as in equation (4.2)), for games when either $m \in M_{4,0}^{S_n^k}$ or $m \in M_{4,1}^{S_n^k}$.

Lemma 4.2.9. *For the game $G(S_n^k, T, m)$ for any $n \geq 3$, for any $k \geq 1$, for any $T \geq m$,*

- *for all $m \in M_{4,0}^{S_n^k}$ we have*

  $$V(S_n^k, T, m) \geq \begin{cases} \frac{1}{2} \text{ for } m < n+k, \\ \frac{m}{2(n+k)} \text{ otherwise,} \end{cases}$$

  achieved by the simple hybrid strategy $\boldsymbol{\pi}_{SimpHyb}$, with $p_1 = p_2 = \frac{1}{2}$ (and $p_0 = 0$), called the simple improvement patroller strategy (with no middle nodes).

- *for all $m \in M_{4,1}^{S_n^k}$ we have*

  $$V(S_n^k, T, m) \geq \frac{m}{m+n+k},$$

  achieved by the simple hybrid patroller strategy $\boldsymbol{\pi}_{SimpHyb}$, with $p_1 = p_2 = \frac{m}{2(m+n+k)}$ (and $p_0 = \frac{n+k}{m+n+k}$), called the simple improvement patroller strategy (with middle nodes).

The proof of lemma 4.2.9 follows by the PIP with the hybrid strategy $\boldsymbol{\pi}_{\text{SimpHyb}}$.

Proof. First consider the case of $m \in M_{4,0}^{S_n^k}$, so $M = \emptyset$, with the simple hybrid strategy $\boldsymbol{\pi}_{\text{SimpHyb}}$. The PIP is

$$\begin{aligned} \text{maximize} \quad & \min_{j \in N} \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, j}(S_n^k, T, m) p_i \\ \text{s.t.} \quad & p_i \in [0,1], i = 0,1,2, \\ & p_0 + p_1 + p_2 = 1. \end{aligned}$$

We can now simplify the objective function as we have either $j \in L$ or $j \in R$ (as $M = \emptyset$) and we know for the two sets that for any choice of p_1 and p_2:

- for all $j \in L$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,1}(S_n^k, T, m)p_i$,
- for all $j \in R$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_1}(S_n^k, T, m)p_i$.

Moreover $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_k}(S_n^k, T, m)p_i$ is equal for any $k \in \{1, ..., n-1\}$ and so we need only consider the nodes 1 and $*_1$ in the PIP. Hence the PIP, along with the performances (in equations (4.13) and (4.14)) and reduction of $p_0 = 1 - p_1 - p_2$, becomes

$$
\begin{array}{ll}
\text{maximize} & \min\left(\dfrac{m}{2(n+k)}(1-p_1-p_2)+p_1, \dfrac{m}{2(n+k)}(1-p_1-p_2)+p_2\right) \\
\text{s.t.} & p_i \in [0,1], i = 1, 2, \\
 & p_1 + p_2 \leq 1.
\end{array}
$$

From the objective function of the PIP we know that it is maximized when $p_1 = p_2$. Hence, if $m \geq n+k$ we get the optimal solution that $p_1 = p_2 = \frac{1}{2}$ as $\frac{m}{2(n+k)} \leq \frac{1}{2}$ when $m \in M_{4,0}^{S_n^k}$. Otherwise, if $m < n+k$ we get the optimal solution $p_1 = p_2 = 0$, so no improvement is made over $\boldsymbol{\pi}_0$. The optimal value gives the lower bound as given in the lemma.

Similarly in the case of $m \in M_{4,1}^{S_n^k}$ we can simplify the objective function of the PIP as for any choice of p_1 and p_2:

- for all $j \in L$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,1}(S_n^k, T, m)p_i$,
- for all $j \in M$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i = \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,\hat{m}+2}(S_n^k, T, m)p_i = \frac{m}{n+k}p_0$,
- for all $j \in R$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_1}(S_n^k, T, m)p_i$.

Moreover $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_k}(S_n^k, T, m)p_i$ is equal for any $k \in \{1, ..., n-1\}$ and so we need only consider the nodes 1, $\hat{m}+2$ and $*_1$ in the PIP. Hence the PIP, along with the performances (in equations (4.13) and (4.14)) and reduction of $p_0 = 1 - p_1 - p_2$, becomes

$$\begin{aligned} \text{maximize} \quad & \min\left(\frac{m}{2(n+k)}p_0 + p_1, \frac{m}{n+k}p_0, \frac{m}{2(n+k)}p_0 + p_2\right) \\ \text{s.t.} \quad & p_i \in [0,1], i = 1,2, \\ & p_1 + p_2 \leq 1. \end{aligned}$$

From the objective function of the PIP we know that it is maximized when $p_1 = p_2$, and furthermore when $(1-2p_1)\frac{m}{2(n+k)} + p_1 = (1-2p_1)\frac{m}{n+k}$. Hence the optimal solution has $p_1 = p_2 = \frac{m}{2(m+n+k)}$ and $p_0 = \frac{n+k}{m+n+k}$. The optimal value gives the lower bound as given in the lemma. □

From lemma 4.2.9 we remark that in the case of $m \in M_{4,0}^{S_n^k}$, we only have a strict improvement over the baseline $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$, when $m < n+k$.

When $m \in M_{4,0}^{S_n^k}$ when $m < n+k$, we have that that $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ are intercepting walks and each node in N is in one of the walks, thus they form a covering set $C = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2\}$, which is in fact minimal (as it is not possible to get a covering set with one patroller). Therefore $\mathcal{C}_{S_n^k,T,m} = 2$ and so we could alternatively use lemma 2.3.12 to achieve the bound present in the lemma 4.2.9 rather than the PIP.

We provide figure 4.2.11, showing the lower bound given by lemma 4.2.9. We see a sharp rise in the lower bound between $m = 11$ and $m = 12$, when the set of middle M transitions from being non-empty to empty, meaning $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ is no longer played and the covering strategy is used.

In the following section we find a partial solution when $m \in M_4^{S_n^k}$, by finding tight upper bounds via bespoke attacker strategies for $m \in M_{4,0}^{S_3^k}$ when $m < 3+k$. After this we apply PIP to $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ when $m \in M_5^{S_n^k}$ to improve the bound in equation (4.2), using similar strategies to those in this section to reach a strict improvement.

4.2.7 Solution when $m \in M_{4,0}^{S_3^k}$

In this section we present a partial solution for $m \in M_{4,0}^{S_n^k}$, by providing the optimal attacker strategy which provides a tight upper bound with the lower bound in lemma 4.2.9 when $n = 3$. In addition we provide an example to show how it is possible to get this tightness when $n \geq 4$.

For $m \in M_{4,0}^{S_n^k}$, we recall that the use of the simple hybrid patroller $\boldsymbol{\pi}_{\text{SimpHyb}}$ strategy (with no middle nodes) gives us a lower bound of

$$V(S_n^k, T, m) \geq \begin{cases} \frac{1}{2} \text{ for } m < n+k, \\ \frac{m}{2(n+k)} \text{ otherwise.} \end{cases}$$

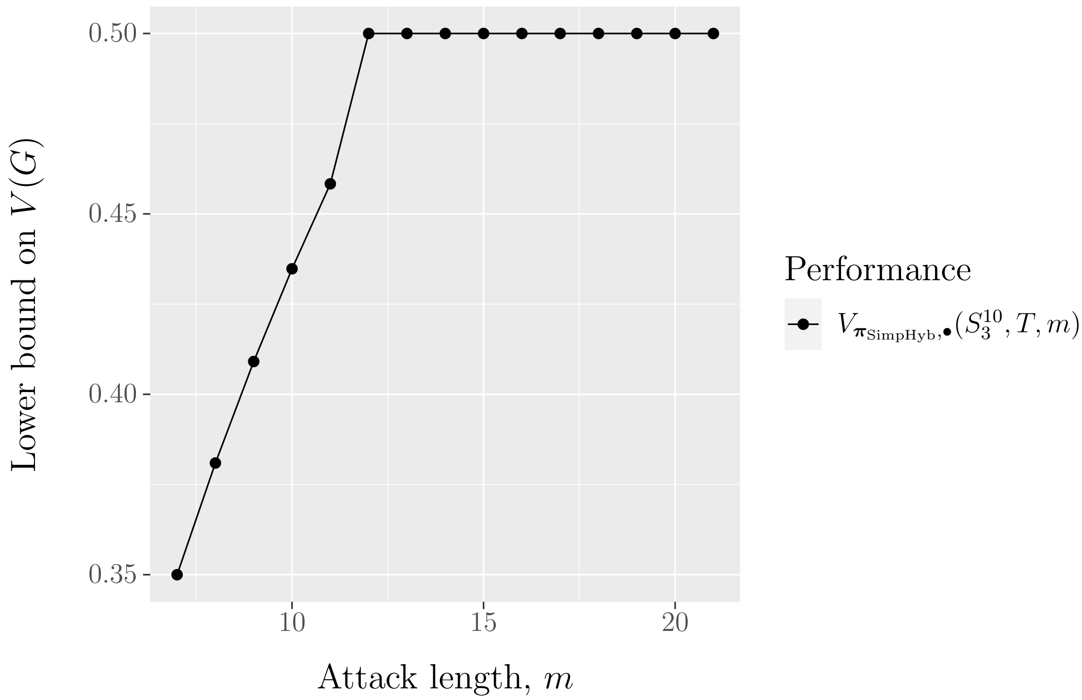


Figure 4.2.11: Lower bound on the game $G(S_3^{10}, T, m)$ provided by the simple patroller strategy $\boldsymbol{\pi}_{\text{SimpHyb}}$, for $m \in M_4^{S_n^k}$ $(7 \leq m \leq 21)$. For any $T \geq m$.

In addition, we recall that in the case of $m < n + k$, the patroller strategy is equivalent to a covering strategy (as $\mathcal{C}_{Q,T,m} = 2$), and is just the random minimal full-node cycle strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ otherwise. In the case of $m < n + k$, by the conditions on $m \in M_{4,0}^{S_n^k}$, we know the attack length region is just $m = n + k - 1$, where $n < k - 1$ and $n + k$ is odd. As the diameter of S_n^k is $\bar{d} = k + 2$ we get by lemma 3.3.13 an upper bound of

$$V(S_n^k, T, m) \leq \max\left(\frac{1}{2}, \min\left(\frac{n+k-1}{2(k+2)}, 1\right)\right),$$

achieved by the time-limited diametric attacker strategy $\boldsymbol{\phi}_{\text{tdi}}$. Hence we know the value of the game when $n = 3$ as $V(S_n^k, T, m) \leq \frac{1}{2}$ and hence arrive at the following lemma.

Lemma 4.2.10. *For the game $G(S_3^k, T, k+2)$ with an even k such that $k \geq 6$ and any $T \geq k + 2$, we have*

$$V(S_3^k, T, k+2) = \frac{1}{2},$$

achieved by the simple improvement patroller strategy $\boldsymbol{\pi}_{SimpHyb}$ (or the covering strategy $\boldsymbol{\pi}_{Cov}$) and the time-limited diametric attacker strategy $\boldsymbol{\phi}_{tdi}$.

For $n \geq 4$, we cannot use the time-limited diametric attacker strategy $\boldsymbol{\phi}_{\text{tdi}}$ to get a tight upper bound, as the bound is strictly greater than a half for all $n \geq 4$. While we have not been able to produce a general attacker strategy for $n \geq 4$, we provide example 4.2.11 to show that it is possible to get tight bounds with the lower bound in lemma 4.2.9 when $n \geq 4$.

Example 4.2.11. Consider the game $G(S_5^{10}, 20, 14)$. Note that $m = 14 \in M_{4,0}^{S_5^{10}}$ and $14 = n + k - 1$ and $n + k = 15$ is odd. Consider using the attacker strategy $\boldsymbol{\phi} \in \Phi$ such that the probability of choosing the pure attack $(j, \tau) \in \mathcal{A}$ is given by

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{16} & \text{if } j = 1 \text{ and } \tau \in \{0, 1, ..., 7\}, \\ \frac{1}{16} & \text{if } j \in \{*_1, ..., *_4\} \text{ and } \tau \in \{3, 4\}. \end{cases}$$

Using theorem 3.2.13 we know that we can evaluate the performance of $\boldsymbol{\phi}$ by

$$V_{\bullet,\boldsymbol{\phi}}(S_5^{10}, 20, 14) = \max_{\omega \in \Omega'''} P(\omega, \boldsymbol{\phi}),$$

in which Ω''' is a set whose elements are move-wait walks

$$\omega = ((j_1, \nu_1), (j_2, 0) ..., (j_l, 0))$$

such that $l \in \mathbb{N}$ where $j_i \in N_A$, for the set of possibly attacked nodes $N_A = \{1, *_1, ..., *_4\}$ and

$$\nu_1 \begin{cases} \in \{0, 1, ..., 7\} & \text{if } j_1 = 1 \\ \in \{3, 4\} & \text{if } j_1 = *_i \text{ for some } i \in \{1, 2, 3, 4\}. \end{cases}$$

Then by computing $P(\omega, \boldsymbol{\phi})$ for every element $\omega \in \Omega'''$ we find that

$$\max_{\omega \in \Omega'''} P(\omega, \boldsymbol{\phi}) = \frac{1}{2}.$$

Hence $V(S_5^{10}, 20, 14) \leq V_{\bullet,\boldsymbol{\phi}}(S_5^{10}, 20, 14) = \frac{1}{2}$ and along with lemma 4.2.9, which gives a tight lower bound, we get that

$$V(S_5^{10}, 20, 14) = \frac{1}{2}.$$

4.2.8 PIP when $m \in M_5^{S_n^k}$

In this section we use the patrol improvement program(PIP) to improve the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ for the game $G(S_n^k, T, m)$ when $m \in M_5^{S_n^k}$, where

$$M_5^{S_n^k} = \{m : 3 \leq m \leq 2n, m < 2(k + 1)\}.$$

Recall that when $m \in M_5^{S_n^k}$, we have $m \leq 2n$, so there is no decrease in performance at nodes along the line section towards the end (node $k+1$), unlike when $m \in M_4^{S_n^k}$. Let $\boldsymbol{\pi}_0 = \boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ be the baseline strategy for the PIP and seek to find strategies which perform better at weakly performing nodes.

We can find a strategy that can be used for the weakly performing left nodes, $L = \{1, ..., \hat{m}+1\}$, where $\hat{m} = \left\lfloor \frac{m}{2} \right\rfloor$. This is done by forming the walk

$$W_{\text{L}} = (1, 2, ..., \hat{m}+1, \hat{m}, ..., 2),$$

which is repeated for the time-horizon. W_{L} is of length $2\hat{m}$ and as $2\hat{m} \leq m$ the walk is intercepting as it visits every node at most m time units apart. The pure strategy W_{L} will used as a candidate for improvement in the PIP, for notational convince we will denote the strategy in mixed form $\boldsymbol{\pi}_1 = W_{\text{L}}$. Then the performance at node $j \in N$

$$V_{\boldsymbol{\pi}_1,\bullet,j}(S_n^k, T, m) = \mathbb{I}_{\{j \in L\}}. \tag{4.15}$$

We note that this is identical to the left improvement when $m \in M_4^{S_n^k}$.

We can find a strategy for the other weakly performing nodes, which are the star node $*_i$ for all $i \in \{1, ..., n-1\}$, at which $\boldsymbol{\pi}_0$ performs equally. However, unlike in the case of $m \in M_4^{S_n^k}$, this cannot always be done by a single intercepting strategy as $m \leq 2n$, as it is only possible when $m \in \{2n-2, 2n-1, 2n\}$. To form the star improvement strategy $\boldsymbol{\pi}_2$, the patroller chooses each subset $\chi \in \wp(\{*_1, *_2, ..., *_{n-1}\}, \hat{m})$ with equal probability, where $\wp(A, b)$ is the elements of the power set of A which are of cardinality b. Once χ is chosen, an intercepting walk is formed which contains all the nodes in χ, by repeating a closed walk which visits the nodes in χ. Note that the order of the star nodes in the closed walk is arbitrary. The star improvement $\boldsymbol{\pi}_2$ has a non-zero performance at nodes in $S = \{c, *_1, ..., *_{n-1}\}$. As the chance any given $*_i$ is chosen to be in χ is $\frac{\hat{m}}{n-1}$ for all $i \in \{1, ..., n-1\}$, the performance of $\boldsymbol{\pi}_2$ at node $j \in N$ is

$$V_{\boldsymbol{\pi}_2,\bullet,j}(S_n^k, T, m) = \begin{cases} \frac{\hat{m}}{n-1} & \text{if } j \in \{*_1, ..., *_{n-1}\}, \\ 1 & \text{if } j = c, \\ 0 & \text{if } j \in \{1, ..., k+1\}. \end{cases} \tag{4.16}$$

We can now form the combinatorial hybrid strategy $\boldsymbol{\pi}_{\text{CombHyb}}$ which plays $\boldsymbol{\pi}_i$ with probability p_i for $i = 0, 1, 2$ and use the PIP to determine the best probabilities which achieve the best improvement over the baseline strategy $\boldsymbol{\pi}_0 (= \boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k})$. However before immediately using the PIP with our left and right improvements, we must first consider if any nodes are not contained within either improvement. That is if the set of middle nodes $M = N \setminus (L \cup S)$ is empty or not. Figure 4.2.12 shows an example of an elongated star graph game in which the left and star improvements are used and contain the entire node set, so $M = \emptyset$. Figure 4.2.13 shows the related game in which the branch elongation k is increased from 2 to 5. Note that the left and star improvements no longer contain the entire node set N, so $M \neq \emptyset$.

By considering the nodes $\hat{m}+1$ and $k+1$, we have that

$$M = \emptyset \iff \hat{m} \geq k.$$

We use this condition to further decompose $M_5^{S_n^k}$ into

$$M_{5,0}^{S_n^k} \equiv M_5^{S_n^k} \cap \{m \mid \hat{m} \geq k\} = \{2k, 2k+1\}$$

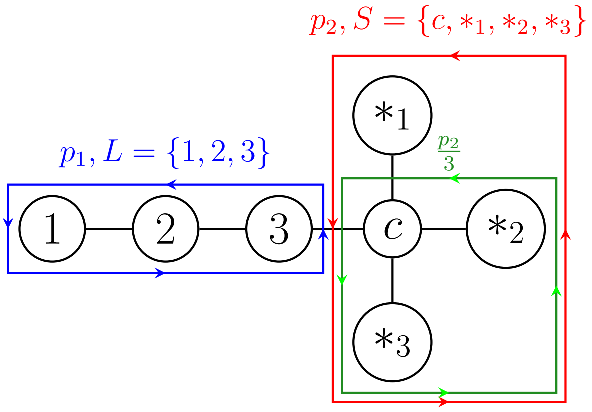


Figure 4.2.12: The game $G(S_4^2, 20, 4)$ with the left improvement $\boldsymbol{\pi}_1$ played with probability p_1 shown in blue and the star improvement $\boldsymbol{\pi}_2$ played with probability p_2. The green box shows one of the three repeated closed walks, using the set $\chi = \{*_2, *_3\}$, chosen from $\wp(\{*_1, *_2, ..., *_{n-1}\}, \hat{m}) = \{\{*_1, *_2\}, \{*_1, *_3\}, \{*_2, *_3\}\}$. The sets are $L = \{1, 2, 3\}$ and $S = \{c, *_1, *_2, *_3\}$ ($M = \emptyset$).

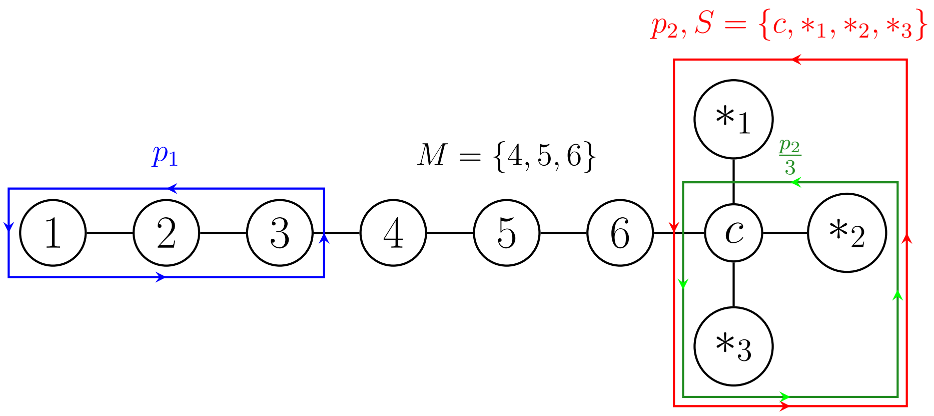


Figure 4.2.13: The game $G(S_4^2, 20, 4)$ with the left improvement $\boldsymbol{\pi}_1$ played with probability p_1 shown in blue and the star improvement $\boldsymbol{\pi}_2$ played with probability p_2. The green box shows one of the three repeated closed walks, using the set $\chi = \{*_2, *_3\}$, chosen from $\wp(\{*_1, *_2, ..., *_{n-1}\}, \hat{m}) = \{\{*_1, *_2\}, \{*_1, *_3\}, \{*_2, *_3\}\}$. The sets are $L = \{1, 2, 3\}$, $S = \{c, *_1, *_2, *_3\}$ and $M = \{4, 5, 6\}$).

and

$$M_{5,1}^{S_n^k} \equiv M_5^{S_n^k} \cap \{m \mid \hat{m} < k\}.$$

As in section 4.2.6, we make this distinction of $M = \emptyset$ or $M \neq \emptyset$, as it affects the PIP's optimization of probabilities p_1 and p_2. The following lemma contains the improved lower bound, over the lower bound achieved by using $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ (as in equation 4.2)) for games when either $m \in M_{5,0}^{S_n^k}$ or $m \in M_{5,1}^{S_n^k}$.

Lemma 4.2.12. *For the game $G(S_n^k, T, m)$, for any $n \geq 3$, for any $k \geq 1$, for all $T \geq m$,*

- *for $m \in M_{5,0}^{S_n^k}$ we have*

$$V(S_n^k, T, m) \geq \frac{\hat{m}}{\hat{m} + n - 1},$$

achieved by the combinatorial hybrid strategy, $\boldsymbol{\pi}_{CombHyb}$, with $p_1 = \frac{k}{k+n-1}$, $p_2 = \frac{n-1}{k+n-1}$ (and $p_0 = 0$), called the combinatorial improvement strategy (with no middle nodes).

- *for $m \in M_{5,1}^{S_n^k}$ we have*

$$V(S_n^k, T, m) \geq \frac{2m}{2(n+k) + m(1 + \frac{n-1}{\hat{m}})},$$

achieved by the combinatorial hybrid strategy, $\boldsymbol{\pi}_{CombHyb}$, with $p_1 = \frac{m}{2(n+k)+m(1+\frac{n-1}{\hat{m}})}$, $p_2 = \frac{m(n-1)}{\hat{m}(2(n+k)+m(1+\frac{n-1}{\hat{m}}))}$ and $p_0 = \frac{2(n+k)}{m+n+k}$, called the combinatorial improvement strategy (with middle nodes).

In which $\hat{m} = \left\lfloor \frac{m}{2} \right\rfloor$.

The proof of lemma 4.2.12 is similar to that of lemma 4.2.9 and follows from the PIP.

Proof. First consider the case of $m \in M_{5,0}^{S_n^k}$, with the combinatorial hybrid strategy $\boldsymbol{\pi}_{\text{CombHyb}}$. The PIP is,

$$\begin{array}{lll} \text{maximize} & \displaystyle\min_{j \in N} \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, j}(S_n^k, T, m) p_i \\ \text{s.t} & p_i \in [0,1], i = 0,1,2, \\ & p_0 + p_1 + p_2 = 1. \end{array}$$

We can now simplify the objective function as we have either $j \in L$ or $j \in S$ (as $M = \emptyset$) and we know for the two sets that for any choice of p_1 and p_2,

- for all $j \in L$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k,T,m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,1}(S_n^k,T,m)p_i$,
- for all $j \in S$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k,T,m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_1}(S_n^k,T,m)p_i$.

Moreover $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_k}(S_n^k,T,m)p_i$ is equal for any $k \in \{1,...,n-1\}$ and so we need only consider the nodes 1 and $*_1$ in the PIP. Hence the PIP, along with the performances (in equations (4.15) and (4.16)) and reduction of $p_0 = 1 - p_1 - p_2$, becomes

$$\begin{array}{ll} \text{maximize} & \min\left(\frac{m}{2(n+k)}(1-p_1-p_2)+p_1, \frac{m}{2(n+k)}(1-p_1-p_2)+\frac{\hat{m}}{n-1}p_2\right) \\ \text{s.t} & p_i \in [0,1], i = 1,2, \\ & p_1 + p_2 \leq 1. \end{array}$$

From the objective function of the PIP we know that it is maximized when $p_1 = \frac{\hat{m}}{n-1}p_2$ so we get the optimal solution that $p_1 = \frac{\hat{m}}{\hat{m}+n-1}$ and $p_2 = \frac{n-1}{\hat{m}+n-1}$ as $\frac{m}{2(n+k)} \leq \frac{1}{2}$ when $m \in M_{5,0}^{S_n^k}$. In addition, when $m \in M_{5,0}^{S_n^k}$ implies that $\hat{m} = k$, so $p_1 = \frac{k}{k+n-1}$ and $p_2 = \frac{n-1}{k+n-1}$. The optimal value gives the bound given in the lemma.

Similarly in the case of $m \in M_{5,1}^{S_n^k}$, we have a simplification of the objective function of the PIP as for any choice of p_1 and p_2,

- for all $j \in L$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k,T,m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,1}(S_n^k,T,m)p_i$,
- for all $j \in M$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k,T,m)p_i = \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,\hat{m}+2}(S_n^k,T,m)p_i = \frac{m}{n+k}p_0$,
- for all $j \in S$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k,T,m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_1}(S_n^k,T,m)p_i$.

Moreover $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_k}(S_n^k,T,m)p_i$ is equal for any $k \in \{1,...,n-1\}$ and so we need only consider the nodes 1, $\hat{m}+2$ and $*_1$ in the PIP. Hence the PIP, along with the performances (in equations (4.15) and (4.16)) and reduction of $p_0 = 1 - p_1 - p_2$, becomes

$$\begin{array}{ll} \text{maximize} & \min\left(\frac{m}{2(n+k)}p_0+p_1, \frac{m}{n+k}p_0, \frac{m}{2(n+k)}p_0+\frac{\hat{m}}{n-1}p_2\right) \\ \text{s.t} & p_i \in [0,1], i = 1,2, \\ & p_1 + p_2 \leq 1. \end{array}$$

From the objective function of the PIP we know that it is maximized when $p_1 = \frac{\hat{m}}{n-1}p_2$, and $(1-(1+\frac{n-1}{\hat{m}})p_1)\frac{m}{2(n+k)}+p_1 = (1-(1+\frac{n-1}{\hat{m}})p_1)\frac{m}{n+k}$. Hence the optimal solution has $p_1 = \frac{m}{2(n+k)+m(1+\frac{n-1}{\hat{m}})}$, $p_2 = \frac{m(n-1)}{\hat{m}(2(n+k)+m(1+\frac{n-1}{\hat{m}}))}$ and $p_0 = \frac{2(n+k)}{m+n+k}$. The optimal value gives the bound as given in the lemma. □

We provide figure 4.2.14 showing the lower bound given by lemma 4.2.12 which highlights the issue with lower bound achieved by using $\boldsymbol{\pi}_{\text{CombHyb}}$ as m alternates between odd and even values. The increase in the lower bound between even to odd attack lengths is shallow, due to choosing the set χ from $\wp(\{1,...,n-1\},\hat{m})$ as $\hat{m} = \lfloor \frac{m}{2} \rfloor$ is the same for m odd and $m-1$ which would be even. Thus m odd and $m-1$ have the same star improvement $\boldsymbol{\pi}_2$ played with the same probability making the strategy $\boldsymbol{\pi}_{\text{CombHyb}}$ inefficient.

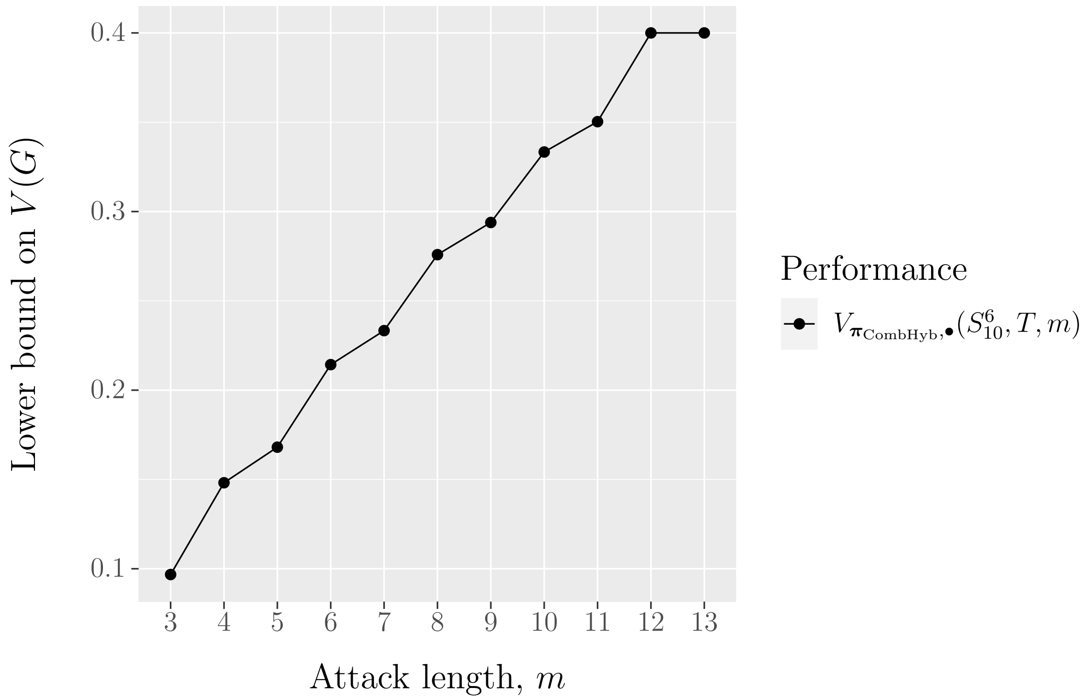


Figure 4.2.14: Lower bound on the game $G(S^6_{10}, T, m)$ provided by the combinatorial hybrid strategy $\boldsymbol{\pi}_{\text{CombHyb}}$ for $m \in M_5^{S^k_n}$ $(3 \leq m \leq 13)$ and any $T \geq m$.

To rectify the inefficiency, we can consider adjusting the star improvement strategy $\boldsymbol{\pi}_2$, when m is odd, to form the adjusted star improvement strategy $\boldsymbol{\pi}'_2$, which replaces $\boldsymbol{\pi}_2$ in $\boldsymbol{\pi}_{\text{CombHyb}}$, to form the adjusted combinatorial hybrid $\boldsymbol{\pi}_{\text{AdjCombHyb}}$. To form $\boldsymbol{\pi}'_2$, we allow the use of non-intercepting patroller strategies, therefore allowing for different length closed walks to be used. We define a *proxy attack length*

$$m' = \begin{cases} m \text{ if } m \text{ even} \\ m+r \text{ if } m \text{ odd}, \end{cases}$$

for some $r \in \{1,3,...,2(n-1)-m\}$. This proxy attack length is used to form a closed walk of length m', in which $\hat{m}' = \lfloor \frac{m'}{2} \rfloor$ star nodes are chosen for each pure patrol in the star improvement. That is $\boldsymbol{\pi}_2'$ now chooses each $\chi \in \wp(\{*_1, ..., *_{n-1}\}, \hat{m}')$ with equal probability to form a closed walk visiting only the nodes in χ. Then the performance of $\boldsymbol{\pi}_2'$ at node $j \in N$ is

$$V_{\boldsymbol{\pi}_2', \bullet, j}(S_n^k, T, m) = \begin{cases} \frac{m}{2(n-1)} & \text{if } j \in \{*_1, ..., *_{n-1}\}, \\ 1 & \text{if } j = c, \\ 0 & \text{if } j \in \{1, ..., k+1\}. \end{cases} \tag{4.17}$$

We can see this as if $j = *_i$ for some $i \in \{1, .., n-1\}$, then the performance is formed by the product of the two independent event probabilities; choosing j to be in the set χ, and j's performance in the patrol. I.e. $\frac{\hat{m}'}{n-1} \times \frac{m}{2\hat{m}'} = \frac{m}{2(n-1)}$. Another way to see this star improvement is to perform the repeated closed walk $(*_1, c, *_2, ..., c, *_{n-1}, c, *_1)$ $(r = 2(n-1) - m)$ which is of length $2(n-1)$ and so provides a performance at star nodes of $\frac{m}{2(n-1)}$. Using $\boldsymbol{\pi}_{\text{AdjCombHyb}}$, we can once again apply the PIP and achieve better lower bounds that that achieved by using $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$. The results of this are shown in the following lemma.

Lemma 4.2.13. *For the game $G(S_n^k, T, m)$, for any $n \geq 3$, for any $k \geq 1$, for all $T \geq m$, we have,*

- *for $m \in M_{5,0}^{S_n^k}$,*
$$V(S_n^k, T, m) \geq \frac{m}{m + 2(n-1)},$$
achieved by the adjusted combinatorial hybrid strategy, $\boldsymbol{\pi}_{AdjCombHyb}$, with $p_1 = \frac{m}{m+2(n-1)}$, $p_2 = \frac{2(n-1)}{m+2(n-1)}$ (and $p_0 = 0$), called the adjusted combinatorial improvement patroller strategy (with no middle nodes).

- *for $m \in M_{5,1}^{S_n^k}$,*
$$V(S_n^k, T, m) \geq \frac{2m}{2(n+k) + m + 2(n-1)},$$
achieved by the adjusted combinatorial hybrid strategy, $\boldsymbol{\pi}_{AdjCombHyb}$, with $p_1 = \frac{m}{2(n+k)+m+2(n-1)}$, $p_2 = \frac{2(n-1)}{2(n+k)+m+2(n-1)}$ and $p_0 = \frac{2(n+k)}{m+n+k}$, called the adjusted combinatorial improvement patroller strategy (with middle nodes).

We leave the proof of lemma 4.2.13 to appendix B.1, as it is essentially the same as that of lemma 4.2.12, with $\hat{m}$ replaced with $\frac{m}{2}$. This small adjustment when m is odd provides a strict improvement. Figure 4.2.15 shows how this adjustment smooths out the increase in the strategies performance (lower bound) as m alternates between even and odd.

4.2.9 Solution when $M_5^{S_n^k}$

In this section, we provide a tight bound to the lower bounds in lemma 4.2.13 for the game $G(S_n^k, T, m)$ when $m \in M_5^{S_n^k}$. We first consider case 1: $m \in M_{5,0}^{S_n^k}$,

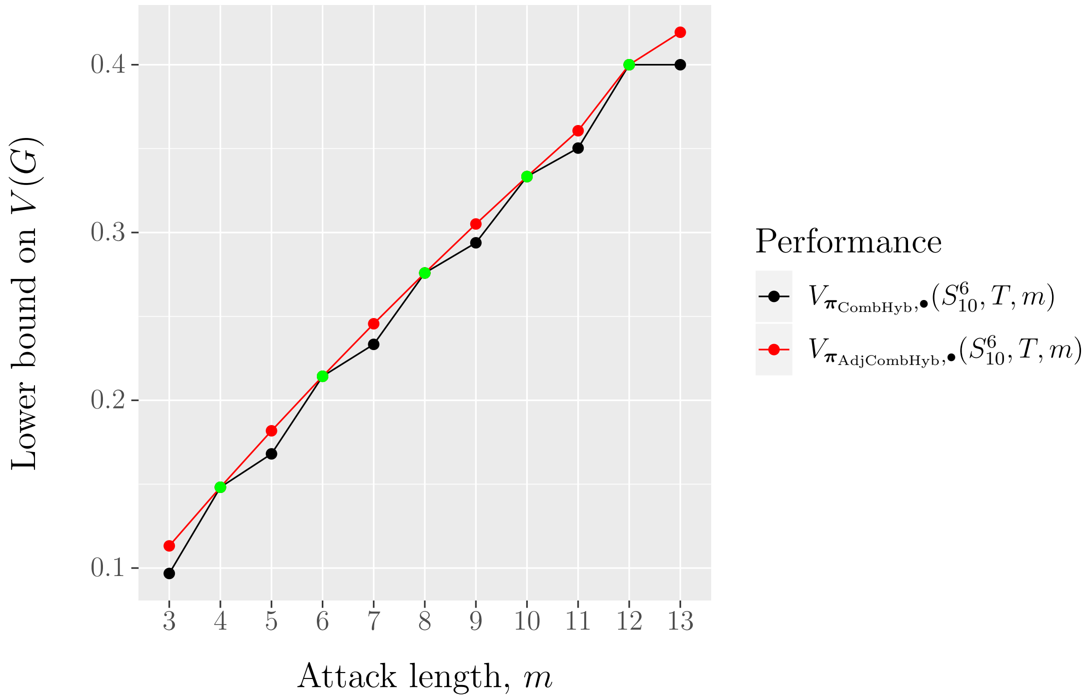


Figure 4.2.15: Lower bounds on the game $G(S^6_{10}, T, m)$ provided by the combinatorial improvement strategy $\boldsymbol{\pi}_{\text{CombHyb}}$ in black and the adjusted combinatorial improvement strategy $\boldsymbol{\pi}_{\text{AdjCombHyb}}$ in red. Shown for $m \in M_5^{S_n^k}$ $(3 \leq m \leq 13)$, for any $T \geq m$. The strategies can be seen to provide the same lower bound when m is even, with points shown in green, as the adjustment is not needed.

the case where $M = \emptyset$ in which we find that a reduced version of the time-centred attacker strategy provides a tight upper bound. This is followed by case 2: $m \in M_{5,1}^{S_n^k}$, the case where $M \neq \emptyset$, in which we start by looking at simplification as a method to obtain an optimal attacker strategy. Case 2 will be concluded in the following section as it requires multiple bespoke attacker strategies to be created in order to obtain a tight or near tight upper bound.

Case 1: $m \in M_{5,0}^{S_n^k}$

Let us first recall that there are only two possible attack lengths of $m = 2k$, or $m = 2k + 1$ when $M = \emptyset$. Further to this, $m = 2k$ is only in the region $M_{5,0}^{S_n^k}$ if $k \leq n$ and similarly $m = 2k+1$ is only in the region if $k \leq n-1$. We seek a tight upper bound to match that of the lower bound from lemma 4.2.13, viz.

$$V(S_n^k, T, m) \geq \frac{m}{m + 2(n-1)}. \tag{4.18}$$

The time-centred attacker strategy $\boldsymbol{\phi}_{\text{tc}}$, for $m < 2(k+1)$ gives us the upper bound of

$$V(S_n^k, T, m) \leq \frac{k+1}{n+k}.$$

So the time-centred attacker strategy does not achieve a tight upper bound to match equation (4.18), but it is extremely close. The strategy is not tight because for $m < 2(k+1)$, the best response pure patroller waits at node 1 to see $2(k+1)$ out of the $2(n+k)$ potential attacker agents. Knowing that this is the reason the time-centred attacker strategy is not tight with equation (4.18), allows us to adjust it to match the desired lower bound. We can do this by reducing the number of agents who attack node 1, reducing it from $2(k+1)$ agents to m agents. This ensures that any pure patroller who waits at node 1 only catches m agents. We define this attack formally as the reduced time-centred attacker strategy.

Definition 4.2.14. The *reduced time-centred* attacker strategy $\boldsymbol{\phi}_{\text{rtc}}$ is such that the probability of choosing the pure attack $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{m+2(n-1)} & \text{if } j = 1 \text{ and } \tau \in \{0, 1, ..., m-1\}, \\ \frac{1}{m+2(n-1)} & \text{if } j \in \{*_1, ..., *_{n-1}\} \text{ and } \tau \in \{\hat{m}, \hat{m}+1\}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\hat{m} = \left\lfloor \frac{m}{2} \right\rfloor$.

In addition to altering how a pure patroller waiting at node 1 only gets m agents, any pure patroller starting at $*_i$ for some $i = 1, ..., n-1$ then arriving at node 1 will get less agents. An example of the reduced time-centred attack can be seen in figure 4.2.16. The lowering of the attack length from $m = 8$ to $m = 7$ can be seen when comparing figure 4.2.16 to the time-centred attack in figure 4.2.6, in which we see the idea of reducing the number of agents that the pure patroller catches. Using the reduced time-centred attacker strategy, we are able to find the desired upper bound which is tight with equation (4.18).

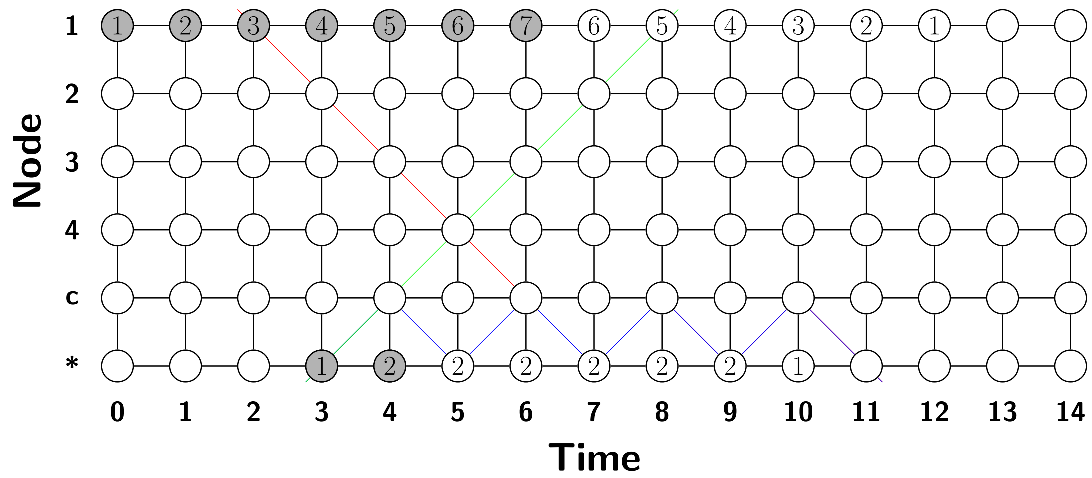


Figure 4.2.16: Space-time agent matrix $\boldsymbol{S}^A_{\text{rtc}}$ for the reduced time-centred attacker strategy $\boldsymbol{\phi}_{\text{rtc}}$ for the game $G(S^3_n, 14, 7)$ for any $n \geq 4$. Three example pure patrollers are drawn in red, blue and green, catching at most 7 out of $7+2(n-1)$ agents.

Lemma 4.2.15. *For a game $G(S^k_n, T, m)$, for all $n \geq 3$, for all $k \geq 1$, for all $m \in M^{S^k_n}_{5,0}$ and for all $T \geq 2m-1$ we have*

$$V(S^k_n, T, m) \leq \frac{m}{m+2(n-1)},$$

achieved by the reduced time-centred attacker strategy $\boldsymbol{\phi}_{rtc}$.

We leave the proof of lemma 4.2.15 to appendix B.2, as it is essentially the same as for lemma 4.2.6 with the waiting time ν_1 for the move-wait walk starting at node 1 reduced from a maximum of $2k+1$ to m. Lemmas 4.2.15 and 4.2.13 have tight upper and lower bounds and hence we know that for $m \in M^{S^k_n}_{5,0}$ that the adjusted combinatorial patroller strategy with no middle nodes and the reduced time-centred attacker strategy are optimal and so we know the value of the game. We summarize this in the following corollary.

Corollary 4.2.16. *For a game $G(S^k_n, T, m)$ for all $n \geq 3$, for all $k \geq 1$, for all $T \geq 2m-1$ and for all $m \in M^{S^k_n}_{5,0}$, we have,*

$$V(S^k_n, T, m) = \frac{m}{m+2(n-1)},$$

achieved by the adjusted combinatorial patroller strategy $\boldsymbol{\pi}_{AdjCombHyb}$ (with no middle nodes) and the reduced time-centred attack strategy $\boldsymbol{\phi}_{rtc}$.

Corollary 4.2.16 concludes the solution for case 1. It may seem natural to consider further reduction to the time-centred attacker strategy when $m \in M^{S^k_n}_{5,1}$, however the lower bound given in lemma 4.2.13 is not tight with this bound, as $\frac{m}{m+2(n-1)} \geq \frac{2m}{2(n+k)+m+2(n-1)}$ for all $m \in M^{S^k_n}_{5,1}$. Thus suggesting, if the adjusted combinatorial improvement strategy (with middle node) is optimal, that the attacker can make an even more efficient attacker strategy, placing more agents. Intuitively, when the attack length is very low, only attacking the leaf nodes seems suboptimal as there are more nodes which are independent which could be attacked along the line segment. We will see this to be true, and also that the placement of these potential attacking agents for non-leaf nodes can be seen through simplification followed by some bespoke adjustments as required.

Case 2: $m \in M_{5,1}^{S_n^k}$

Case 2 deals with the case of $M \neq \emptyset$, that is middle nodes are present. We seek a tight upper bound to match that of the lower bound from lemma 4.2.13, viz.

$$V(S_n^k, T, m) \geq \frac{2m}{2(n+k)+m+2(n-1)}. \tag{4.19}$$

To find such an upper bound to match that of equation (4.19) we first consider a simplification of S_n^k to $S_{n+\left\lfloor \frac{k}{2} \right\rfloor}$ in which the set

$$N_{\mathrm{s}} = \begin{cases} \{1,3,5,...,k\} & \text{if } k \text{ is odd,} \\ \{2,4,6,...,k\} & \text{if } k \text{ is even,} \end{cases}$$

are node-identified with the centre c. Figure 4.2.17 shows an example of this simplification, when $k=5$, simplifying S_4^5 to S_6. For the game $G(S_{n+\left\lfloor \frac{k}{2} \right\rfloor}, T, m)$ the value and optimal strategies are known and hence we can use corollary 3.3.8 to get an upper bound on the game $G(S_n^k, T, m)$ of

$$V(S_n^k, T, m) \leq V(S_{n+\left\lfloor \frac{k}{2} \right\rfloor}, T, m) = \frac{m}{2n+2\left\lfloor \frac{k}{2} \right\rfloor}, \tag{4.20}$$

achieved by the embedding of an optimal attacker strategy for the game $G(S_{n+\left\lfloor \frac{k}{2} \right\rfloor}, T, m)$. Namely $\boldsymbol{\phi}_{\mathrm{fs}}$ such that

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2\left(n+\left\lfloor \frac{k}{2} \right\rfloor\right)} & \text{for } j \in N_{\mathrm{s}} \cup \{*_1, ..., *_{n-1}\} \text{ and } \tau \in \{0,1\}, \\ 0 & \text{otherwise.} \end{cases}$$

achieves the bound in equation (4.20).

It is clear that the upper bound in equation (4.20) does not match the lower bound in equation (4.19) (unless $m=2$, in which case this embedded attacker strategy is equivalent to the independence attacker strategy and is optimal). We provide figure 4.2.18 to highlight that using such a simplification, of S_n^k to $S_{n+\left\lfloor \frac{k}{2} \right\rfloor}$, produces a greatly suboptimal strategy. Note that while we do not know it is suboptimal, we later see, in this section and the following section, that it is possible to get the desired upper bound which matches that of equation (4.19) and so the inequality in equation (4.19) becomes equality.

We see in the following section, which continues the solution for when $m \in M_{5,1}^{S_n^k}$, that by considering other simplifications we achieve a tight upper bound with

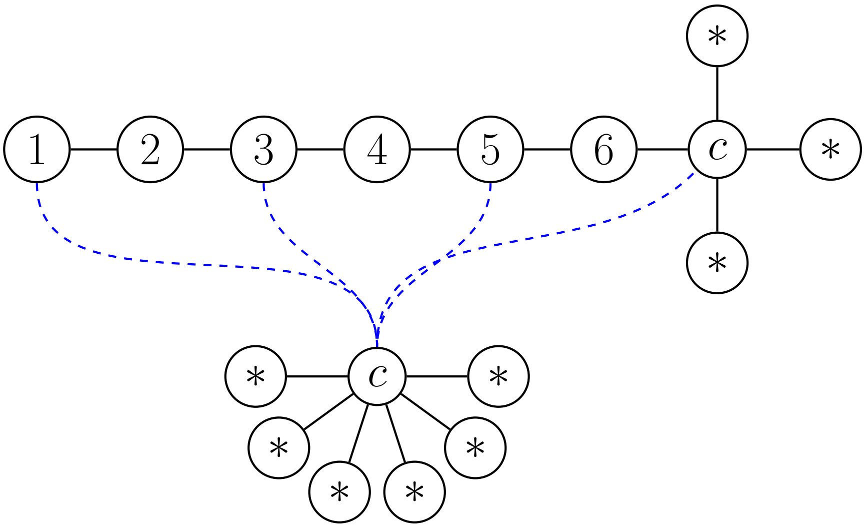


Figure 4.2.17: Simplification of S_4^5 to S_6 with nodes $1, 3, 5$ node-identified with c shown in the figure with dashed lines. Note due to the symmetry of the star nodes we have chosen to simply denote them by $*$ in the figure.

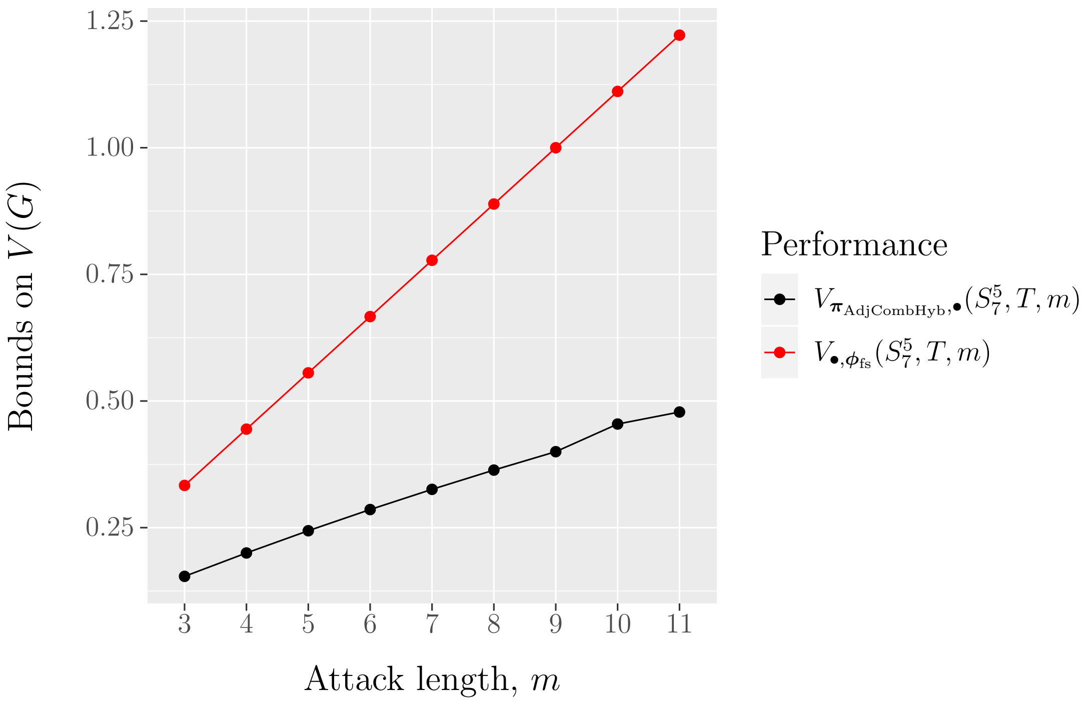


Figure 4.2.18: Lower bound on the game $G(S_7^5, T, m)$, provided by adjusted combinatorial improvement strategy $\boldsymbol{\pi}_{\text{AdjCombHyb}}$, in black, (which we later see is an optimal strategy) and the upper bound provided by the simplified time-centred attacker strategy $\boldsymbol{\phi}_{\text{fs}}$, in red. Plotted for $m = 3, ..., 11$ and for any $T \geq m + 1$.

equation (4.19) for certain attack lengths in the set $M_{5,1}^{S_n^k}$. To do so, we require the simplification to depend on the attack length m, with the simplified graph being an elongated star graph for which the patrolling game's solution is known for the attack length.

4.2.10 Optimal attacker strategies by simplification

In this section we continue to seek an upper bound which is tight with the lower bound given in equation (4.19) (from lemma 4.2.13 in the case of $m \in M_{5,1}^{S_n^k}$). We consider the simplification of S_n^k to S_{n+l}^{k-2l} for some $l \in \{0, 1, ..., \lfloor \frac{k}{2} \rfloor + 1\}$ and by doing so we immediately find a solution for a subset of $M_{5,1}^{S_n^k}$, when $m - 2k - 2 \mod 4 = 0$. For other attack lengths we require bespoke manipulations of embedded attacker strategies in order to find effective attacker strategies. We define $\rho = m - 2k - 2 \mod 4$ to further divide the attack length regions $M_{5,1}^{S_n^k}$, where the optimal attacker strategy differs. However we do not use new notation for these attack length regions, as we predict that the value remains the same for any ρ, with only the optimal attacker strategy varying dependent on ρ. In fact we show that for $\rho \in \{0, 2\}$ the value is the same and for $\rho \in \{1, 3\}$ we have extremely close bounds.

To simplify S_n^k to S_{n+l}^{k-2l} for some $l = 0, 1, ..., \lfloor \frac{k}{2} \rfloor + 1$, each node in the set $N_s(l) = \{k, k-2, ..., k-2(l-1)\}$ is identified with c. In addition we relabel nodes $j = k+1, ..., k+1-2(l-1)$ as $j = *_n, ..., *_{n+l-1}$. By using the time-centred attacker strategy $\boldsymbol{\phi}_{\text{tc}}$, which is optimal for the game on the graph S_{n+l}^{k-2l}, we get the following bound, by corollary 3.3.8 and lemma 4.2.6. For any $n \geq 3$, for any $k \geq 1$, for any $m \geq 2(k - 2l + 1)$ and for any $T \geq 2k + m + 1 - 4l$, of

$$V(S_n^k, T, m) \leq V(S_{n+l}^{k-2l}, T, m) = \frac{m}{2(n+k-l)}. \tag{4.21}$$

Note that it is also possible to get an upper bound for $m < 2(k - 2l + 1)$, which is $V(S_n^k, T, m) \leq \frac{k-2l}{n+k-l}$, however this is not used. The time-centred attacker strategy $\boldsymbol{\phi}_{\text{tc}}$ for the game $G(S_{n+l}^{k-2l}, T, m)$ chooses the pure attack $(j, \tau) \in \mathcal{A}$ with probability

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{n+k-2l} & \text{for } j = 1 \text{ and } \tau \in \{0, ..., 2(k-2l)+1\}, \\ \frac{1}{n+k-2l} & \text{for } j \in \{*_1, ..., *_{n+l-1}\} \text{ and } \tau \in \{k-2l, k+1-2l\}, \\ 0 & \text{otherwise.} \end{cases}$$

Embedding $\boldsymbol{\phi}_{tc}$ from $G(S_{n+l}^{k-2l}, T, m)$ into the game $G(S_n^k, T, m)$ results in an attacker strategy $\boldsymbol{\phi}(l)$ which chooses $(j, \tau) \in \mathcal{A}$ with probability

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{n+k-2l} & \text{for } j = 1 \text{ and } \tau \in \{0, ..., 2(k-2l)+1\}, \\ \frac{1}{n+k-2l} & \text{for } j =\in \{*_1, ..., *_{n-1}\} \text{ and } \tau \in \{k-2l, k+1-2l\}, \\ \frac{1}{n+k-2l} & \text{for } j \in N_s(l) \text{ and } \tau \in \{k-2l, k+1-2l\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.22}$$

That is $\phi(l)$ gives the bound in equation (4.21) and from this upper bound it is clear that to achieve the best upper bound the attacker should choose the minimal l possible. However, the bound is only valid for $m \geq 2(k-2l+1)$ and so as we are considering $m \in M_{5,1}^{S_n^k}$ we are limited to pick the minimal l such that $2(k-2l+1) \in M_{5,1}^{S_n^k}$. That is we pick l to be $l^*(m) = \lfloor \frac{2k+2-m}{4} \rfloor$, and achieve the best possible upper bound (by this simplification), for any $n \geq 3$, for any $k \geq 1$, for any $m \in M_{5,1}^{S_n^k}$ and for any $T \geq 2k+m+1-4l^*(m)$ of

$$V(S_n^k, T, m) \leq \frac{m}{2(n+k-l^*(m))}. \tag{4.23}$$

We call the attacker strategy which generates the upper bound in equation (4.23), the 0-*simplified time-centred* attacker strategy $\phi_{0-stc} = \phi(l^*(m))$, which has the probability of choosing (j, τ) as equation (4.22) with $l = l^*(m)$. That is, for $\phi_{\rho=0}$ the attack chooses; with probability

$$\frac{2m}{2(n+k)+m+2(n-1)}$$

to attack node 1, then with equal probability choosing a commencement time from $0, ..., m-1$; or chooses node $*_i$ for $i = 1, ..., n-1$ with probability

$$\frac{2}{2(n+k)+m+2(n-1)}$$

and then with equal probability a commencement time from $\frac{m}{2}-1, \frac{m}{2}$; or chooses nodes j for $j = k+1, k-1, ..., \frac{m}{2}+2$ with probability

$$\frac{2}{2(n+k)+m+2(n-1)}$$

and then with equal probability a commencement time from $\frac{m}{2}-1, \frac{m}{2}$. Figure 4.2.19 shows the space-time agent matrix $\boldsymbol{S}_{0-stc}^{A}$ representation for the attacker strategy ϕ_{0-stc}.

The upper bound in equation (4.23) and the lower bound in equation (4.19) are tight for attack lengths that are such that $\rho = 0$. This can be seen in figure 4.2.20 in which the upper and lower bounds are equal when $\rho = 0$. We note that $\rho = 0$ corresponds to when $l^*(m)$ does not require rounding by the floor function and so the upper bound in equation (4.23) becomes

$$V(S_n^k, T, m) \leq \frac{2m}{2(n+k)+m+2(n-1)}. \tag{4.24}$$

Hence, for $\rho = 0$ we arrive at the following lemma, as the upper bound in equation (4.24) is tight with the lower bound in lemma 4.2.13.

Lemma 4.2.17. *For the game $G(S_n^k, T, m)$, for all $n \geq 3$, $k \geq 1$, for all $m \in M_{5,1}^{S_n^k} \cap \{m : \rho = 0\}$ and for all for all $T \geq 2(k - 2l^*(m)) + m$ we have*

$$V(Q, T, m) = \frac{2m}{2(n+k)+m+2(n-1)},$$

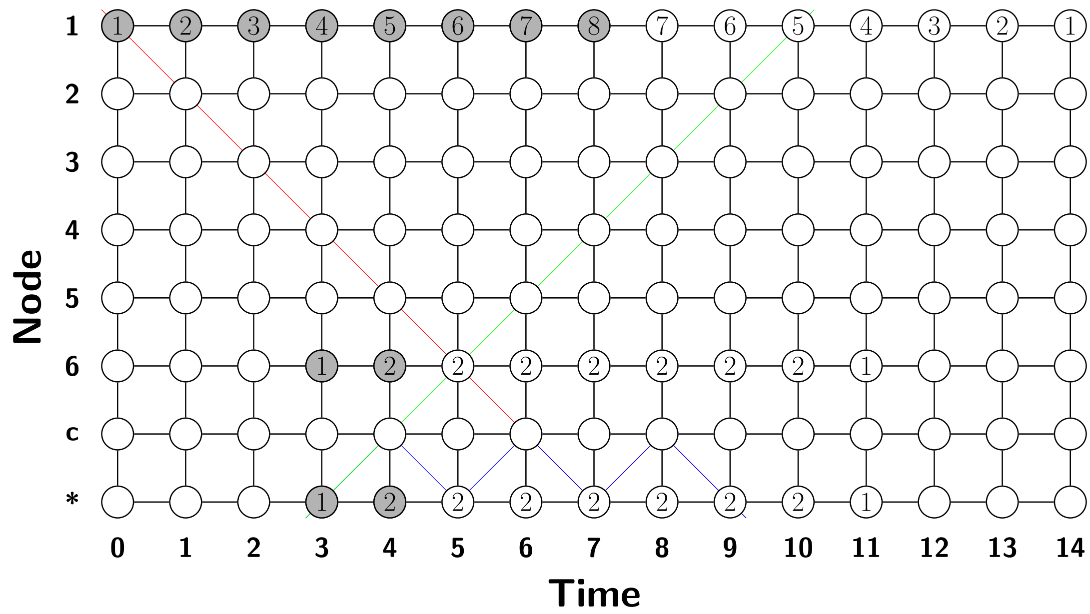


Figure 4.2.19: Space-time agent matrix $\boldsymbol{S}^A_{0-stc}$ for the 0-simplified time-centred attacker strategy $\boldsymbol{\phi}_{0-stc}$ for the game $G(S^5_7, 20, 8)$. With the strategy equivalent to the embedded time-centred attacker strategy from the game $G(S^3_8, 20, 8)$, for which node 7 is identified with c and 6 is relabelled as a star node. Three example pure patrollers are shown in red, green and blue.

achieved by the adjusted combinatorial improvement patroller strategy (with middle nodes) $\boldsymbol{\pi}_{AdjCombHyb}$ *and the* 0*-simplified time-centred attacker strategy* $\boldsymbol{\phi}_{0-stc}$.

Figure 4.2.20 shows our progression of finding tight upper bounds with the lower bound in lemma 4.2.13 by using the upper bound in equation (4.24). Notice that the bounds are tight when m is such that $\rho = 0$ and that for other values of m there is a slight sub-optimality in the attacker using $\boldsymbol{\phi}_{0-stc}$ (as we see later that $\boldsymbol{\pi}_{\text{AdjCombHyb}}$ is optimal).

Having seen an optimal attacker strategy when $\rho = 0$ we are left to find optimal attacker strategies for $\rho = 1, 2, 3$. We find an optimal attacker strategy for $\rho = 2$ and near optimal strategies for $\rho = 1, 3$ by creating bespoke attacker strategies. Due to the way we will adapt the strategies for $\rho = 1, 3$, we present a 'doubled' space-time agent matrix for the 0-simplified time-centred attacker strategy, in order to avoid changing half an agent. From the lower bound given in lemma 4.2.13, viz.

$$V(S^k_n, T, m) \geq \frac{2m}{2(n+k) + m + 2(n-1)},$$

it should be clear why this doubling of the original representation is needed when m is odd, which corresponds to $\rho = 1, 3$. Therefore the goal to is to find a space-time agent matrix using $2(n+k) + m + 2(n-1)$ agents in which any pure patroller can at most catch $2m$ agents. However, we will find near optimal

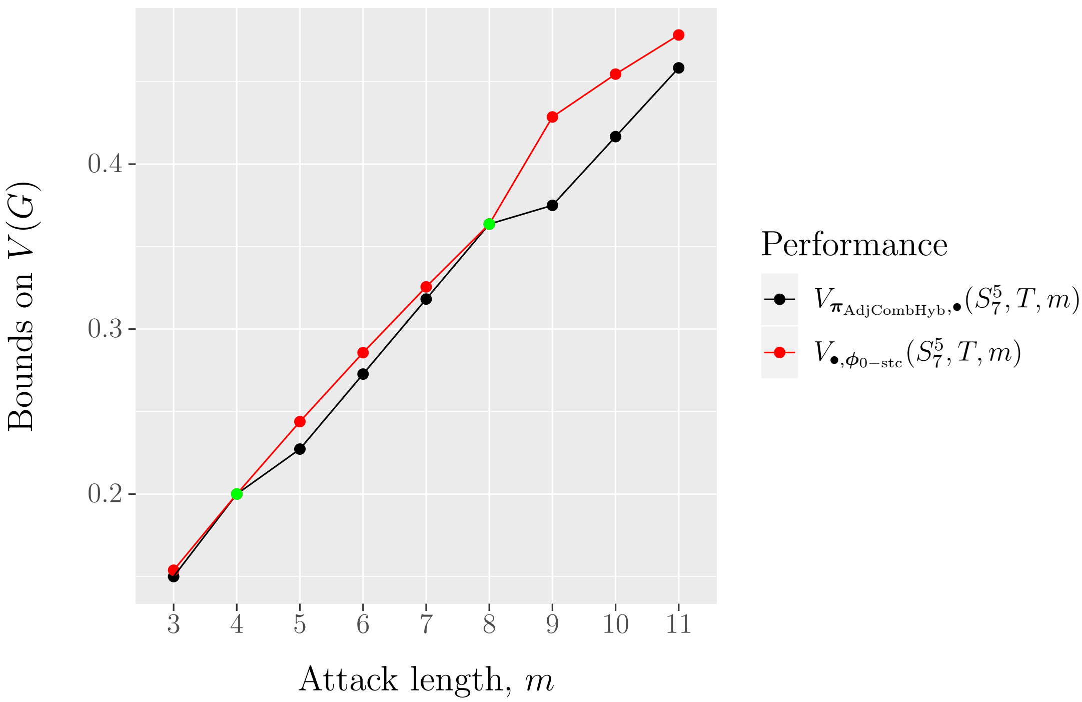


Figure 4.2.20: The graph shows the lower bound on the game $G(S_7^5, 50, m)$ provided by adjusted combinatorial improvement patroller strategy (with middle nodes) $\boldsymbol{\pi}_{\mathrm{AdjCombHyb}}$ in black (which will later see is optimal) and the upper bound provided by the simplified time-centred attacker strategy $\boldsymbol{\phi}_{0-stc}$, in red. Plotted for $m = 3, ..., 11$ with cases of m such that $\rho = 0$ shown in green, for which we know the bounds are tight.

strategies in which any pure patroller can catch at most $2m + 1$ agents. Figure 4.2.21 shows the same attacker strategy as figure 4.2.19, that being $\boldsymbol{\phi}_{0-stc}$, with a ring around a space-time point (j, τ) representing the presence of two agents. Note that probabilities of the strategy $\boldsymbol{\phi}_{0-stc}$ remain the same, as we are doubling the numerator and denominator of each probability of playing (j, τ).

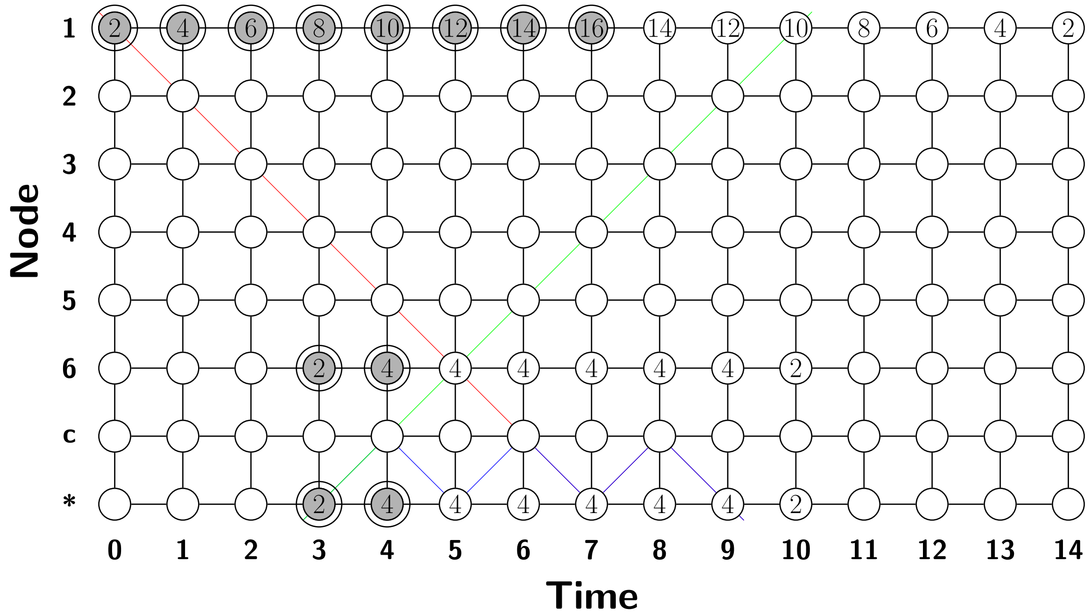


Figure 4.2.21: 'Doubled' space-time agent matrix for the 0-*simplified time-centred* attacker strategy $\boldsymbol{\phi}_{0-stc}$, for the game $G(S_7^5, 20, 8)$. Equivalent in terms of strategy to that shown in figure 4.2.19, with double the number of agents at each space-time point, represented by an additional ring around nodes with double agents.

As each of the remaining $\rho = 1, 2, 3$ require different adaptations to create bespoke attacker strategy, we cover them in the following three cases. We cover them in descending order, $\rho = 3, 2, 1$, due to their creation coming from removing an agent from the previously done case, while ensuring that any pure patroller to catches at least one agent less than before.

Case 1: $\rho = 3$

For the case of $\rho = 3$, we have to remove a single agent from the 'doubled' strategy that would be used for an attack length of $m + 1$ (that is the $\rho = 0$-simplified time-centred attacker strategy for the game with $m+1$). We remove two potential agents at the node 1 which commence at times 0 and $2(k - l^*(m + 1)) + 1$. We also introduce a single potential attack at node $k + 1 - l^*(m + 1)$ starting at time $k - l^*(m + 1)$. This results in the following attacker strategy.

Definition 4.2.18. The 3-*simplified time-centred* attacker strategy $\boldsymbol{\phi}_{3-stc}$ is such that the probability of choosing (j, τ) is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2(n+k)+m+2(n-1)} & \text{for } j = 1, \tau \in \{0, m\}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j = 1, \tau \in \{1, ..., m-1\}, \\ \frac{1}{2(n+k)+m+2(n-1)} & \text{for } j = \frac{m+1}{2} + 1, \tau = \frac{m+1}{2} - 1, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j \in \{k+1, k-1, ..., \frac{m+1}{2} + 2\}, \tau \in \{\frac{m+1}{2} - 1, \frac{m+1}{2}\}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j \in \{*_1, ..., *_{n-1}\}, \tau \in \{\frac{m+1}{2} - 1, \frac{m+1}{2}\}. \end{cases}$$

That is $\boldsymbol{\phi}_{3-stc}$ chooses; with probability

$$\frac{2m}{2(n+k)+m+2(n-1)}$$

to attack node 1, then choosing a commencement time from $\tau = 0, ..., m$ with probability $\frac{1}{2m}$ for $\tau = 0, m$ and $\frac{2}{2m}$ for $\tau = 1, ..., m-1$; or chooses node $*_i$ for $i = 1, ..., n-1$ with probability

$$\frac{4}{2(n+k)+m+2(n-1)}$$

and then with equal probability a commencement time from $\frac{m+1}{2} - 1, \frac{m+1}{2}$; or chooses nodes j for $j = k+1, k-1, ..., \frac{m+1}{2} + 2$ with probability

$$\frac{2}{2(n+k)+m+2(n-1)}$$

and then with equal probability a commencement time from $\frac{m+1}{2} - 1, \frac{m+1}{2}$; or chooses node $\frac{m+1}{2} + 1$ with probability $\frac{1}{2(n+k)+m+2(n-1)}$ and a commencement time of $\frac{m+1}{2} - 1$.

Figure 4.2.22 shows an example of the space-time agent matrix $\boldsymbol{S}^A_{3-stc}$ for $\boldsymbol{\phi}_{3-stc}$, which can be compared to the 'double' space-time agent matrix in figure 4.2.21 in order to see the adaptation/removal of an agent. By removing two agents at the ends of the commencement time distribution and adding agent above the last node along the line which is simplified to a star node we are able to restrict the best pure patroller to only catch $2m + 1$ agents. We formally state this result as achieving the desired upper bound, in the following lemma.

Lemma 4.2.19. *For the game $G(S^k_n, T, m)$, for all $n \geq 3$, $k \geq 1$, for all $m \in M^{S^k_n}_{5,M} \cap \{m : \rho = 3\}$ and for all $T \geq 2m$ we have*

$$V(S^k_n, T, m) \leq \frac{2m+1}{2(n+k)+m+2(n-1)},$$

achieved by the 3-simplified time-centred attacker strategy $\boldsymbol{\phi}_{3-stc}$.

The proof of lemma 4.2.19, is left to appendix B.3.1, as it follows by computing the performance of the strategy $\boldsymbol{\phi}_{3-stc}$. The upper bound in lemma 4.2.19 is not

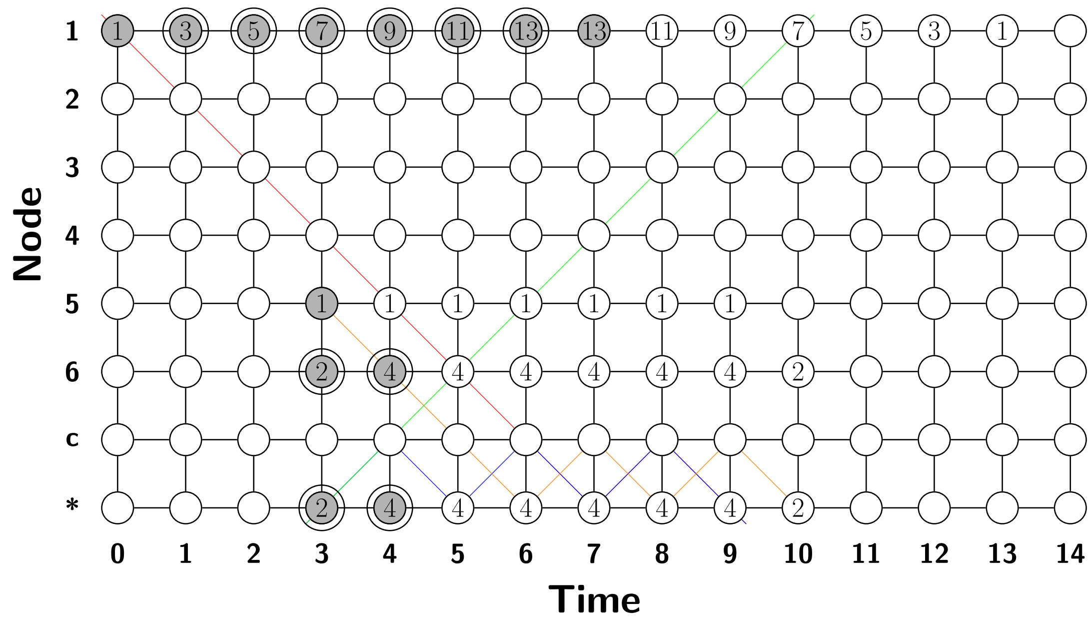


Figure 4.2.22: Space-time agent matrix $\boldsymbol{S}^{A}_{3-stc}$ for the 3-simplified time-centred attacker strategy $\boldsymbol{\phi}_{3-stc}$ for the game $G(S^5_7, 15, 7)$. Four example pure patrollers are shown in red, orange, green and blue each catching at most 15 out of 43 agents.

tight with the lower bound in equation (4.21), but does provide a very close upper value. Therefore while $\boldsymbol{\phi}_{3-stc}$ may not be optimal it is near optimal as

$$\frac{2m}{2(n+k)+m+2(n-1)} \leq V(S^k_n, T, m) \leq \frac{2m+1}{2(n+k)+m+2(n-1)}.$$

Case 2: $\rho = 2$

For the case of $\rho = 2$, we have to remove a single agent from the 'single' strategy that would be used for an attack length of $m+2$ (that is the $\rho = 0$-simplified time-centred attacker strategy for the game with $m+2$). We first re-allocate potential agents at node 1 which start at odd times, t, to start at $t-1$, the prior even times. We also remove the potential attacks at $t = 2(k - l^*(m+2))$ and $t = 2(k - l^*(m+2)) + 1$ at node 1. We then have to re-centralise the potential agents commencing at nodes $k+2-2l^*(m+2), ..., k$ and $*_i$ for $i = 1, ..., n-1$ and re-allocate even commencement times, to commence at times $k - 2l^*(m+2) - 2$,$k - 2l^*(m+2)$. Finally as two potential agents are removed we need to introduce a single potential agents, we do so at node $k+1-2l^*(m+2)$ commencing at time $k - 2l^*(m+2) - 1$. This results in the following attacker strategy.

Definition 4.2.20. The $\rho = 2$-*simplified time-centred* attacker strategy $\boldsymbol{\phi}_{2-stc}$ is such that the probability of choosing (j, τ) is

$$\varphi_{j,\tau} = \begin{cases} \frac{4}{2(n+k)+m+2(n-1)} & \text{for } j = 1, \tau \in \{0, 2, ..., m-2\}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j = \frac{m+4}{2}, \tau = \frac{m-2}{2}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j \in \{k+1, k-1, ..., \frac{m+4}{2}+1\}, \tau \in \{\frac{m-4}{2}, \frac{m}{2}\}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j \in \{*_1, ..., *_{n-1}\}, \tau \in \{\frac{m-4}{2}, \frac{m}{2}\}. \end{cases}$$

That is $\boldsymbol{\phi}_{2-stc}$ chooses; with probability

$$\frac{2m}{2(n+k)+m+2(n-1)}$$

to attack node 1, then choosing a commencement time from $\tau = 0, 2, ..., m-2$ with equal probability ; or chooses node $*_i$ for $i = 1, ..., n-1$ with probability

$$\frac{4}{2(n+k)+m+2(n-1)}$$

and then with equal probability choosing a commencement time from $\frac{m-4}{2}, \frac{m}{2}$; or chooses node j for $j = k+1, k-1, ..., \frac{m+4}{2}+1$ with probability

$$\frac{4}{2(n+k)+m+2(n-1)}$$

and then with equal probability a commencement time from $\frac{m-4}{2}, \frac{m}{2}$; or chooses node $\frac{m+4}{2}$ with probability

$$\frac{2}{2(n+k)+m+2(n-1)}$$

and a commencement time of $\frac{m-2}{2}$.

Figure 4.2.23 shows an example of the space-time agent matrix $\boldsymbol{S}^A_{2-stc}$ for $\boldsymbol{\phi}_{2-stc}$, which can be compared to the 'single' space-time agent matrix in figure 4.2.19 in order to see the adaptation/removal of an agent. Unlike the case of $\rho = 3$, as m is even, ensuring even commencement times for the agents placed at node 1 allow us to get an attacker strategy where the best pure patroller is only able to catch m agents. We formally state this result as achieving the desired bound, as in the following lemma.

Lemma 4.2.21. *For the game $G(S^k_n, T, m)$, for all $n \geq 3$, $k \geq 1$, for all $m \in M^{S^k_n}_{5,1} \cap \{m : \rho = 2\}$ and for all $T \geq 2m-3$ we have*

$$V(Q, T, m) \leq \frac{2m}{2(n+k)+m+2(n-1)},$$

achieved by the 2-simplified time-centred attacker strategy $\boldsymbol{\phi}_{2-stc}$.

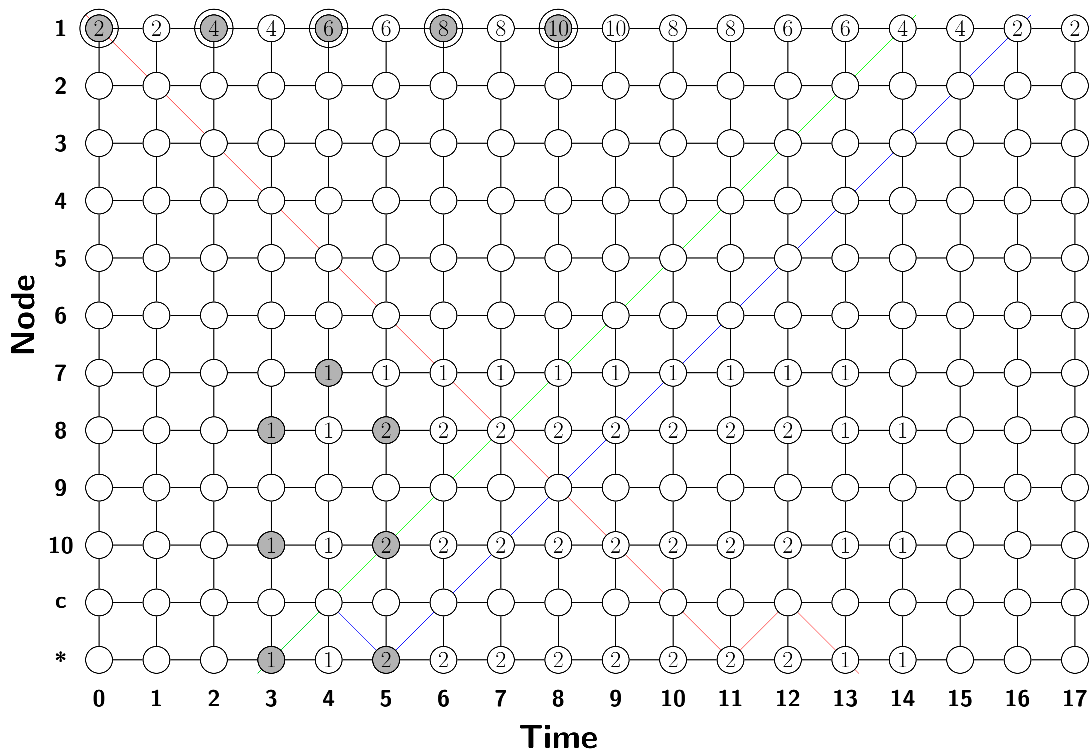


Figure 4.2.23: Space-time agent matrix $\boldsymbol{S}^{A}_{2-stc}$ for the 2-simplified time-centred attacker strategy $\boldsymbol{\phi}_{2-stc}$ for the game $G(S^9_7, 18, 10)$. Three example pure patrollers are shown in red, green and blue each catching 10 out of 27.

The proof of lemma 4.2.21, is left to appendix B.3.2, as it follows by computing the performance of $\boldsymbol{\phi}_{2-stc}$. The upper bound in lemma 4.2.23 is tight with the lower bound in equation (4.21) and therefore $\boldsymbol{\phi}_{2-stc}$ is optimal.

Case 3: $\rho = 1$

For the case of $\rho = 1$, while it is possible to imagine it as a removal and re-allocation of agents from the attacker strategy when the attack length is $m + 3$ (that is the $\rho = 0$-simplified time-centred attacker strategy when $m + 3$), it is much easier to see the addition and re-allocation of potential agents when the attack length is $m - 1$ (that is the $\rho = 0$-simplified time-centred attacker strategy when $m + 3$). To adapt the 'doubled' attacker strategy with $m - 1$, we first time shift the strategy forward by one time unit in order to allow for a new single potential attack to be introduced at node 1 commencing at time 0 and then the re-allocation of a single potential attack at node $k - 2l^*(m - 1) + 2$ commencing at time $k - 2l^*(m - 1) + 1$ (originally at $k - 2l^*(m - 1)$ before the time shift) to node 1 commencing at time $2(k - 2l^*(m - 1)) + 3$. This results in the following attacker strategy.

Definition 4.2.22. The 1-simplified time-centred attack strategy $\boldsymbol{\phi}_{1-stc}$ is such that the probability of choosing (j, τ) is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2(n+k)+m+2(n-1)} & \text{for } j = 1, \tau \in \{0, m\}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j = 1, \tau \in \{1, ..., m-1\}, \\ \frac{1}{2(n+k)+m+2(n-1)} & \text{for } j = \frac{m+3}{2}, \tau \in \{\frac{m-1}{2}, \frac{m-1}{2}+1\}, \\ \frac{1}{2(n+k)+m+2(n-1)} & \text{for } j = 1 + \frac{m+3}{2}, \tau = \frac{m-1}{2}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j \in \{k+1, k-1, ..., \frac{m+3}{2}+2, \}\tau \in \{\frac{m-1}{2}, \frac{m-1}{2}+1\}, \\ \frac{2}{2(n+k)+m+2(n-1)} & \text{for } j \in \{*_1, ..., *_{n-1}\}, \tau \in \{\frac{m-1}{2}, \frac{m-1}{2}+1\}. \end{cases}$$

That is $\boldsymbol{\phi}_{1-stc}$ chooses; with probability

$$\frac{2m}{2(n+k)+m+2(n-1)}$$

to attack node 1, then choosing a commencement time from $\tau = 0, 1, ..., m$ with probability $\frac{1}{2m}$ for $\tau = 0, m$ and $\frac{2}{2m}$ for $\tau = 1, ..., m-1$; or chooses node $*_i$ for $i = 1, ..., n-1$ with probability

$$\frac{4}{2(n+k)+m+2(n-1)}$$

and then with equal probability choosing a commencement time from $\frac{m-1}{2}, \frac{m+1}{2}$; or chooses node j for $j = k+1, k-1, ..., 2+\frac{m+3}{2}$ with probability

$$\frac{4}{2(n+k)+m+2(n-1)}$$

and then with equal probability a commencement time from $\frac{m-1}{2}, \frac{m+1}{2}$; or chooses node $1+\frac{m+3}{2}$ with probability

$$\frac{1}{2(n+k)+m+2(n-1)}$$

and a commencement time of $\frac{m-1}{2}$; or chooses node $\frac{m+3}{2}$ with probability

$$\frac{2}{2(n+k)+m+2(n-1)}$$

and then with equal probability chooses a commencement time from $\frac{m-1}{2}, \frac{m+1}{2}$.

Figure 4.2.24, shows an example of the space-time agent matrix $\boldsymbol{S}^A_{1-stc}$ for $\boldsymbol{\phi}_{1-stc}$, which can be compared to the 'doubled' space-time agent matrix in figure 4.2.21 in order to see the adaptation/removal of three agents. In addition we can see a comparison to the space-time agent matrix in figure 4.2.22 in which two agents are removed, by altering the closest agents along the line section to node 1. This adaptation, like the others, means the best pure patroller is only able to catch $2m + 1$ agents. We formally state this result as achieving the desired bound, as in the following lemma.

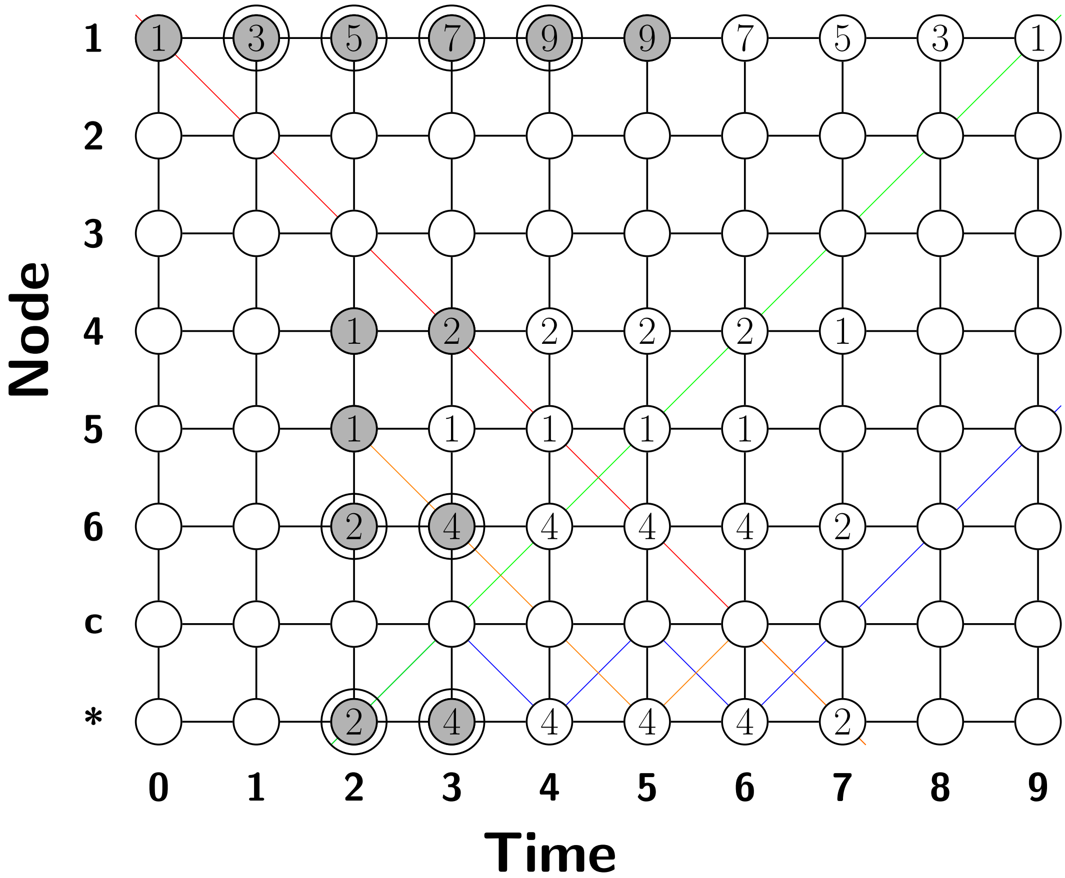


Figure 4.2.24: Space-time agent matrix $\boldsymbol{S}^{A}_{1-stc}$ for the 1-simplified time-centred attacker strategy $\boldsymbol{\phi}_{1-stc}$ for the game $G(S^5_7, 10, 5)$. Three example pure patrollers are shown in red, orange, green and blue each catching 11 out of 41 agents.

Lemma 4.2.23. *For the game $G(S_n^k, T, m)$, for all $n \geq 3$, $k \geq 1$, for all $m \in M_{5,1}^{S_n^k} \cap \{m : \rho = 1\}$ and for all $T \geq 2m$ we have*

$$V(S_n^k, T, m) \leq \frac{2m+1}{2(n+k)+m+2(n-1)},$$

achieved by the 1-simplified time-centred attacker strategy $\boldsymbol{\phi}_{1-stc}$.

The proof of lemma 4.2.23, is left to appendix B.3.3, as it follows by computing the performance of the strategy $\boldsymbol{\phi}_{1-stc}$. The upper bound in lemma 4.2.23 is not tight with the lower bound in equation (4.21), but does provide a very close upper value. Therefore while $\boldsymbol{\phi}_{1-stc}$ may not be optimal it is near optimal as

$$\frac{2m}{2(n+k)+m+2(n-1)} \leq V(S_n^k, T, m) \leq \frac{2m+1}{2(n+k)+m+2(n-1)}.$$

4.2.11 Conclusion on elongated star graphs

In this section, we collate our results obtained throughout our work on the game $G(S_n^k, T, m)$ into the following theorem.

Theorem 4.2.24. *For the game $G(S_n^k, T, m)$, the value of the game is given by, for all $n \geq 3$, for all $k \geq 1$,*

- *for all $T \geq 1$ and $m \in M_0^{S_n^k} = \{m : m = 1\}$,*

$$V(S_n^k, T, 1) = \frac{1}{n+k+1},$$

  achieved by the choose and wait patroller strategy $\boldsymbol{\pi}_{cw}$ and the position-uniform attacker strategy $\boldsymbol{\phi}_{pu}$;

- *for all $T \geq m$ and $m \in M_1^{S_n^k} = \{m : m \geq 2(n+k)\}$,*

$$V(S_n^k, T, m) = 1,$$

  achieved by the minimal full-node cycle patroller strategy $W_{MFNC}^{S_n^k}$ (or $\boldsymbol{\pi}_{RMFNC}^{S_n^k}$) and any attacker strategy;

- *for $T \geq m$ and $m \in M_2^{S_n^k} = \{m : 2(k+1) \leq m \leq 2(n+k)\}$,*

$$V(S_n^k, T, m) = \frac{m}{2(n+k)},$$

  achieved by the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{S_n^k}$ and the time-centred attacker strategy $\boldsymbol{\phi}_{tc}$;

- *for* $T \geq 2$ *and* $m \in M_3^{S_n^k} = \{m : m = 2\}$,

$$V(S_n^k, T, 2) = \frac{1}{n + \lceil \frac{k}{2} \rceil},$$

achieved by the covering patroller strategy $\boldsymbol{\pi}_{Cov}$ *and the independent attacker strategy* $\boldsymbol{\phi}_{Ind}$*;*

- *for* $T \geq 1$ *and* $m \in M_4^{S_n^k}\{m : m > 2n, m < 2(k+1)\}$,

  – *if* $m \in M_{4,0}^{S_n^k} \equiv M_4^{S_n^k} \cap \{m \mid \hat{m} \geq \frac{n+k-1}{2}\}$,

$$V(S_n^k, T, m) \geq \frac{1}{2},$$

  achieved by the simple improvement patroller strategy $\boldsymbol{\pi}_{SimpHyb}$ *(with no middle nodes) . Moreover* $V(S_3^k, T, m) = \frac{1}{2}$*, by the time-limited diametric attacker strategy* $\boldsymbol{\phi}_{tdi}$*;*

  – *if* $m \in M_{4,0}^{S_n^k} \equiv= M_4^{S_n^k} \cap \{m \mid \hat{m} < \frac{n+k-1}{2}\}$,

$$V(S_n^k, T, m) \geq \frac{m}{m+n+k},$$

  achieved by the simple improvement patroller strategy $\boldsymbol{\pi}_{SimpHyb}$ *(with middle nodes);*

- *for* $T \geq 2m$ *and* $m \in M_5^{S_n^k} = \{m : 3 \leq m \leq 2n, m < 2(k+1)\}$,

  – *if* $m \in M_{5,0}^{S_n^k} \equiv \{2k, 2k+1\}$,

$$V(S_n^k, T, m) = \frac{m}{m + 2(n-1)},$$

  achieved by the adjusted combinatorial improvement patroller strategy $\boldsymbol{\pi}_{AdjCombHyb}$ *(with no middle nodes) and the reduced time-centred attacker strategy* $\boldsymbol{\phi}_{rtc}$*;*

  – *if* $m \in M_{5,1}^{S_n^k} \equiv M_5^{S_n^k} \cap \{m \mid \hat{m} < k\}$,

    ■ *and* $\rho \in \{0, 2\}$ *then*

$$V(S_n^k, T, m) = \frac{2m}{2(n+k) + m + 2(n-1)},$$

    ■ *and* $\rho \in \{1, 3\}$ *then*

$$\frac{2m}{2(n+k) + m + 2(n-1)} \leq V(S_n^k, T, m) \leq \frac{2m+1}{2(n+k) + m + 2(n-1)},$$

  achieved by the adjusted combinatorial improvement patroller strategy $\boldsymbol{\pi}_{AdjCombHyb}$ *(with middle nodes) and the* ρ*-simplified time-centred attacker strategy* $\boldsymbol{\phi}_{\rho-stc}$*. Where* $\rho \equiv m - 2k - 2 \mod 4$ *determines the exact strategy the attacker takes (see sections 4.2.9 and 4.2.10 for details);*

Theorem 4.2.24 provides optimal solutions to the vast majority of patrolling games played on the elongated star graph. We note that we require a game length of at least $2m$ for the region of attack lengths $M_5^{S_n^k}$ in order to be able to use the attacker strategies. Furthermore, we note that we were only able to find near optimal solutions when $\rho \in \{1, 3\}$. While we did not find analytical bespoke attacker strategies which match the upper bound when $m \in M_4^{S_n^k}$, we provide examples of games in which these were found to be tight. We highlight this now as it is future work that can be investigated. The partial solution and examples in section 4.2.7 may help to provide an insight on how to construct more optimal strategies for $M_4^{S_n^k}$.

Theorem 4.2.24 concludes our work on $G(S_n^k, T, m)$. In the following section we consider $S_n^{\boldsymbol{k}}$ where more star nodes undergo node splitting ending a distance k_i+1 from the centre for each star node $*_i$ for all $i \in \{1, ..., n-1\}$. We are able to extend some of our strategies from theorem 4.2.24 to obtain the solution to the game $G(S_n^{\boldsymbol{k}}, T, m)$.

4.3 Generalised star graph

To further extend the structure of elongated star graphs $S_n^k \in \mathcal{SE}$, which model a border with a star structure at one end, we introduce the *generalised star* graph. The generalised star graph $S_n^{\boldsymbol{k}}$, for some $\boldsymbol{k} \in \mathbb{N}^n$ for some $n \in \mathbb{N}$, is formed by performing k_i node-splittings on the node $*_i$ for all $i \in \{1, ..., n\}$, where k_i is the i^{th} element of $\boldsymbol{k}$. This node-splitting is done in a similar fashion as for the elongated star graph, resulting in node $*_i$ being a distance of $k_i + 1$ from the centre c. The generalised star graph models multiple borders, of varying length, connected to each other by a central location. In addition it models important locations, at various distances from a central location. In this section we provide solutions to the generalised star graph for attack length regions $M_i^{S_n^{\boldsymbol{k}}}$ for $i = 0, 1, 2, 3$, defined analogous to those for the elongated star graph ($M_i^{S_n^k}$ for $i = 0, 1, 2, 3$). In addition we provide ideas for solutions for the final regions (analogous to $M_4^{S_n^k}$ and $M_5^{S_n^k}$)

Definition 4.3.1. The *generalised star* (graph) is a graph $S_n^{\boldsymbol{k}} = (N, E)$ where $\boldsymbol{k} \in \mathbb{N}^n$ is the vector of *branch extensions*. The set of nodes is given by

$$N = \{c\} \bigcup_{r=1}^{n} \{*_{i,1}, ...*_{i,k_r+1}\},$$

and the set of edges is given by

$$E = \{(c, *_{1,1}), (c, *_{2,1}), ..., (c, *_{n,1})\} \bigcup_{r=1}^{n} \{(*_{i,1}, *_{i,2}), ..., (*_{i,k_i}, *_{i,k_i+1})\}.$$

We say k_i is the i^{th} *branch extension* and denote the set of all generalised star graphs by $\mathcal{SG}$.

Figure 4.3.1 shows the generalised star graph $S_4^{4,3,1,0}$. The labelling of nodes along branches is done such that $*_{i,r}$ is the r^{th} node along the i^{th} branch for all $r \in \{1, ..., k_i + 1\}$ and for all $i \in \{1, ..., n\}$. For example the node $*_{2,3}$ is the 3^{rd} node along the 2^{nd} branch from the centre c.

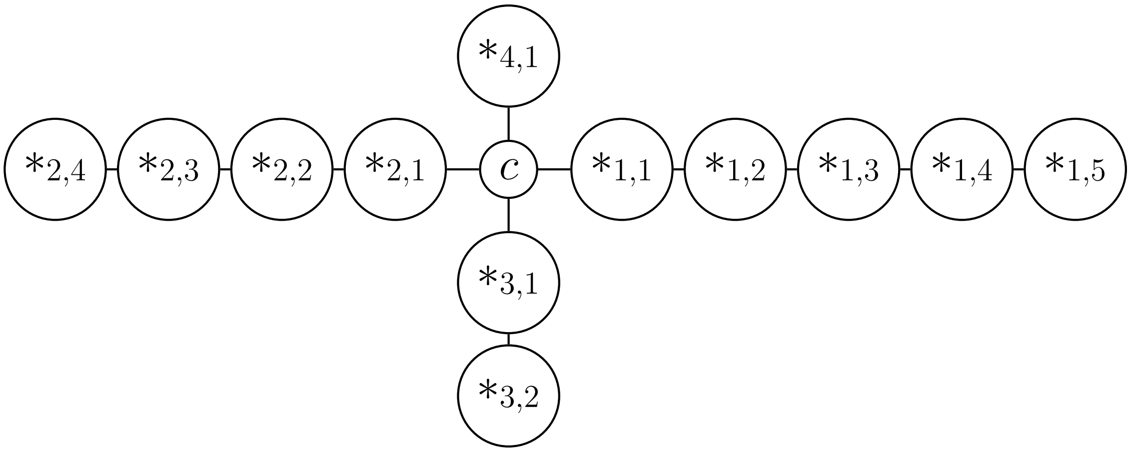


Figure 4.3.1: The generalised star graph $S_4^{4,3,1,0} \in \mathcal{SG}$.

We immediately note that $\mathcal{SE} \subset \mathcal{SG}$, under a relabelling (isomorphism) and that if $\boldsymbol{k}$ has only one non-zero element, then the generalised star graph is an elongated star graph. Further if $n \in \{1, 2\}$ then the generalised star graph is a line graph. Hence we assume that $\boldsymbol{k}$ has at least two non-zero elements and that $n \geq 3$. We will also assume that the branch extension vector $\boldsymbol{k}$ is in descending order, so that $k_{i+1} \leq k_i$ for all $i \in \{1, ..., n-1\}$. In addition we omit any k_i which are zero in our notation to avoid excessive numbering, for example the graph in figure 4.3.1 could be written as $S_4^{4,3,1} \equiv S_4^{4,3,1,0}$. To avoid further notational clutter, we define some summary notation for generalised star graphs.

Definition 4.3.2. For the generalised star graph $S_n^{\boldsymbol{k}} \in \mathcal{SG}$, with branch extensions $\boldsymbol{k} \in \mathbb{N}^n$ let:

- The sum of branch extensions be denoted by $k_{\text{sum}} = \sum_{i=1}^{n} k_i$.
- The maximum branch extension be denoted by $k_{\max} = \max_{i=1,...,n} k_i$.

In order to solve the game $G(S_n^{\boldsymbol{k}}, T, m)$ for various parameters we decompose the set of attack lengths into regions for which the optimal strategies will differ. This decomposition is done in analogous fashion to that as done for the elongated star graph. We decompose the set of attack lengths into the regions:

- $M_0^{S_n^{\boldsymbol{k}}} = \{m : m = 1\}$,
- $M_1^{S_n^{\boldsymbol{k}}} = \{m : m \geq 2(n + k_{\text{sum}})\}$,
- $M_2^{S_n^{\boldsymbol{k}}} = \{m : 2(k_{\max} + 1) \leq m < 2(n + k_{\text{sum}})\}$,

- $M_3^{S_n^{\boldsymbol{k}}} = \{m : m = 2\}$,
- $M_4^{S_n^{\boldsymbol{k}}} = \{m : 2 < m < 2(k_{\max} + 1)\}$.

We provide optimal solutions and the value of the game $G(S_n^{\boldsymbol{k}}, T, m)$ when $m \in M_i^{S_n^{\boldsymbol{k}}}$ for $i = 0, 1, 2, 3$, but not for $i = 4$. When $m \in M_4^{S_n^{\boldsymbol{k}}}$ bespoke solutions for the attacker strategy are required, as in $M_4^{S_n^k}$ and $M_5^{S_n^k}$. Therefore due to the complexity of adapting such solutions, we do not focus on solutions in the region of $m \in M_4^{S_n^{\boldsymbol{k}}}$, instead we provide ideas for solutions. For the scenario where $S_n^{\boldsymbol{k}}$ is used to model multiple cities connected by a central hub, we can use work done on patrolling games with edge distances in chapter 6, section 6.1, to find optimal strategies by 'ignoring' all but the centre and leaf nodes, allowing us to get strategies for $m \in M_4^{S_n^{\boldsymbol{k}}}$ without the need for bespoke attacker strategies. While these games are not mathematically equivalent, as we cannot ignore nodes in patrolling games, ignoring nodes can ease the complexity and allow for easier results to be given when it is sensible to ignore locations which are otherwise protected.

As all patrolling games with $m = 1$ are solved by lemma 2.3.26, we know optimal strategies and the value of the game when $m = 1$ ($m \in M_0^{S_n^{\boldsymbol{k}}}$). That is for all $n \geq 1$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for $T \geq 1$ the value of the game is

$$V(S_n^{\boldsymbol{k}}, T, 1) = \frac{1}{n + 1 + k_{\text{sum}}}.$$

In the following sections we will consider $m \in M_i^{S_n^{\boldsymbol{k}}}$ for $i \in \{1, ..., 4\}$ in turn.

4.3.1 Solution when $m \in M_1^{S_n^{\boldsymbol{k}}}$

For $m \in M_1^{S_n^{\boldsymbol{k}}}$ we begin by looking for a minimal full-node cycle of the generalised star graph. This is done analogous to the case of $m \in M_1^{S_n^k}$. An easily identifiable minimal full-node cycle is one which starts at the centre and then goes along each branch, reaching $*_{i,k_i+1}$ and then heading beck to the centre and repeating for all $i \in \{1, ..., n\}$. So

$$W_{\text{MFNC}}^{S_n^{\boldsymbol{k}}} = \{c, *_{1,1}, ..., *_{1,k_1}, ..., *_{1,1}, c, *_{2,1}, ..., *_{2,k_2}, ..., *_{2,1}, c, ... *_{n,1}\},$$

is a minimal full-node cycle of length $2(n + k_{\text{sum}})$ and so for $m \in M_1^{S_n^{\boldsymbol{k}}}$ it visits every node at least every m time units and is therefore intercepting. As $W_{\text{MFNC}}^{S_n^{\boldsymbol{k}}}$ is intercepting we know that the game is a guaranteed win for the patroller using the pure strategy which repeats it for the time-horizon, so we know the value of the game is 1 as using this pure strategy gives a lower bound of 1 (from having $\mathcal{C}_{S_n^{\boldsymbol{k}},T,m} = 1$ in lemma 2.3.12 or theorem 3.3.26) and the trivial upper bound for any patrolling is 1. Therefore we arrive at the following lemma.

Lemma 4.3.3. *For the game $G(S_n^{\boldsymbol{k}}, T, m)$ for all $n \geq 3$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $m \in M_1^{S_n^{\boldsymbol{k}}}$ and for all $T \geq m$ we have*

$$V(S_n^{\boldsymbol{k}}, T, m) = 1,$$

achieved by a minimal full-node cycle patroller strategy $W_{MFNC}^{S_n^{\boldsymbol{k}}}$ and any attacker strategy.

4.3.2 Solution when $m \in M_2^{S_n^{\boldsymbol{k}}}$

We can use $W_{\text{MFNC}}^{S_n^{\boldsymbol{k}}}$, when $m \in M_2^{S_n^{\boldsymbol{k}}}$, by utilising the random minimal full-node cycle strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^{\boldsymbol{k}}}$, using $W_{\text{MFNC}}^{S_n^{\boldsymbol{k}}}$, to get the lower bound as given in theorem 3.3.26. That is for all $n \geq 3$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $m \geq 1$ and for $T \geq m$ we have

$$V(S_n^{\boldsymbol{k}}, T, m) \geq \frac{m}{2(n + k_{\text{sum}})}, \tag{4.25}$$

achieved by the random minimal full-node cycle strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^{\boldsymbol{k}}}$.

As always with a random full-node cycle, we note that random means starting in the cycle $W_{\text{MFNC}}^{S_n^{\boldsymbol{k}}}$ uniformly and not uniformly amongst nodes of the graph. To get a tight upper bound with the lower bound in equation 4.25, we use a similar idea to the time-centred attacker strategy used on the elongated star graph. That is we use a weighting proportional to the distance from c for each $*_{i,k_i+1}$ for $i \in \{1, ..., n\}$. Then the commencement times are distributed and centred according to this weighting. This gives us the following *type-centred* attacker strategy.

Definition 4.3.4. The *type-centred* attacker strategy $\boldsymbol{\phi}_{\text{type}}$ is such that the probability of choosing to play the pure attack $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2(n+k_{\text{sum}})} & \text{for } j = *_{i,k_i+1}, \tau \in T_i \text{ for some } i \in \{1, ..., n\}, \\ 0 & \text{otherwise,} \end{cases}$$

where $T_i = \{k_{\max} - k_i, ..., k_{\max} + k_i + 1\}$.

That is the type-centred attacker strategy $\boldsymbol{\phi}_{\text{type}}$ chooses with probability

$$\frac{2(k_i + 1)}{2(n + k_{\text{sum}})}$$

to attack node $*_{i,k_i+1}$, then choosing a commencement time τ from the set

$$T_i = \{k_{\max} - k_i, ..., k_{\max} + k_i + 1\}$$

each with equal probability, for any branch $i \in \{1, ..., n\}$. As with all attack strategies which distribute the potential attacks in the commencement time, the typed type-centred attacker strategy requires a condition on the game length in order to be feasible. Namely $T \geq 2k_{\max} + m + 1$ is required so that $\boldsymbol{\phi}_{\text{type}} \in \Phi(S_n^{\boldsymbol{k}}, T, m)$. As with the time-centred attacker strategy, the idea of distributing the weighting in commencement time is done so that any pure patroller moving between attacked nodes arrives when all potential attacks have already begun. Therefore we get an analogous result to lemma 4.2.6 in the following lemma.

Lemma 4.3.5. *For a game $G(S_n^{\boldsymbol{k}}, T, m)$ for all $n \geq 1$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $m \geq 1$ and for all $T \geq 2k_{max} + m + 1$ we have*

$$V(S_n^{\boldsymbol{k}}, T, m) \leq \max\left(\frac{k_{max}+1}{n+k_{sum}}, \frac{m}{2(n+k_{sum})}\right),$$

achieved by the type-centred attacker strategy $\boldsymbol{\phi}_{type}$.

The proof of lemma 4.3.5 follows by evaluating $V_{\bullet,\boldsymbol{\phi}_{\text{type}}}(S_n^{\boldsymbol{k}}, T, m)$, the performance of $\boldsymbol{\phi}_{\text{type}}$, by relying on the work done in section 3.2.2. This is done in a similar, but more general fashion, than the proof of lemma 4.2.6.

Proof. We aim to calculate $V_{\bullet,\boldsymbol{\phi}_{\text{type}}}(S_n^{\boldsymbol{k}}, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet,\boldsymbol{\phi}_{\text{type}}}(S_n^{\boldsymbol{k}}, T, m) = V_{\bullet,\boldsymbol{\phi}_{\text{type}}}(S_n^{\boldsymbol{k}}, 2k_{\max}+m+1, m) = \max_{W \in \mathcal{W}(S_n^{\boldsymbol{k}}, 2k_{\max}+m+1, m)} P(W, \boldsymbol{\phi}_{\text{type}}),$$

so we only need to consider pure walks for $2k_{\max} + m + 1$ units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet,\boldsymbol{\phi}_{\text{type}}}(S_n^{\boldsymbol{k}}, 2k_{\max}+m+1, m) = \max_{\omega \in \Omega'''(S_n^{\boldsymbol{k}}, 2k_{\max}+m+1, m)} P(\omega, \boldsymbol{\phi}_{\text{type}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_n^{\boldsymbol{k}}, 2k_{\max}+1+m, m)$. That is move-wait walks such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ for which the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$, where $N_A = \{*_{r,k_r+1} \mid r \in \{1, ..., n\}\}$,
- $\nu_i \in \{k_{\max} - k_r, ..., k_{\max} + k_r + 1\}$ when $j_i = *_{r,k_r+1}$ for any $i \in \{1, ..., q\}$ and
- $\nu_1 + \sum_{i=1}^{q-1} (d(j_i, j_{i+1}, N_A) + \nu_{i+1}) \equiv t_q + \nu_q = 2k + m.$

That is a move-wait walk such that nodes belong to those which have a non-zero probability of catching the attacker at if travelled to, with no waiting aside from at the initial node and that the arrival at the final node plus the final waiting match the end of the time-horizon.

For any such walk $\omega \in \Omega'''(S_n^{\boldsymbol{k}}, 2k_{\max}+m+1, m)$ let r_i be such that $j_i = *_{r_i, k_{r_i}+1}$. Then the payoff is given by

$$\begin{aligned}
P(\omega, \boldsymbol{\phi}_{\text{type}}) &= \sum_{i=1}^{q} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_1,t} \\
&= \sum_{t=k_{\max}-k_{r_1}}^{\nu_1} \varphi_{j_i,t} + \sum_{i=2}^{q} \sum_{t=n_i(\omega)}^{k_{\max}+k_{r_i}+1} \varphi_{j_i,t}. \\
&= \sum_{t=k_{\max}-k_{r_1}}^{\min(\nu_1, k_{\max}+k_{r_1}+1)} \frac{1}{2(n+k_{\text{sum}})} + \sum_{i=2}^{q} \sum_{t=n_i'(\omega)}^{k_{\max}+k_{r_i}+1} \frac{1}{2(n+k_{\text{sum}})}. \\
&\leq \sum_{t=k_{\max}-k_{r_1}}^{\min(\nu_1, k_{\max}+k_{r_1}+1)} \frac{1}{2(n+k_{\text{sum}})} + \sum_{i=2}^{q} \sum_{t=n_i''(\omega)}^{k_{\max}+k_{r_i}+1} \frac{1}{2(n+k_{\text{sum}})}. \qquad (4.26)
\end{aligned}$$

Where $n_i(\omega) = \max(0, l_i(\omega)+1, t_i(\omega)-m+1)$, $n_i'(\omega) = \max(k_{\max}-k_{r_i}, l_i(\omega)+1, t_i(\omega)-m+1)$ and $n_i''(\omega) = \max(k_{\max}-k_{r_i}, t_i(\omega)-m+1)$ in equation (4.26). Essentially the inequality follows by ignoring when a node was last visited.

For any $i' \in \{2, ..., q\}$ such that $n_{i'}''(\omega) = t_{i'}(\omega) - m + 1$ we have for all $i \in \{i'+1, ..., q\}$ that $k_{\max} + k_{r_i} + 1 > n_i''(\omega)$ as $t_{i'+1}(\omega) = t_{i'}(\omega) + k_{r_{i'}} + 2 + k_{r_{i'+1}} > k_{\max} + k_{r_{i'+1}} + 1$. Equation (4.26) therefore becomes

$$\begin{aligned}
P(\omega, \boldsymbol{\phi}_{\text{type}}) &\leq \sum_{t=k_{\max}-k_{r_1}}^{\min(\nu_1, k_{\max}+k_{r_1}+1)} \frac{1}{2(n+k_{\text{sum}})} + \sum_{i=2}^{q} \sum_{t=n_i''(\omega)}^{k_{\max}+k_{r_i}+1} \frac{1}{2(n+k_{\text{sum}})}. \\
&= \frac{\min(\nu_1+1-k_{\max}+k_{r_1}, 2k_{r_1}+2)}{2(n+k_{\text{sum}})} + \frac{\sum_{i=2}^{i'-1} 2(k_{r_i+1})}{2(n+k_{\text{sum}})} \\
&\quad + \frac{\max(k_{\max}+k_{r_{i'}}+2-t_{i'}(\omega)+m-1, 0)}{2(n+k_{\text{sum}})} \\
&= \frac{\min(\nu_1+1-k_{\max}+k_{r_1}, 2k_{r_1}+2) + \max(k_{\max}-\nu_1-k_{r_1}+m-1, 0)}{2(n+k_{\text{sum}})}
\end{aligned} \tag{4.27}$$

From equation (4.27) it is clear that in order to maximize the payoff for the walk ω it should have $\nu_1 = k_{\max} + k_{r_1} + 1$ and hence we get that

$$P(\omega, \boldsymbol{\phi}_{\text{type}}) \leq \frac{\max(2(k_{r_1}+1), m)}{2(n+k_{\text{sum}})}.$$

So the best r_1 is such that $k_{r_1} = k_{\max}$ and therefore we have

$$V_{\bullet, \boldsymbol{\phi}_{\text{type}}}(S_n^{\boldsymbol{k}}, 2k_{\max}+m+1, m) \leq \max\left(\frac{k_{\max}+1}{n+k_{\text{sum}}}, \frac{m}{2(n+k_{\text{sum}})}\right). \tag{4.28}$$

The upper bound on the performance of $\boldsymbol{\phi}_{\text{type}}$, as in equation (4.28), gives

$$V(S_n^{\boldsymbol{k}}, T, m) \leq \max\left(\frac{k_{\max}+1}{n+k_{\text{sum}}}, \frac{m}{2(n+k_{\text{sum}})}\right).$$

□

As the lower bound in equation (4.25) and the upper bound in lemma 4.3.5 are tight we arrive at the following lemma.

Lemma 4.3.6. *For a game $G(S_n^{\boldsymbol{k}}, T, m)$ for all $n \geq 1$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $m \in M_2^{S_n^{\boldsymbol{k}}}$ and for all $T \geq 2k_{max} + m + 1$ we have*

$$V(S_n^{\boldsymbol{k}}, T, m) = \frac{m}{2(n + k_{sum})},$$

achieved by a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{S_n^{\boldsymbol{k}}}$ and the type-centred attacker strategy $\boldsymbol{\phi}_{type}$.

4.3.3 Solution when $m = 2$ $(m \in M_3^{S_n^{\boldsymbol{k}}})$

When $m = 2$ the solution for the game $G(S_n^{\boldsymbol{k}}, T, 2)$ follows the same approach as the elongated star graph, that is using the covering patroller strategy $\boldsymbol{\pi}_{\mathrm{Cov}}$ and the independence attacker strategy $\boldsymbol{\phi}_{\mathrm{Ind}}$. These require the construction of a minimal covering set and a maximal independent set, in order to obtain the covering number, $\mathcal{C}_{S_n^{\boldsymbol{k}},T,2}$, and the independence number, $\mathcal{I}_{S_n^{\boldsymbol{k}},T,2}$.

For the game $G(S_n^{\boldsymbol{k}}, T, 2)$ an intercepting patroller strategy is equivalent to an edge, so we can form a covering set C in a similar fashion to that done for the game $G(S_n^k, T, 2)$ for some $k \geq 1$. That is, to construct C select any edge which has an incident leaf node (node of degree 1), add this edge to C and then delete the edge and the two incident nodes to form the graph Q_2. Repeat this process on the graph Q_2 until for some $l \in \mathbb{N}$ we have that $Q_l = (N_l, E_l)$ is a graph such that the $E_l = \emptyset$. Then for every node $j \in N_l$ add a connected edge from the original graph $Q_1 = S_n^{\boldsymbol{k}}$. Performing this process leads to the minimal covering set

$$C = \begin{cases} \{(c, *_{1,1})\} \cup \bigcup\limits_{x=1}^{n} N'(\boldsymbol{k}, x) & \text{if } k_i \text{ is odd for all } i \in \{1, ..., n\}, \\ \bigcup\limits_{x=1}^{n} (N'(\boldsymbol{k}, x) \cup N''(\boldsymbol{k}, x)) & \text{if } k_i \text{ is even for some } i \in \{1, ..., n\}, \end{cases}$$

where

$$N'(\boldsymbol{k}, x) = \begin{cases} \{(*_{x,k_x+1}, *_{i,k_x}), ..., (*_{x,2}, *_{x,1})\} & \text{if } k_x \text{ is odd}, \\ \emptyset & \text{otherwise}, \end{cases}$$

and

$$N''(\boldsymbol{k}, x) = \begin{cases} \{(*_{x,k_x+1}, *_{x,k_x}), ..., (*_{x,1}, c)\} & \text{if } k_x \text{ is even}, \\ \emptyset & \text{otherwise}, \end{cases}$$

The cardinality of C gives us the covering number for the game $G(S_n^{\boldsymbol{k}}, T, 2)$ and so

$$\mathcal{C}_{S_n^{\boldsymbol{k}},T,2} = \begin{cases} \frac{1}{1+\sum_{i=1}^{n} \frac{k_i+1}{2}} & \text{if } k_i \text{ is odd for all } i \in \{1, ..., n\}, \\ \frac{1}{\sum_{i=1}^{n} \lceil \frac{k_i+1}{2} \rceil} & \text{if } k_i \text{ is even for some } i \in \{1, ..., n\}, \end{cases} \tag{4.29}$$

Similarly, to construct a maximal independent set we start at leaf nodes and look at alternating nodes along branches at a distance of 2 apart. Hence we get a maximal independent set,

$$L = \begin{cases} \{c\} \cup \bigcup_{i=1}^{n} \{*_{i,k_i+1}, *_{i,k_i-1}, ..., *_{i,2}\} & \text{if } k_i \text{ is odd for all } i = 1, ..., n, \\ \bigcup_{i=1}^{n} \{*_{i,k_i+1}, *_{i,k_i-1}, ..., *_{i,2}\} & \text{if } k_i \text{ is even for some } i = 1, ..., n. \end{cases}$$

The cardinality of the set L gives us the independence number for the game $G(S_n^{\boldsymbol{k}}, T, 2)$ and so

$$\mathcal{I}_{S_n^{\boldsymbol{k}},T,2} = \begin{cases} \frac{1}{1+\sum_{i=1}^{n} \frac{k_i+1}{2}} & \text{if } k_i \text{ is odd for all } i \in \{1, ..., n\}, \\ \frac{1}{\sum_{i=1}^{n} \lceil \frac{k_i+1}{2} \rceil} & \text{if } k_i \text{ is even for some } i \in \{1, ..., n\}, \end{cases} \tag{4.30}$$

As the numbers in equations (4.29) and (4.30) are equal we have, by lemma 2.3.12 and 2.3.21, tight lower and upper bounds and the value of the game.

Lemma 4.3.7. *For a game $G(S_n^{\boldsymbol{k}}, T, m)$ for all $n \geq 1$, $\boldsymbol{k} \in \mathbb{N}^n$, for $T \geq 2$ and for all $m \in M_3^{S_n^{\boldsymbol{k}}}$ we have*

$$V(S_n^{\boldsymbol{k}}, T, 2) = \begin{cases} \frac{1}{1+\sum_{i=1}^{n} \frac{k_i+1}{2}} & \textit{if } k_i \textit{ is odd for all } i \in \{1, ..., n\}, \\ \frac{1}{\sum_{i=1}^{n} \lceil \frac{k_i+1}{2} \rceil} & \textit{if } k_i \textit{ is even for some } i \in \{1, ..., n\}, \end{cases}$$

achieved by the covering patroller strategy $\boldsymbol{\pi}_{Cov}$ and the independent attacker strategy $\boldsymbol{\phi}_{Ind}$.

4.3.4 Ideas for a solution when $m \in M_4^{S_n^{\boldsymbol{k}}}$

In this section we provide an idea, through example 4.3.8, on how to create bespoke strategies which improve the lower bound given by the random minimal full-node cycle (equation 4.25). In addition, we find a tight upper bound for the example and hence know that the strategies are optimal.

Example 4.3.8. For the game $G(S_3^{3,3}, 20, 3)$ we recall that a minimal full-node cycle $W_{\text{MFNC}}^{S_3^{3,3}}$ is

$$(*_{1,4}, *_{1,3}, *_{1,2}, *_{1,1}, c, *_{2,1}, *_{2,2}, *_{2,3}, *_{2,4}, *_{2,3}, *_{2,2}, *_{2,1}, c, *_{3,1}, c, *_{1,1}, *_{1,2}, *_{1,3}),$$

and that using it for a random minimal full-node cycle $\boldsymbol{\pi}_{\text{RMFNC}}^{S_3^{3,3}}$ gives a lower bound on the value of the game, by theorem 3.3.26, of

$$V(S_3^{3,3,}, 20, 3) \geq \frac{1}{6}.$$

We consider improving this baseline strategy $\boldsymbol{\pi}_0 = \boldsymbol{\pi}_{\text{RMFNC}}^{S_3^{3,3}}$ with three intercepting pure patrols $W_1 = (*_{1,4}, *_{1,3})$, $W_2 = (*_{2,4}, *_{2,3})$ and $W_3 = (*_{3,1}, c)$ (repeated for the time-horizon), which are played with probabilities p_1, p_2 and p_3 respectively to form $\boldsymbol{\pi}_1'$. We can then set up and solve the PIP (equation 3.19) in order to try to improve the current lower bound of $\frac{1}{6}$. As the performance of the hybrid strategy is minimal at the leaf nodes, $*_{1,4}$, $*_{2,4}$ and $*_{3,1}$, we know the PIP must have $p_1 = p_2 = p_3 \equiv p$. Thus solving the PIP is equivalent to solving

$$(1-3p)\frac{1}{6} + p = (1-3p)\frac{1}{3}.$$

Solving this gives $p = \frac{1}{9}$ and hence a lower bound of the game,

$$V(S_3^{3,3}, 20, 3) \geq \frac{2}{9} > \frac{1}{6},$$

which means the hybrid strategy is a strict improvement.

However it is possible to get an even better improvement by not using $\boldsymbol{\pi}_{\text{RMFNC}}^{S_3^{3,3}}$ as a baseline strategy. Consider using the following full-node cycle, which is not minimal and visits the leaf node $*_{3,1}$ twice, W_{FNC} which is

$$(*_{1,4}, *_{1,3}, *_{1,2}, *_{1,1}, c, *_{3,1}, c, *_{2,1}, *_{2,2}, *_{2,3}, *_{2,4}, *_{2,3}, *_{2,2}, *_{2,1}, c, *_{3,1}, c, *_{1,1}, *_{1,2}, *_{1,3}),$$

to form a random full-node cycle $\boldsymbol{\pi}_{\text{FNC}}$. Using $\boldsymbol{\pi}_{\text{FNC}}$ provides a better performance at the node $*_{3,1}$ but a worse performance at all other nodes. Using $\boldsymbol{\pi}_0 = \boldsymbol{\pi}_{\text{FNC}}$ with two intercepting pure patrols $W_1 = (*_{1,4}, *_{1,3})$ and $W_2 = (*_{2,4}, *_{2,3})$ played with probabilities p_1 and p_2 respectively to form $\boldsymbol{\pi}_2'$, we can again solve the PIP to get the best possible improvement. As before $p_1 = p_2 \equiv p$ and we are left to solve

$$(1-2p)\frac{3}{20} + p = (1-2p)\frac{6}{20}.$$

This gives $p = \frac{3}{26}$ and hence a lower bound provided of

$$V(S_3^{3,3}, 20, 3) \geq \frac{3}{13} > \frac{2}{9},$$

a strict improvement over the prior hybrid strategy.

To show that this second hybrid strategy $\boldsymbol{\pi}_2'$ is optimal, we develop a bespoke attacker strategy which provides a tight upper bound with the above lower bound. To do so we take the optimal attacker strategy for the game $G(S_3^3, 20, 3)$ and make use of node symmetry to create $\boldsymbol{\phi}_b$ which is such that the probability of choosing

$(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{26} & \text{for } j \in \{*_{1,4}, *_{2,4}\}, \tau \in \{0, 3\}, \\ \frac{2}{26} & \text{for } j \in \{*_{1,4}, *_{2,4}\}, \tau \in \{1, 2\}, \\ \frac{1}{26} & \text{for } j \in \{*_{1,2}, *_{2,2}\}, \tau = 1, \\ \frac{2}{26} & \text{for } j \in \{*_{1,1}, *_{2,1}\}, \tau \in \{1, 2\}, \\ \frac{2}{26} & \text{for } j = *_{3,1}, \tau = 1, 2. \end{cases} \tag{4.31}$$

Note that we leave the probabilities in equation (4.31) such that they have a common denominator, so we can compare the number of agents at each space-time point. Figure 4.3.2 shows the space-time agent matrix $\boldsymbol{S}_b^A$ for the attacker strategy $\boldsymbol{\phi}_b$, with 26 agents placed at space-time points. A simple computation shows that any pure patroller strategy can only catch a maximum of 6 potential agents out of the 26 played by the attacker strategy $\boldsymbol{\phi}_b$. Hence we have the upper bound

$$V(S_3^{3,3}, 20, 3) \leq \frac{6}{26}$$

which is tight with the lower bound given achieved by using $\boldsymbol{\pi}_2'$, so we know $\boldsymbol{\pi}_2'$ and $\boldsymbol{\phi}_b$ are optimal strategies and

$$V(S_3^{3,3}, 20, 3) = \frac{6}{26}.$$

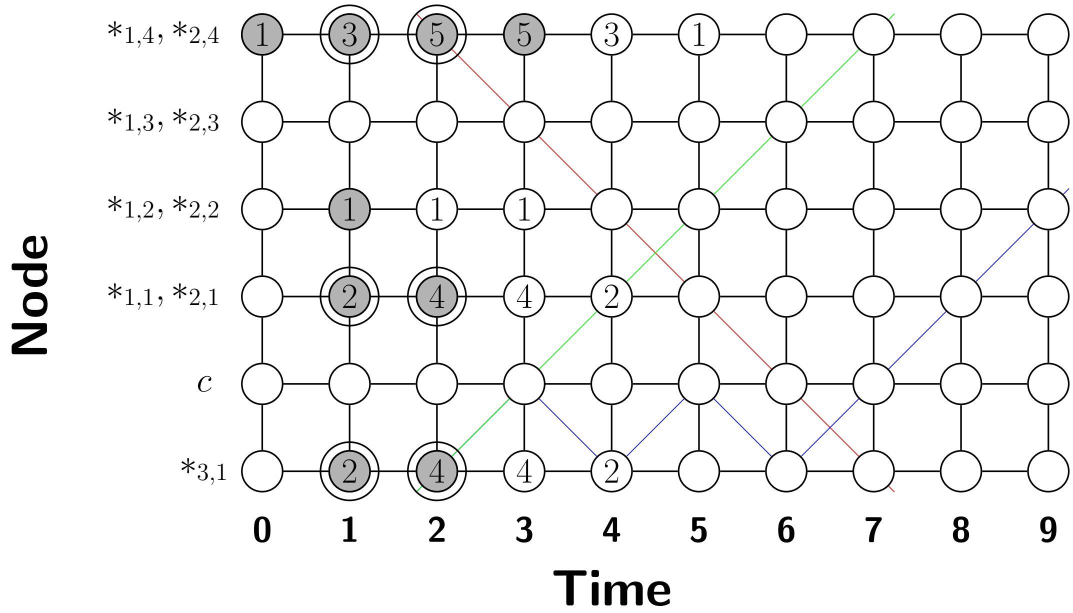


Figure 4.3.2: Space-time agent matrix $\boldsymbol{S}_b^A$ for the attacker strategy $\boldsymbol{\phi}_b$ as described in equation 4.31 for example 4.3.8 for the game $G(S_3^{3,3}, 20, 3)$. Three pure patroller strategies are shown in red, green and blue.

Example 4.3.8 shows us, that unlike for the elongated star graph, for the generalised star graph the baseline strategy may not be the random minimal full-node cycle when $m \in M_4^{S_n^k}$ (analogous to $M_4^{S_n^k} \cup M_5^{S_n^k}$). We provide no further working for the generalised star graph in this chapter and conclude our analysis here, noting that future work could explore how to utilise these ideas to create bespoke strategies. In chapter 6, section 6.1 we look at using distances on edges to allow us to model the scenario of a central hub location and locations at various distances from it. Such a model can be used for patrolling multiple cities which are at varying distances from a centralised hub. In the following section, we look at linking star graphs by there centres.

4.4 Linking generalised star graphs

In this section we build on the work done for a single star graph (be it the original, elongated or generalised) by linking them together to form more complex tree graphs. We start by solving for all attack lengths the *dual star* graph, which is formed by linking two star graphs, S_{n_1} and S_{n_2}, at their centres. We then follow this by looking at linking multiple generalised star graphs, finding a solution for a range of attack lengths in which the exact linking of the generalised star graphs' centres has no effect.

4.4.1 Dual star graph

In this section we look at the *dual star* graph S_{n_1,n_2}, which is formed by taking two star graphs S_{n_1} and S_{n_2} which are initially disconnected and adding in an edge between the two centres.

Definition 4.4.1. The *dual star* graph $S_{n_1,n_2} = (N, E)$, for $n_1, n_2 \in \mathbb{N}$, is such that the set of nodes is

$$N = \{c_1, *_{1,1}, ..., *_{n_1,1}\} \cup \{c_2, *_{1,2}, ..., *_{n_2,2}\}$$

and the set of edges is

$$E = \{(c_1, c_2)\} \cup \{(c_i, *_{r,i}) \mid i \in \{1, 2\} \text{ and } r \in \{1, ..., n_i\}\}.$$

We denote the set of all dual star graphs by $\mathcal{SD}$

Figure 4.4.1 shows an example of the dual star $S_{5,2}$. We note that when $n_1 = 1$ or $n_2 = 1$, then the dual star is an elongated star graph $S_{1,n} \equiv S_{n,1} \equiv S^1_{n+1}$. Therefore, we assume that $n_1, n_2 \geq 2$. In addition, without loss of generality, we assume $n_1 \geq n_2 \geq 2$. The nodes are labelled such that $*_{r,i}$ is the r^{th} leaf node adjacent to the i^{th} centre, c_i. The nodes $*_{r,i}$ for all $r \in \{1, ..., n_i\}$ for any $i \in \{1, 2\}$ are symmetric (under a relabelling), so an optimal attacker strategy must place pure attacks with the same probability at such nodes.

To begin our analysis of the game $G(S_{n_1,n_2}, T, m)$ we consider the decomposition of the game $G(S_{n_1,n_2}, T, m)$ into the subgraph games $G(S_{n_1}, T, m)$ and $G(S_{n_2}, T, m)$.

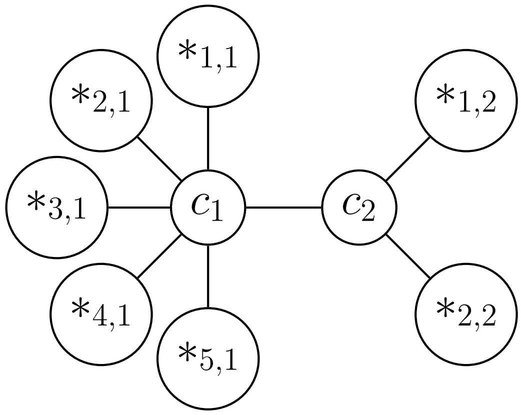


Figure 4.4.1: The dual star graph $S_{5,2} \in \mathcal{SD}$.

By lemma 2.3.14 we achieve the following lower bound, by the performance of the decomposed patroller strategy $\boldsymbol{\pi}_{\text{Dec}}^{S_{n_1,n_2}}$ of

$$\begin{aligned} V(S_{n_1,n_2},T,m) &\geq \frac{1}{\frac{1}{V(S_{n_1})}+\frac{1}{V(S_{n_2})}} \\ &= \frac{1}{\frac{1}{\min(1,\frac{m}{2n_1})}+\frac{1}{\min(1,\frac{m}{2n_2})}} \\ &= \begin{cases} \frac{m}{2(n_1+n_2)} & \text{for } m \leq 2n_2, \\ \frac{m}{2n_1+m} & \text{for } 2n_2 \leq m \leq 2n_1, \\ \frac{1}{2} & \text{for } m \geq 2n_1. \end{cases} \end{aligned} \tag{4.32}$$

When $m \leq 2n_2$ we can achieve a tight upper bound with the lower bound in equation (4.32) by considering a simplification, and thus an embedded attacker strategy. Consider the simplification of the dual star S_{n_1,n_2} into the star $S_{n_1+n_2}$ by node-identifying the centres c_1 and c_2. By corollary 3.3.8 this simplification gives the upper bound for any $n_1, n_2 \geq 2$, for all $m \geq 1$ and for all $T \geq m$ of

$$V(S_{n_1,n_2},T,m) \leq V(S_{n_1+n_2},T,m) = \frac{m}{2(n_1+n_2)}. \tag{4.33}$$

An optimal strategy for the game $G(S_{n_1+n_2},T,m)$ is the optimized 2-polygonal attacker strategy, which gives $V(S_{n_1+n_2},T,m)$. This strategy can be embedded into the game $G(S_{n_1,n_2},T,m)$ to form the attacker strategy $\boldsymbol{\phi}_s$ such that the probability of choosing $(j,\tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{n_1+n_2} & \text{for } j = *_{r,i} \text{ and } \tau = 0,1 \text{ for } i = 1,2, r = 1,...,n_i, \\ 0 & \text{otherwise,} \end{cases}$$

which achieves the bound in equation (4.33). Therefore, the attacker strategy $\boldsymbol{\phi}_s$ and the decomposition patroller $\boldsymbol{\pi}_{\text{Dec}}^{S_{n_1,n_2}}$ strategy are optimal and give the value of the game $G(S_{n_1,n_2},T,m)$ for $m \leq 2n_2$.

Next for $m > 2n_2$ we consider a minimal full-node cycle

$$W_{\text{MFNC}}^{S_{n_1,n_2}} = (c_1, *_{1,1}, c_1, *2, 1, ..., *_{n_1,1}, c_1, c_2, *_{1,2}, c_2, ..., *_{n_2,1}, c_2).$$

The minimal full-node cycle $W_{\text{MFNC}}^{S_{n_1,n_2}}$ is of length $2(n_1+n_2+1)$ and so, by theorem 3.3.26, the random full-node cycle patroller strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_{n_1,n_2}}$, gives us that for any $n_1, n_2 \geq 2$, for any $m \geq 1$ and for any $T \geq m$ a lower bound of

$$V(S_{n_1,n_2}, T, m) \geq \frac{m}{2(n_1+n_2+1)}. \tag{4.34}$$

For $m > 2n_2$, we can get a tight upper bound with the lower bound in equation (4.34) by using an attacker strategy with an asymmetric distribution in commencement time, attacking nodes on each star (nodes $*_{r,1}$ are nodes on star one and nodes $*_{r,2}$ are nodes on star two) with slightly different commencement time distributions.

Definition 4.4.2. The *time-spread* attacker strategy $\boldsymbol{\phi}_{\text{ts}}$ is such that the probability of choosing the pure attack $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2n_2(n_1+n_2+1)} & \text{for } j = *_{r,1},\ \tau \in \{1, ..., 2n_2\},\ \text{for some } r \in \{1, ..., n_1\}, \\ \frac{1}{2n_2(n_1+n_2+1)} & \text{for } j = *_{k,2},\ \tau \in \{0, ..., 2n_2+1\},\ \text{for some } k \in \{1, ..., n_2\}, \\ 0 & \text{otherwise.} \end{cases}$$

That is the time-spread attacker strategy chooses to attack node $*_{r,1}$ with probability

$$\frac{2n_2}{2n_2(n_1+n_2+1)}$$

for each $r \in \{1, ..., n_1\}$, then chooses a commencement time from $\{1, ..., 2n_2\}$ with equal probability; or chooses to attack node $*_{k,2}$ with probability

$$\frac{2(n_2+1)}{2n_2(n_1+n_2+1)}$$

for each $k \in \{1, ..., n_2\}$, then chooses a commencement time from $\{0, ..., 2n_2+1\}$ with equal probability. The time-spread attacker strategy is designed so that it compensates the pure patroller who chooses to start at $*_{r,1}$ for some $r = 1, ..., n_1$ and travels to $*_{k,2}$ for some $k = 1, ..., n_2$ by having an additional unit of commencement time for the additional unit of travel between nodes c_1 and c_2. By evaluating the performance of $\boldsymbol{\phi}_{\text{ts}}$ we are able to get a tight upper bound with the lower bound in equation (4.34).

Lemma 4.4.3. *For the game $G(S_{n_1,n_2}, T, m)$ for all $n_1 \geq n_2 \geq 2$, for all $m > 2n_2$ and for all $T \geq 2n_2 + m + 1$ we have*

$$V(S_{n_1,n_2}, T, m) \leq \frac{m}{2(n_1+n_2+1)},$$

achieved by the time-spread attacker strategy $\boldsymbol{\phi}_{ts}$.

The proof of lemma 4.4.3 follows by the design of the time-spread attacker strategy compensating any pure patroller who moves between stars with an extra potential agent for using one unit of time in moving between centres. We leave the proof of lemma 4.4.3 to appendix B.4, as it follows similar arguments to that of the proof of lemma 4.2.6. The results in this section yield in the following theorem.

Theorem 4.4.4. *For the game $G(S_{n_1,n_2}, T, m)$ for all $n_1 \geq n_2 \geq 2$, the value of the game is given by,*

- *for all $m \leq 2n_2$ and for all $T \geq 2$ we have*
$$V(S_{n_1,n_2}, T, m) = \frac{m}{2(n_1 + n_2)},$$
achieved by the decomposition patroller strategy $\boldsymbol{\pi}_{Dec}^{S_{n_1,n_2}}$ (into subgraph games $G(S_{n_1}, T, m)$ and $G(S_{n_2}, T, m)$) and the embedded attacker strategy $\boldsymbol{\phi}_s$ (from $G(S_{n_1+n_2}, T, m)$)

- *for all $m > 2n_2$ and for all $T \geq 2n_2 + m + 1$ we have*
$$V(S_{n_1,n_2}, T, m) = \frac{m}{2(n_1 + n_2 + 1)},$$
achieved by a random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{RMFNC}^{S_{n_1,n_2}}$ and the time-spread attacker strategy $\boldsymbol{\phi}_{ts}$.

4.4.2 Multi general star graphs

In this section we look at p-linked general star graphs, $(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)$, which are formed by taking generalised star graphs $S_{n_l}^{\boldsymbol{k}_l}$ for $l = 1, ..., p$, which are initially disconnected and have the graph $Q_c = (N_c, E_c)$, called the *centre link* graph, where N_c contain all the centres of $S_{n_l}^{\boldsymbol{k}_l}$ for $l = 1, ..., p$.

Definition 4.4.5. The *p-linked general star* graph, $(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)$ (where $Q_c = (N_c, E_c)$ is a connected graph with $N_c = \{c_1, ..., c_p\}$) is a graph (N, E) such that
$$N = \bigcup_{l=1}^{p} \left(\{c_l\} \cup \{*_{l,i,r} \mid i = 1, ..., n_l \text{ and } r = 1, ..., k_i + 1\}\right),$$
and
$$E = E_c \cup \left(\bigcup_{l=1}^{p} (E_c(l) \cup E_b(l))\right),$$
where
$$E_c(l) = \{(c_l, *_{l,i,r}) \mid r = 1, ..., n_l\}$$
and
$$E_b(l) = \{(*_{l,i,r}, *_{l,i,r+1}) \mid i = 1, ..., n_l \text{ and } r = 1, ..., k_i\}.$$
We denote the class of linked general star graphs by $\mathcal{SL}$.

The nodes are labelled such that $*_{l,i,r}$ is the r^{th} node along the i^{th} branch of the l^{th} generalised star graph $S_{n_l}^{\boldsymbol{k}_l}$. In order to ease the notation, we let $k_{l,i}$ be the i^{th} element of the vector $\boldsymbol{k}_l$ and as with generalised star graphs we define some summary information:

- $k_{l,\text{sum}} \equiv \sum_{i=1}^{n_l} k_{l,i}$,
- $k_{l,\max} \equiv \max_{i=1,\dots,n_l} k_{l,i}$.

To begin our analysis we will note it is possible to construct a minimal full-node cycle $W_{\text{MFNC}}^{(S_{n_1}^{\boldsymbol{k}_1},\dots,S_{n_p}^{\boldsymbol{k}_p} | Q_c)}$ via minimal full-node cycles $W_{\text{MFNC}}^{S_{n_l}^{\boldsymbol{k}_l}}$ for $l = 1, \dots, p$ and the minimal full node cycle $W_{\text{MFNC}}^{Q_c}$ by inserting $W_{\text{MFNC}}^{S_{n_l}^{\boldsymbol{k}_l}}$ into $W_{\text{MFNC}}^{Q_c}$ when the node c_l for the first time for each $l = 1, \dots, p$. Therefore we know the length of $W_{\text{MFNC}}^{(S_{n_1}^{\boldsymbol{k}_1},\dots,S_{n_p}^{\boldsymbol{k}_p} | Q_c)}$ is

$$2 \sum_{l=1}^{p} (n_l + k_{l,\text{sum}}) + F(Q_c),$$

where $F(Q_c)$ is the minimal full-node cycle length of the graph Q_c. Note that we know that as $p \leq F(Q_c) \leq 2(p-1)$, by considering the best case scenario of Q_c being Hamiltonian and the worst case scenario of it being a line graph. Thus we immediately know

$$V((S_{n_1}^{\boldsymbol{k}_1}, \dots, S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m) = 1$$

if

$$m \geq 2 \sum_{l=1}^{p} (n_l + k_{l,\text{sum}}) + F(Q_c)$$

as $W_{\text{MFNC}}^{(S_{n_1}^{\boldsymbol{k}_1},\dots,S_{n_p}^{\boldsymbol{k}_p} | Q_c)}$ is an intercepting patroller strategy covering all nodes. When

$$m < 2 \sum_{l=1}^{p} (n_l + k_{l,\text{sum}}) + F(Q_c)$$

the patroller can get the lower bound, by theorem 3.3.26, of

$$V((S_{n_1}^{\boldsymbol{k}_1}, \dots, S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m) \geq \frac{m}{2 \sum_{l=1}^{p} (n_l + k_{l,\text{sum}}) + F(Q_c)}, \tag{4.35}$$

achieved by the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{(S_{n_1}^{\boldsymbol{k}_1},\dots,S_{n_p}^{\boldsymbol{k}_p} | Q_c)}$.

Figure 4.4.2 shows the 4-linked general star graph $(S_4^{3,1,1}, S_3^2, S_2^{2,1}, S_2^{1,1} \mid Q_c)$, where the centre link graph Q_c has an edge set $E_c = \{(c_1, c_2), (c_2, c_3), (c_2, c_4), (c_3, c_4)\}$. We do not label all the nodes on the graph in order to make the graph easy to parse. In addition we provide a minimal full-node cycle on the graph Q_c allowing the construction of $W_{\text{MFNC}}^{(S_{n_1}^{\boldsymbol{k}_1},\dots,S_{n_p}^{\boldsymbol{k}_p} | Q_c)}$.

We now consider a patroller decomposition and an attacker simplification in order to find the value of the game, which is independent of the graph Q_c. Consider the decomposition of the p-linked general star, into the corresponding separate generalised star graphs, $S_{n_l}^{\boldsymbol{k}_l}$ for $l = 1, \dots, p$. By lemma 3.3.2, and the random minimal full-node cycle strategies $\boldsymbol{\pi}_{\text{RMFNC}}^{S_{n_l}^{\boldsymbol{k}_l}}$ for $l = 1, \dots, p$, (for which the lower

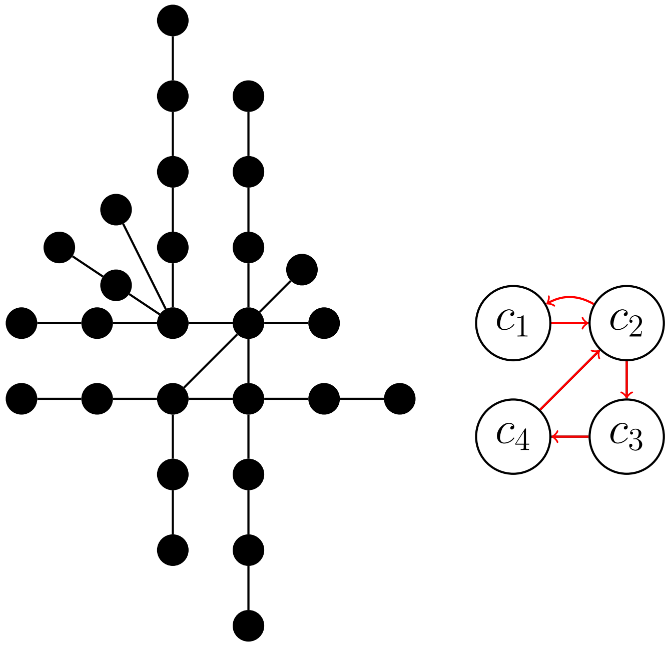


Figure 4.4.2: The 4-linked star graph $(S_4^{3,1,1}, S_3^2, S_2^{2,1}, S_2^{1,1}, Q_c)$, shown on the left with the centre linking subgraph, Q_c, as seen on the right. A minimal full-node cycle for Q_c is shown in red, repeating one node, c_2, for a length of 5.

bounds are given by equation (4.2), we achieve the following lower bound, by the performance of $\boldsymbol{\pi}_{\text{Dec}}^{(S_{n_1}^{\boldsymbol{k}_1},\ldots,S_{n_p}^{\boldsymbol{k}_p}|Q_c)}$, for

$$m \leq 2 \min_{l=1,\ldots,p} (n_l + k_{l,\text{sum}}),$$

of

$$V((S_{n_1}^{\boldsymbol{k}_1}, \ldots, S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m) \geq \frac{m}{2\sum_{l=1}^{p}(n_l + k_{l,\text{sum}})}, \tag{4.36}$$

We remark that we do not know the value of subgraph games $G(S_{n_l}^{\boldsymbol{k}_l}, T, m)$ for all $m \leq 2\min_{l=1,\ldots,p}(n_l + k_{l,\text{sum}})$ for all $l \in \{1, \ldots, p\}$, however we do know that the random minimal full-node cycle provides the lower bound given in equation (4.25) thus giving equation (4.36). The limit on the attack length of $m \leq 2\min_{l=1,\ldots,p}(n_l + k_{l,\text{sum}})$ is to ensure that the lower bound on the value of the subgraph games $(\frac{m}{2(n_l+k_{l,\text{sum}})})$ does not go above 1 (for which would require us to take the minimum with 1 of such a lower bound).

Consider the simplification of a p-linked star graph, by identifying nodes c_l for $l = 2, \ldots, p$ with c_1. Therefore simplifying the graph from $(S_{n_1}^{\boldsymbol{k}_1}, \ldots, S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)$ to, under a relabelling (isomorphism), $S_A^{\boldsymbol{K}}$, where $A = \sum_{i=1}^{p} n_i$ and $\boldsymbol{K}$ is the concatenation (and reordering) of the vectors $\boldsymbol{k}_l$ for $l = 1, \ldots, p$. Thus, by corollary 3.3.8, we have for all $p \geq 1$, for all $n_l \geq 1$, $\boldsymbol{k}_l \in \mathbb{N}_0^{n_l}$, for all Q_c, for all $m \geq 2(\max(\boldsymbol{K}) + 1)$

and for all $T \geq m$ that

$$V((S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m) \leq V(S_A^{\boldsymbol{K}}, T, m) \leq \frac{m}{2\sum\limits_{l=1}^{p}(n_l + k_{l,\text{sum}})}. \tag{4.37}$$

The upper bound in equation 4.37 is achieved by the embedding of the time-centred attacker strategy $\boldsymbol{\phi}_{\text{type}} \in \Pi(S_A^{\boldsymbol{K}}, T, m)$ into the game $G((S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m)$ resulting in $\boldsymbol{\phi}_{\text{ltype}} \in \Phi((S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m)$ such that the probability of choosing pure attack $(j, \tau) \in \mathcal{A}$ is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{2\sum\limits_{l=1}^{p}(n_l + k_{l,\text{sum}})} & \text{for } j = *_{l,i,k_i+1}, \tau \in \mathcal{T}_{\text{ltype}}(l, i) \text{ for some } l \in X, i \in Y(l), \\ 0 & \text{otherwise,} \end{cases}$$

where $X = \{1, ..., p\}$, $Y(l) = \{1, ..., n_l\}$, $K_{\max} = \max_{l \in \{1,...,p\}}(k_{l,\max})$ and

$$\mathcal{T}_{\text{ltype}}(l, i) = \{2K_{\max} + 1 - k_{l,i}, ..., 2K_{\max} + 1 + k_{l,i}\}.$$

As the upper bound in equation (4.37) is tight with the lower bound in equation (4.36) we arrive at the following lemma.

Lemma 4.4.6. *For the game $G((S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m)$, for all $p \geq 1$, for all $n_l \geq 1$, for all $\boldsymbol{k}_l \in \mathbb{N}_0^{n_l}$, for all $Q_c \subset K_p$, for all $2(\max_{l=1,...,p}(k_{l,max} + 1) \leq m \leq 2\min_{l=1,...,p}(n_l + k_{l,sum})$ and for all $T \geq m$ we have*

$$V((S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c), T, m) = \frac{m}{2\sum\limits_{l=1}^{p}(n_l + k_{l,sum})},$$

achieved by the decomposition patroller strategy $\boldsymbol{\pi}_{Dec}^{(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c)}$ (into subgraph games $G(S_{n_l}^{\boldsymbol{k}_l}, T, m)$ for which the random minimal full-node cycle is played for $l = 1, ..., p$) and embedded type time-centred attacker strategy $\boldsymbol{\phi}_{ltype}$ (from the game $G(S_A^{\boldsymbol{K}}, T, m)$).

We note that lemma 4.4.6 has no dependence on Q_c as the graphs are played on separately by both players. We can say that the edges in E_c in such a case are *superfluous*, as they are not needed by the patroller in order to achieve optimality. In chapter 6, section 6.1, that this idea can allow extensions of the patrolling game to utilise previous results.

4.5 Chapter conclusion

In this chapter we have defined multiple extensions to the star graph; the elongated star graph , the generalised star graph, the dual-star graph and the p-linked general star graph.

The majority of the work focused on the class of patrolling games when the graph is an elongated star graph $S_n^k \in \mathcal{SE}$, for which we decomposed the set of attack lengths into $M_i^{S_n^k}$ for $i = 0, 1, ..., 5$ and found optimal strategies for all regions aside from $M_4^{S_n^k}$, in which we were only able to provide a solution when $n = 3$ for $m \in M_{4,0}^{S_n^k}$.

We started by finding a minimal full-node cycle $W_{\text{MFNC}}^{S_n^k}$ and were able to use $W_{\text{MFNC}}^{S_n^k}$ to state the value of the game in the region $M_1^{S_n^k}$. This was followed by using $W_{\text{MFNC}}^{S_n^k}$ to form the random minimal full-node cycle strategy $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$ in order to find a lower bound in the region $M_2^{S_n^k}$, which we then developed the time-centred attacker strategy $\boldsymbol{\phi}_{\text{tc}}$ in order to achieve a tight upper bound and hence be able to state the value of the game. For the region $M_0^{S_n^k}$ the value was already known by lemma 2.3.26 and for the region $M_3^{S_n^k}$ we were able to show that the covering strategy $\boldsymbol{\pi}_{\text{Cov}}$ and independent strategy $\boldsymbol{\phi}_{\text{Ind}}$ were optimal and hence we were able to state the value of the game.

For the two remaining regions $M_4^{S_n^k}$ and $M_5^{S_n^k}$ we used the patrol improvement program(PIP) in order to improve the performance of $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^k}$, focusing on nodes for which it performed weakly at. In doing so we saw that PIP gave two different lower bounds in each region dependent on the attack length m and hence decomposed $M_4^{S_n^k}$ and $M_5^{S_n^k}$ further into $M_{4,0}^{S_n^k}$, $M_{4,1}^{S_n^k}$, $M_{5,0}^{S_n^k}$ and $M_{5,1}^{S_n^k}$. We then found a tight upper bound in the region $M_{4,0}^{S_n^k}$ when $n = 3$ by creating a bespoke attacker strategy. This was followed by the creation of four bespoke attacker strategies $\boldsymbol{\phi}_{\rho-\text{stc}}$ dependent on $\rho = m - 2k - 2 \mod 4$, which for $\rho \in \{0, 2\}$ provided tight upper bounds and for $\rho \in \{1, 3\}$ provided near tight upper bounds. We acknowledge, that similar to the patrollling games when $Q = L_n$ for some $n \in \mathbb{N}$, finding bespoke attacker solutions is time-consuming and requires multiple attacker strategies which are each used for different attack length values dependent on ρ. Our findings are collated in theorem 4.2.24.

We then extended the class of elongated star graphs further to look at generalised star graphs $S_n^{\boldsymbol{k}} \in \mathcal{SG}$ and found the value and optimal strategies when $m \in M_i^{S_n^{\boldsymbol{k}}}$ for $i = 0, 1, 2, 3$. However for the remaining attack lengths we halted our analysis, providing an idea of how we could find optimal strategies, as we suspect that for such attack lengths the game will require numerous bespoke attacker strategies. In particular we solved the game $G(S_n^{\boldsymbol{k}}, T, m)$ when the random minimal full-node cycle $\boldsymbol{\pi}_{\text{RMFNC}}^{S_n^{\boldsymbol{k}}}$ is optimal.

When we extended star graphs to dual stars $S_{n_1,n_2} \in \mathcal{SD}$, we were able to find the value of the game by adapting known strategies for both the attacker and patroller. We then further extended the idea to linking generalised star graphs to $(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} \mid Q_c) \in \mathcal{SL}$, solving the game in which the decomposition patroller is optimal $\boldsymbol{\pi}_{\text{Dec}}^{(S_{n_1}^{\boldsymbol{k}_1}, ..., S_{n_p}^{\boldsymbol{k}_p} | Q_c)}$, meaning that the graph Q_c connecting the centres is irrelevant and both the attacker and patroller play proportionally on each of the generalised stars $S_{n_l}^{\boldsymbol{k}_l}$ proportional to the value of the game $G(S_{n_l}^{\boldsymbol{k}_l}, T, m)$.

We conclude by remarking that for any game $G(Q, T, m)$ when Q is a tree, we have seen that the random minimal full-node cycle $\boldsymbol{\pi}^{Q}_{\mathrm{RMFNC}}$ is optimal for some range of attack lengths. This idea forms the foundation of the following chapter, in which we conjecture about the range of attack lengths for which $\boldsymbol{\pi}^{Q}_{\mathrm{RMFNC}}$ is optimal for games when Q is a tree (or forest). More precisely we conjecture that when the lower bound achieved by using $\boldsymbol{\pi}^{Q}_{\mathrm{RMFNC}}$ is at least a half, then $\boldsymbol{\pi}^{Q}_{\mathrm{RMFNC}}$ is optimal.

Chapter 5

Patrolling games on general trees

5.1 Chapter introduction

In this chapter, we provide optimal strategies to the game $G(Q, T, m)$ where Q is a tree with n nodes. We provide optimal strategies when $m = 2$ and in addition we conjecture about the optimality of the random minimal full-node cycle strategy $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ when $m > n - 1$ in conjecture 5.3.2. Further to this we discuss when these results might be applicable to non-tree graphs.

In section 5.2 we present a solution to the patrolling game $G(Q, T, 2)$, where Q is a tree. Optimality in this game is reached by the use of the covering and independence strategies. While explicit covering and independence numbers are known for previously studied graphs, for a general tree it is not possible to write explicit values. However, even without explicit values these strategies will be seen to be optimal as their corresponding numbers are equal. To this end, we provide an algorithm to calculate the covering and independence number for any tree.

In section 5.3 we will present conjecture 5.3.2 about the optimality of the random minimal full-node cycle, $\boldsymbol{\pi}^Q_{\text{RMFNC}}$, where Q is a tree. We conjecture about the range of attack lengths for which $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ is optimal. While we are not able to provide proof of our claim, we do provide an intuitive understanding as to why we believe that the conjecture holds. In addition we note that for all games on trees with known solutions that our about the optimality of $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ holds. Furthermore, we see that the conjecture holds by an exhaustive search for the solutions to the games $G(Q, T, m)$, for all trees with $n \leq 8$ nodes, which meet the conjecture's criterion.

5.2 Solution to $G(Q, T, 2)$ when Q is a tree

In this section we find the solution to the game $G(Q, T, 2)$, for any tree Q with n nodes. We do so by the use of non-explicit covering and independent strategies. Having already seen many examples of trees in chapter 4 for which these strategies are optimal, we extend this idea to all trees. However unlike when the graph strucutre is defined, we will not be able to give explicit covering and independence numbers. In this section of work, we make use of the fact that when the attack length $m = 2$, a covering set is an edge covering and an independent set is a vertex independent set. Known algorithms for trees allow us to construct such sets in polynomial time ([136]). We will adapt these algorithms to find the sets simultaneously, thus making them of equal size and hence providing optimality. Lastly, we remark when this approach works for non-tree graphs.

To start we formally define the graph theoretic ideas of an edge covering and a vertex independence set.

Definition 5.2.1. For a graph $Q = (N, E)$, a set $E^C \subset E$ is called an *edge covering* if for all $i \in N$, there exists a $j \in N$ such that $(j, i) \in E^C$. Such a set of minimal cardinality is called a *minimal edge covering* and this cardinality is called the *minimal edge covering number*, denoted $\mathcal{E}_Q^C \in \mathbb{N}$.

For a graph $Q = (N, E)$, a set $V^I \subset N$ is called a *vertex independent set* if for all $i, j \in V^I$ we have $(i, j) \notin E$. Such a set of maximal cardinality set is called a *maximal vertex independent set* and this cardinality is called the *maximal vertex independent number*, denoted $\mathcal{V}_Q^I$.

For a graph $Q = (N, E)$, a set $V^C \subset N$ is called a vertex covering if for all $(i, j) \in E$, there exists $i \in V^C$ (or $j \in V^C$). Such a set of minimal cardinality set is called a *minimal vertex covering* and this cardinality is called the *minimal vertex covering number*, denoted $\mathcal{V}_Q^C$.

We note that a vertex covering set is the complement of a vertex independent set and therefore algorithms which find a minimal vertex covering set can be used to find a maximal independent set. Therefore, any vertex covering algorithm can be made into a vertex independent algorithm. We will later use this to find $\mathcal{V}_Q^I$.

To utilise these graph theory concepts in our patrolling game $G(Q, T, 2)$, we know that the covering number for the game $G(Q, T, 2)$ is $\mathcal{C}_{Q,T,2} = \mathcal{E}_Q^C$ and similarly the independence number for the game $G(Q, T, 2)$ is $\mathcal{I}_{Q,T,2} = \mathcal{V}_Q^I$. Therefore, by lemmas 2.3.12 and 2.3.21 we have for the game $G(Q, T, 2)$, for any Q and any $T \geq 2$ that

$$\frac{1}{\mathcal{E}_Q^C} \leq V(Q, T, 2) \leq \frac{1}{\mathcal{V}_Q^I}. \tag{5.1}$$

Hence if $\mathcal{E}_Q^C = \mathcal{V}_Q^C$ for a graph Q, then we will have found optimal strategies to the game $G(Q, T, 2)$. Next, in example (5.2.2) we give examples of graphs, showing that both equality and strict inequality of equation 5.1 are possible. Note that in the example Q_1 and Q_2 are not trees, this is done in order to enforce the fact that being a tree is not necessary to get equality from equation (5.1). We will later see that being a tree is however a sufficient condition.

Example 5.2.2. For Q_1, as seen in figure 5.2.1, we have $E^C_{\min} = \{(1, 2), (3, 4), (4, 5)\}$ and $V^I_{\max} = \{1, 5\}$ (Note that other minimal and maximal sets are possible). So $\mathcal{E}^C_{Q_1} = 3$ and $\mathcal{V}^I_{Q_1} = 2$. Therefore we cannot use the covering strategy and independent strategy, equation (5.1) to immediately get the value of the game. At least one of the strategies is not optimal.

For Q_2, as seen in figure 5.2.1, we have $E^C_{\min} = \{(1,2),(3,5),(4,6)\}$ and $V^I_{\max} = \{1,5,6\}$ (Again, other minimal and maximal sets are possible). So $\mathcal{E}^C_{Q_2} = 3$ and $\mathcal{V}^I_{Q_2} = 3$. Therefore we immediately have, by equation (5.1), for any $T \geq 2$ that

$$V(Q_2, T, 2) = \frac{1}{3}.$$

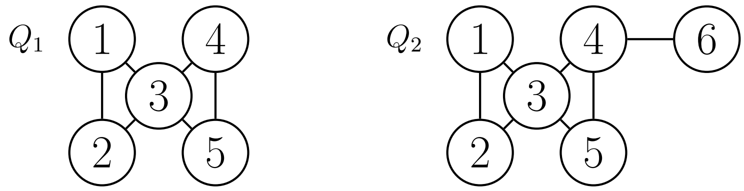


Figure 5.2.1: Graphs Q_1 and Q_2 as in example 5.2.2

We will now consider what graphical properties for Q are sufficient to get $\mathcal{E}^C_Q = \mathcal{V}^I_Q$. To do so we will adapt a known algorithm which attempts to finds a maximal independent set for any graph Q (algorithm 326 in [136]) into algorithm 2 which attempts to find a minimal edge covering and a maximal vertex independent set for the graph Q. While the algorithm is not guaranteed to produce a minimal edge covering and a maximal vertex independent set for any graph Q it is guaranteed to find them if Q is a tree. As such, we are able to say that Q being a tree is a sufficient condition to get two such sets, and by the algorithm which adds elements to both sets simultaneous these sets are of equal cardinality.

Algorithm 2 finds all leaf nodes (nodes with degree 1), adds them to the vertex independent set and adds the edge to their adjacent node to the edge covering. This is followed by a deletion of all leaf nodes and their adjacent nodes. This process is repeated until there are no leaf nodes left, at which point if there are no nodes left in the graph we have found a minimal edge covering and a maximal vertex independent set for the graph Q. Otherwise, no such sets are found and other similar algorithms (see [136]) would need to be attempted to find sets simultaneous.

For the application of algorithm 2 we note that we previously assumed Q has no loops. Therefore $M[i,i] = 0$ for all $i \in N$ and so the sum of a row is equivalent to the degree of the corresponding node. In addition we note that single node inclusions, as done when $j = NULL$, are included in the minimal edge covering in the form (i,k), where k is the lowest number node adjacent to i. However it is possible to replace this edge with (i,x) for any x adjacent to i. This decision of the adjacent node is irrelevant in the edge covering set which forms the covering strategy by having the patroller use walks which alternate between nodes using each edge with equal probability (which are intercepting walks), as the node x is already used in another edge in the covering set. Therefore, even though this choice may affect the performance at nodes, it does not affect the minimum of these and hence does not affect the performance of the covering strategy. Essentially an intercepting patrol can be chosen to wait at node i rather than repeatedly

Input: Graph Q
Result: Minimal covering set for Q, $E^C_{\min}$ and maximal vertex independent set for Q, $V^I_{\max}$ if final graph is empty. Otherwise no result.

Covering set $C = \emptyset$, Independence set $I = \emptyset$, Original node numbering $V = (1, ..., |N|)$, $M =$ Adjacency matrix of Q, $i = 1$;

while $i \leq$ *number of rows of* M **do**
  if *Sum of the* i^{th} *row of* $M == 0$ *or* 1 **then**
    Add node corresponding to i to independence set, $I = I \cup \{V[i]\}$;
    Set $j =$ to the column such that $M[i, j] == 1$;
    if $j \neq NULL$ **then**
      Add edge corresponding to (i, j) to the covering set, $C = C \cup \{(V[i], V[j])\}$;
      Remove nodes corresponding to i and j from the set of nodes, $V = V[(-i, -j)]$;
      Remove nodes corresponding to i and j from the adjacency matrix, $M = M[(-i, -j), (-i, -j)]$;
    else
      Set $k =$ first column such that $M[i, k] == 1$;
      Add edge corresponding to (i, k) to covering set, $C = C \cup \{(V[i], V[k])\}$;
      Remove node corresponding to i from the set of nodes, $V = V[-i]$;
      Remove node corresponding to i from the adjacency matrix, $M = M[-i, -i]$;
    end
    Graph Q' is formed with removed nodes and edges, repeat process from first row, $i = 0$;
  end
  No leaf node found, move to look at next row, $i = i + 1$
end
if $M == NULL$ **then**
  return $E^C_{min} = C$, $V^I_{max} = I$
end

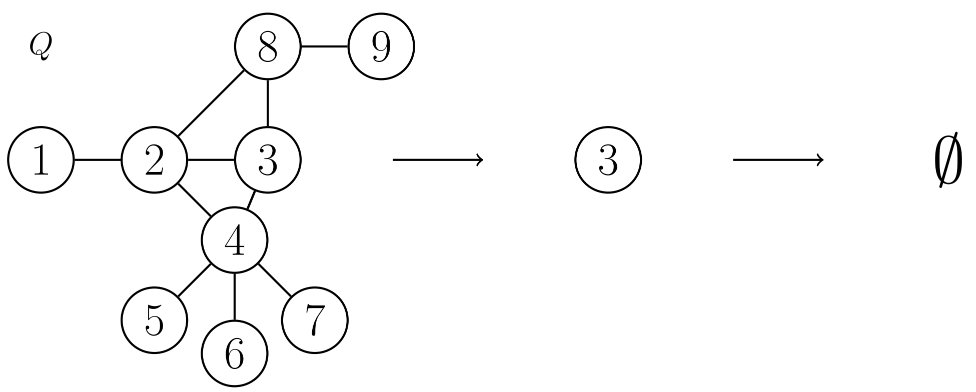
Algorithm 2: Algorithm to find a minimal edge covering set and maximal vertex independent set for graph Q.

move between i and x in the covering strategy as there is no loss at nodes with minimal performance.

Algorithm 2 takes polynomial time to terminate and in the case of the null graph at the end (M = NULL) returns a minimal edge covering and maximal vertex independent set, which by construction have equal cardinality. A sufficient condition for the algorithm to return such sets is that Q is a forest, as if Q is a forest (a collection of disconnected trees) then a deletion of a leaf node and its adjacent node from the graph results in a forest. Hence the algorithm will not stop adding to the sets (and removing leaves) until the graph is made empty (M = NULL). We note that we must consider forests as removing leaves and adjacencies from trees can disconnect the graph resulting in trees. However this condition is not necessary and algorithm 2 may terminate when Q is not a tree (see example 5.2.3).

While algorithm 2 is relatively fast, taking polynomial time to terminate, it is possible to speed up its implementation by using parallel processing of leaf nodes. However, care must be taken when two leaf nodes are adjacent/L_2 is isomorphic to a subgraph of the current graph. This approach is utilised in example 5.2.3 in order to illustrate the gain in speed by parallelization.

Example 5.2.3. For the graph Q, as seen below, we apply algorithm 2 in an attempt to find a minimal edge covering E^C_{min} and a maximal vertex independent set V^I_{max}. Note that we used a parallelized version of the algorithm in which all leaf nodes are considered at the same time.



Running algorithm 2 with Q as the input graph gives us,

- Initially: $I = \emptyset$, $C = \emptyset$.
- After 1 iteration: $I = \{1, 5, 6, 7, 9\}$, $C = \{(1, 2), (4, 5), (4, 6), (4, 7), (8, 9)\}$.
- Returned sets: $V^C_{\text{max}} = \{1, 3, 5, 6, 7, 9\}$,

$$E^C_{\text{min}} = \{(1, 2), (3, 2), (4, 5), (4, 6), (4, 7), (8, 9)\}$$

(Where $(3, 2)$ could be replaced by $(3, 4)$ or $(3, 8)$).

So for the graph Q the algorithm terminates as the final graph is empty so we have $\mathcal{C}_{Q,T,2} = \mathcal{I}_{Q,T,2} = 6$. Hence,

$$V(Q,T,2) = \frac{1}{6}.$$

Lemma 5.2.4. *For the game $G(Q,T,m)$ for any tree Q, for any $T \geq 2$, we have*

$$V(Q,T,2) = \frac{1}{\mathcal{C}_{Q,T,2}} = \frac{1}{\mathcal{I}_{Q,T,2}}.$$

This is achieved by the covering patroller strategy and independence attacker strategy. The exact value of $\mathcal{C}_{Q,T,2} = \mathcal{I}_{Q,T,2} \in \mathbb{N}$ is determined by algorithm 2 (with $\mathcal{C}_{Q,T,2} = \mathcal{I}_{Q,T,2} = |EC_{min}| = |VI_{max}|$).

Having seen the value and optimal strategies, albeit non-explicitly, for the game $G(Q,T,2)$ where Q is a tree (or for non-tree graphs when Algorithm 2 terminates and returns sets), we move onto looking for solutions for much higher attack lengths for trees (again with some remarks to non-tree graphs). More precisely we conjecture that the random minimal full-node cycle, $\boldsymbol{\pi}^Q_{\text{RMFNC}}$, is optimal for patrolling games on trees when $m \geq n-1$.

5.3 Conjecture on the optimality of $\boldsymbol{\pi}^Q_{\text{RMFNC}}$

In this section, we will discuss the optimality of the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}^Q_{\text{RMFNC}}$, for the game $G(Q,T,m)$. We will initially allow any graph Q, later restricting Q to trees and making conjecture 5.3.2. The idea of having a single intercepting patroller which contains all nodes is easy to see when considering a minimal full-node cycle under large attack lengths. That is we consider when $\mathcal{C}_{Q,T,m} = 1$ by the use of a minimal full-node cycle W^Q_{MFNC}. If $m \geq F(Q)$, where $F(Q)$ is the length of W^Q_{MFNC} then W^Q_{MFNC} is intercepting and as it contains all nodes we have $\mathcal{C}_{Q,T,m} = 1$. Hence, we obtain lemma 5.3.1.

Lemma 5.3.1. *For the game $G(Q,T,m)$ for any graph Q, for any $T \geq m$, for any $m \geq F(Q)$, where $F(Q)$ is the length of a minimal full-node cycle for Q, we have*

$$V(Q,T,m) = 1,$$

achieved by a minimal full-node cycle patroller strategy W^Q_{MFNC} and any attacker strategy.

This idea of the minimal full-node cycle has been used for a variety of graphs, including the line graphs L_n (see [107]) and various star graphs such as the generalised star graph $Q = S^{\boldsymbol{k}}_n$ in chapter 4. Further to this when the attack length m is lowered to be below the length of the minimal full-node cycle $F(Q)$, we have seen the solution in the next attack length region remains as the random minimal full-node cycle, $\boldsymbol{\pi}^Q_{\text{RMFNC}}$. The size of this attack length region depends upon at what m there is a strict improvement $\boldsymbol{\pi}^Q_{\text{RMFNC}}$. The performance of $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ at nodes is given by equation 3.17 and helps determine when such an improvement is possible. In order to decide if strict improvement is possible, we must consider how many and which nodes require improvement to improve the minimal performance among all nodes. That is all currently minimal nodes must be considered in improvement strategies. As we are using a minimal full-node cycle, leaf nodes are not repeated within W^Q_{MFNC} and so have the same performance as each other, which is the minimal among all nodes. Therefore in the case of Q being a tree, then all the leaf node are minimal nodes as they are the only nodes not repeated. Furthermore when Q is a tree, regardless of the edges of Q we know that $F(Q) = 2(n-1)$ (where $n = |N|$). For the rest of this section we focus on trees and thus assume Q is a tree, unless mentioned otherwise. The number of leaf nodes for a tree can vary depending on the graph structure from 2 to $n-1$ (corresponds with a line graph L_n and a star graph S_{n-1} respectively). As the number of leaf nodes increases the attack length at which strict improvement is possible has been seen to lower. This is intuitive as for more leaf nodes more intercepting patrols are needed and thus can only be effectively used for lower attack lengths. Thus we know that that the highest such attack length among all trees is given by that attack length for L_n i.e. $m \geq n-1$. We make the following conjecture to formalise this idea, noting that we have already seen empirical evidence that for all currently solved trees the conjecture holds.

Conjecture 5.3.2. *For the game $G(Q,T,m)$, where $Q = (N,E)$ ($|N| = n$) is a tree, if $\frac{1}{2} \leq V(Q,T,m) < 1$, then $\boldsymbol{\pi}^Q_{RMFNC}$ is an optimal strategy for the patroller. I.e, for any $m \in \mathbb{N}$ such that $n-1 \leq m < 2(n-1)$ and for any $T \geq 2m$ we have*

$$V(Q,T,m) = \frac{m}{2(n-1)}.$$

We may immediately note that we can limit ourselves to $n \geq 3$ (where $n = |N|$) as the only trees with $n = 1,2$ nodes are L_1 and L_2 on which all patrolling games are solved.That is any patrolling games with $Q = L_1$ has a value of 1 and the only patrolling game on $Q = L_2$ to not have a value of one is $G(L_2,T,2)$ for any $T \geq 2$ which has value $\frac{1}{2}$ by lemma 5.2.4 aligning with conjecture. Note that $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ is optimal for the game $G(L_2,T,2)$ as the lower bound it provides is equal to the lower bound from the covering strategy.

As previously mentioned all patrolling games solutions provided in this thesis alludes to the conjecture being true. That is it holds for S_n, L_n, $S^{\boldsymbol{k}}_n$ and S_{n_1,n_2}. Furthermore for those trees with more than 2 leaf nodes we have seen a much larger attack length region for which $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ is optimal. In order to make the case that the conjecture is likely to be true, we provide an intuitive reasoning when we assume that we are able to restrict the set of patroller strategies to only using closed walks. Afterwards, to further support our case about conjecture 5.3.2 we provide exhaustive results by testing every game on every tree with $n \leq 8$ nodes (which are not covered by lemma 5.3.1).

5.3.1 Intuitive reasoning for conjecture

In this section we provide the intuitive reasoning to why we make conjecture 5.3.2. Specifically, we show that $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ has no strict improvements when we consider *only using repeated minimal length closed walks between currently minimally performing nodes*. While such an assumption invalidates this reasoning as proof of the conjecture, it is intuitive to do so as only repeated closed walks provide unifromity in their performance at each minimal node seen during the closed walk for all time. Strategies which do not have this uniformity in time are only considered at their weakest time interval and so do not perform as well.

To see why $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ has no strict improvements, let us first recall its performance at the leaf nodes which is

$$\frac{m}{2(n-1)}.$$

Now in order to use the PIP (equation (3.19)) in an attempt to improve $\boldsymbol{\pi}^Q_{\text{RMFNC}}$, we must decide on other strategies which give a better performance at these leaf nodes. For the set of leaf nodes for the graph Q, $L \subset N$, we consider *using all minimal length closed walks* (a subset of patroller strategies) where each closed walk, W_i, contains a unique strict subset $L_i \subset L$, for $i = 1, ..., 2^{|L|} - 2$. Each of these closed walks W_i are repeated in the time-horizon to form the patroller strategies $\boldsymbol{\pi}_i$ for $i = 1, ..., 2^{|L|}-2$ which are considered for improvement of $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ in the PIP. Let $L(W_i)$ be the length of walk W_i, then the performance restricted to consideration of leaf nodes is

$$V_{j,\boldsymbol{\pi}_i,\bullet}(Q,T,m) = \begin{cases} \frac{m}{L(W_i)} & \text{if } j \in L_i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that we need not care about non-leaf nodes as the performance at such nodes is strictly better than any leaf node in each $\boldsymbol{\pi}_i$ for $i = 1, ..., 2^{|L|} - 2$. The PIP then becomes

$$\begin{aligned} &\text{maximize} \quad \min_{j \in N} \left(\frac{m}{2(n-1)} p_0 + \sum_{i=1}^{2^{|L|-2}} V_{j,\boldsymbol{\pi}_i,\bullet}(Q,T,m) p_i \right) \\ &\text{subject to} \quad \sum_{i=0}^{2^|L|-2} p_i = 1, \\ &\qquad\qquad\quad p_i \in [0,1], \ \text{for } i = 0, ..., 2^|L| - 2. \end{aligned} \tag{5.2}$$

Thus to solve the PIP (equation 5.2) we must have $\sum_{i=1}^{2^{|L|-2}} V_{j,\boldsymbol{\pi}_i,\bullet}(Q,T,m)p_i$ equal for all $j \in L$. Each $j \in L$ is contained in $2^{|L|-1} - 1$ of the $2^{|L|} - 2$ strategies, that is each node j is contained in exactly half of all strategies. Therefore, regardless of the performance $V_{j,\boldsymbol{\pi}_i,\bullet}(Q,T,m)$ and for any choice of p_i for $i = 1, ..., 2^{|L|} - 2$ we have $\sum_{i=1}^{2^{|L|-2}} V_{j,\boldsymbol{\pi}_i,\bullet}(Q,T,m)p_i \leq \frac{1}{2}$. Hence, to maximize the minimum among all nodes of this hybrid strategy the choice of $p_0 = 1$ should be made and subsequently we know the strategies $\boldsymbol{\pi}_i$ for $i = 1, ..., 2^{|L|} - 2$ provide no strict improvement for $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$. Therefore we conclude that if we only consider closed repeated walks as improvements then the best strategy is $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$ for the game $G(Q, T, m)$.

5.3.2 Computer testing

In this section we explain how we can test conjecture 5.3.2 using a computer program. We use linear program (2.10) to find the value of the game $G(Q, T, m)$ and hence if $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$ is optimal, doing so for each tree Q with a specific number of nodes n for $3 \leq m \leq 2(n-1)$ with $T = 2m$. By doing so we see that conjecture 5.3.2 holds for all trees with $n \leq 8$ and hence arrive at lemma 5.3.3.

To test the conjecture for a given number of nodes n we first generate all the trees with n nodes. In order to ensure we have generated all trees with n nodes, we take all trees with $n - 1$ nodes and consider adding a leaf node to each of the nodes forming separate graphs, after which all isomorphic trees are removed. Such tree generation is done by algorithm 3, to test for isomorphism in our tree generation we use algorithm 4.

When all the trees for a given number of nodes are calculated we can then compute the value of the game by using the linear program set up in section 2.2.3 (equation (2.10)). We do this for $n - 1 \leq m \leq 2(n - 1)$ to check the validity of conjecture 5.3.2 and for completeness we check $1 \leq m < n - 1$ decreasing down until a tree is found to not have the value predicted by the conjecture (that is $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$ is not optimal).

While the number of trees grows exponentially with the number of nodes, for our computational experiment we look at all trees with $n = 3, 4, 5, 6, 7, 8$ nodes sequentially and hence only have to check $1, 2, 3, 6, 11, 23$ trees respectively for each number of nodes (for OEIS (Online Encyclopaedia of Integer Sequences) A000055 is the sequence number for the number of trees with a given number of nodes, where labelling isn't relevant). While the number of trees we look at is not considerably large the computation time for the linear program on the patrolling game is considerable and this needs to be run for multiple attack lengths for each tree. In addition we assume $T = 2(n-1)$, so that the game length is large enough to allow for the attacker to distribute in the commencement time which further adds to the computation time due to the number of pure strategies exponentially increasing. We present our results in figure 5.3.1 in which we see that in the green region that the conjecture holds for all patrolling games on trees with up to 8 nodes. While it is possible to test out conjecture for all trees with $n \geq 9$ nodes, the computational time for one such game makes this infeasible. For example testing the all the patrolling games on the 28 trees with 8 nodes takes multiple days and in total our complete test took over a week with general computing power. Therefore for patrolling games on trees with $n \geq 9$ we only conjecture about the optimality of $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$. Figure 5.3.1 shows the region concerning the conjecture which has not been exhaustively tested in purple.

```
Input: Maximum number of nodes in trees n.
Result: Set of adjacency matrices for trees 𝒬 with n nodes or less nodes.
Tree of one node's adjacency matrix M = (1), Collection of trees
 𝒬 = {M} ;
𝒞𝒬 = 𝒬 ;
while i ≤ n do
  Empty the working set of trees, 𝒩𝒞𝒬 = NULL ;
  j = 1 ;
  while j ≤ |𝒞𝒬| do
    Set M to j^th adjacency matrix in the set CQ, Q = 𝒞𝒬[j] ;
    k = 1 ;
    while k ≤ number of rows of M do
      Add leaf node to node k to the adajanceny matrix M to form
       M' ;
      if M' is not isomorphic to any graph in 𝒩𝒞𝒬 then
        Add M' to 𝒩𝒞𝒬 ;
      end
      k = k + 1 ;
    end
    j = j + 1 ;
  end
  Set the set of trees with i + 1 nodes to 𝒞𝒬 = 𝒩𝒞𝒬 ;
  Add set of trees with i + 1 nodes to collection, 𝒬 = 𝒬 ∪ 𝒩𝒞𝒬 ;
  i = i + 1 ;
end
return 𝒬
```

Algorithm 3: An algorithm to generate all trees which have up to n nodes up to an isomorphic relabelling of nodes and edges.

Input: Adjacency matrix M and set of adjacency matrices $\mathcal{Q}$.
Result: TRUE if M is isomorphic to any matrix in $\mathcal{Q}$, FALSE otherwise.

```
i = 1 ;
while i ≤ |Q| do
    Set T equal to the i^th adjancey matrix in Q, T = Q[i] ;
    if number of rows of M == number of rows of T then
        Set Perms = set of all permutations of
          {1, ..., number of rows of T} excluding the permutation
          (1, ..., number of rows of T) ;
        j = 1 ;
        while j ≤ |Perms| do
            Permute the rows of matrix T using the j^th permutation from
              Perms to form matrix TR ;
            if TR == M then
                return TRUE
            end
        end
        j = j + 1 ;
    end
end
i = i + 1 ;
return FALSE
```

Algorithm 4: An algorithm to check if a graph is isomorphic to any graph in a set of graphs.

Lemma 5.3.3. *For the game $G(Q,T,m)$, where $Q=(N,E)$ is a tree and $|N|=n\leq 8$, if $V(Q,T,m)\geq\frac{1}{2}$, then $\boldsymbol{\pi}^Q_{RMFNC}$ is an optimal strategy for the patroller. I.e, for any $T\geq 2m$, and for any $m\geq n-1$, we have*

$$V(Q,T,m)=\frac{m}{2(n-1)}.$$

Further to this computer testing for the game $G(Q,T,m)$ where Q is a tree, we performed a similar test when Q is a non-tree graph in order to test the optimality of $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ for non-trees. Initial computer testing seems to suggest that a similar conjecture to conjecture 5.3.2 can be made where the boundary between regions where $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ is guaranteed to be optimal depends on the comparison of the attack length to the length of a minimal full-node cycle. For $m\geq\frac{1}{2}F(Q)$ (where $F(Q)$ is the length of the minimal full-node cycle for Q) we saw during computer tests that $\boldsymbol{\pi}^Q_{\text{RMFNC}}$ is optimal for non-trees as well as trees, thus we state conjecture 5.3.4.

Conjecture 5.3.4. *For the game $G(Q,T,m)$, if $V(Q,T,m)\geq\frac{1}{2}$, then $\boldsymbol{\pi}^Q_{RMFNC}$ is an optimal strategy for the patroller. I.e, for any $T\geq 2m$, and for any $m\geq\frac{1}{2}F(Q)$, we have*

$$V(Q,T,m)=\frac{m}{F(Q)},$$

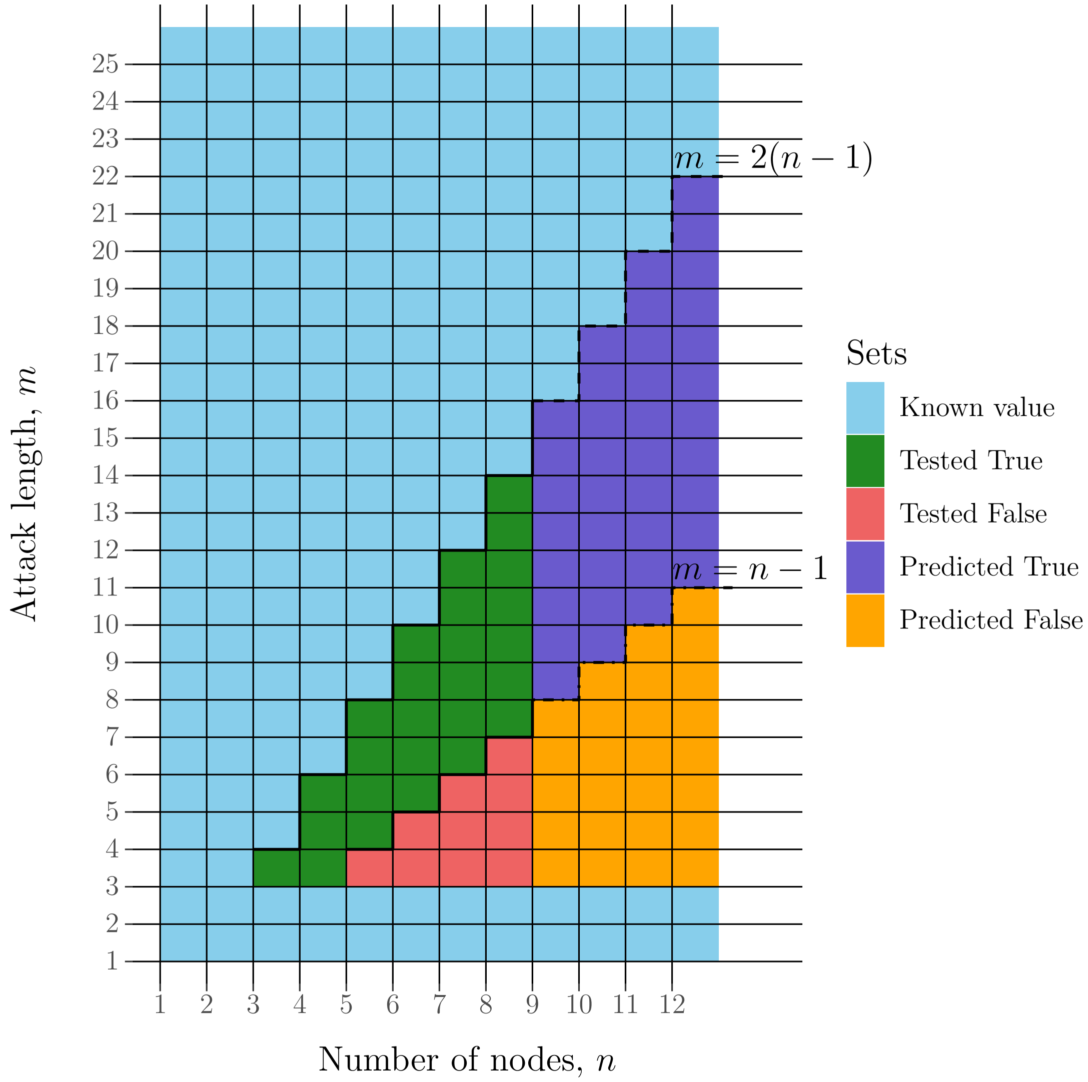


Figure 5.3.1: The figure shows different regions of (n, m) for which the blue region has a known value by either lemma 2.3.26, 5.2.4 or 5.3.1. By computer testing the green region has the conjecture hold true for all trees. The region below, in red, has $m < n - 1$ and is not covered by the conjecture. In this region some games have $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ as an optimal strategy, but not all. To the right of the green region we have the purple region in which we predict the conjecture will be true. To the right of the red region we have the orange region in which we know that $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ is only optimal for some patrolling games on trees. We know this by looking at the solution to the patrolling game on a star and tree. The regions are divided by the boundary lines $m = n - 1$ and $m = 2(n - 1)$ which are shown in the figure. All results assume $T \geq 2(n - 1)$.

where $F(Q)$ is the length of a minimal full-node cycle for Q. Moreover if Q is a tree then for any $T \geq 2(n-1)$, and for any $m \geq n-1$

$$V(Q,T,m) = \frac{m}{2(n-1)}.$$

5.4 Chapter conclusion

In this chapter we have found solutions to the patrolling games $G(Q,T,m)$ where Q is a tree. When $m = 2$ we did so by using the covering and independent strategies. However as $\mathcal{C}_{Q,T,2} = \mathcal{I}_{Q,T,2}$ are not explicit for trees we are not able to explicitly state $V(Q,T,m)$ and instead it must be calculated by algorithm 2. In addition we found the solution for any tree when $m \geq 2(n-1)$ (where $|N| = n$), by the use of the (random) minimal full-node cycle which is intercepting in such patrolling games.

Lastly we conjectured on the value of the patrolling game when $n-1 \leq m < 2(n-1)$, for $T \geq 2(n-1)$, in conjecture 5.3.2. This was done by looking when $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ is optimal, which we believe it is for this region of attack lengths. We intuitively argued why $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ is optimal in this attack length region, by looking at whether any improvement is possible through PIP. In doing so we restricted the improvements to only use repeated closed walk, showing that the entire collection of repeated closed walks provide no strict improvement over $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$. To construct a formal proof one would have to show no improvement is possible for all such strategies. While intuitively repeated closed walks seem the best considerations for improvement we are unable to prove this claim. For further evidence that the conjecture is true, by exhaustive computation, we showed that it holds for any tree with $n \leq 8$ nodes. As we are only able to conjecture such a result, future work needs to be undertaken in order to prove our conjecture. Essentially we must understand why it is possible to eliminate other strategies which are not repeated closed walks. In addition through computational experiments we are lead to believe that such optimality of $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ may be guaranteed for all patrolling games on non-tree graphs when $m \geq F(Q)$ and $T \geq 2m$. While this was not the focus of our work, it provides an interesting direction of study for future work.

This concludes our work on the classic patrolling game $G(Q,T,m)$, in the next chapter we will look at extensions to the patrolling game, which allow for more varied patrolling scenarios to be modelled.

Chapter 6

Extensions to patrolling games

In this chapter, we extend the notion of the patrolling game $G(Q, T, m)$ in several directions, allowing for a larger class of patrolling scenarios to be modelled using the patrolling game framework. We will describe the set-up for these extended patrolling games before providing solutions to certain classes of these games. This chapter will look at the following extensions to the patrolling game:

- Having an arbitrary distance on each edge rather than a constant unitary distance, allowing us to model a situation where connections between locations take longer to use.
- Having the attack length be node dependent rather than constant, allowing us to model situations where certain locations are more vulnerable than others.
- Making the game multi-player, allowing for a wide variety of scenarios depending on the coordination and symmetry of players and their collectives.

For each extension we will first describe the extension and formally state the alterations to the patrolling game made. We then obtain some results by relying on solutions to the patrolling game as studied earlier in chapters 2, 4 and 5. In order to avoid confusion between the game defined in chapter 2 and the following patrolling games with extensions, we refer to the previously defined patrolling game $G(Q, T, m)$ (as in chapter 2) as the *classic* patrolling game.

We start this chapter with the inclusion of edge distances in section 6.1, studying the game $G(Q, D, T, m)$ with D representing a positive real 'weight' for each edge of the graph Q modelling the distance between nodes. Of particular note in this section is the solution to a model of multiple cities at various distances from a centralised hub city, as discussed in chapter 4, 4.3. By using the game $G(Q, D, T, m)$ rather than the classic $G(Q, T, m)$ to model the scenario we are able to provide the optimal solution because we can ignore internal nodes along the branches of a generalised star graph $S_n^{\boldsymbol{k}}$ and instead use a distance between the centre and branch end which represent the hub and a city respectively.

In section 6.2 we look at extending the attack length m to be node dependent instead of constant, studying the game $G(Q, T, \boldsymbol{m})$, where $\boldsymbol{m} \in \mathbb{N}^n$. The game $G(Q, T, \boldsymbol{m})$ allows the modelling of a patrolling scenario where there are fortified or vulnerable locations, which respectively take the attacker longer or shorter amounts of time to complete a successful attack at.

Lastly, in section 6.3, we consider four different ways to introduce additional entities to either the patrolling or attacking side of the scenario, turning the game from a two player patrolling game into a multi-player patrolling game $G_i(k, l, Q, T, m)$ for some $i \in \{1, 2, 3, 4\}$. In doing so we will consider four models including selfish attackers and collaborative attackers whose objectives differ in order to see how altering the objective of the attackers alters optimal strategies between these games.

6.1 A patrolling game with edge distances

6.1.1 Introduction to patrolling games with edge distances

In this section we extend the classic patrolling game $G(Q, T, m)$ to the *patrolling game with edge distances* $G(Q, D, T, m)$ in which the additional parameter D assigns a positive real length ('weight') to each edge of the graph Q. By allowing D to be non-constant and non-unitary amongst edges we extend the classic patrolling game as D can be chosen to allow the game to model the time taken in travelling between locations. As in the classic game, we will still be under the assumption that the patroller moves with unitary speed as otherwise a simple scaling of the lengths will allow for a constant but non-unitary speed. As we do not restrict the lengths of the edges to be natural numbers but instead allow real values the game $G(Q, D, T, m)$ is played in real time instead of discrete time with the position of the patroller being along an edge or at a node.

The *patrolling game with edge distances* $G(Q, D, T, m)$ is parametrized by a 4-tuple (Q, D, T, m) where $Q = (N, E)$ is a simple undirected graph, $D : E \to \mathbb{R}$ is the *edge distance* function, $T \in \mathbb{R}$ is the game length (with the time-horizon $\mathcal{J} = [0, T]$) and $m \in \mathbb{R}$ is the attack length. While the classic patrolling game $G(Q, T, m)$ is a discrete game time game, with each move for the patroller taking one time epoch and each time epoch the attacker is at a node counting towards the required attack length m, the patrolling game with edge distances $G(Q, D, T, m)$ is a continuous time game. A continuous walk W of length $l \in \mathbb{R}$ is a function $W : [0, l] \to N \cup E$ such that for all $t_1 \in [0, l]$ if $W(t_1) \in N$ then $t_2 \geq t_1 + D(W(t_1, W(t_2)))$, where $t_2 = \min\{t \in [t_1, l] \mid W(t) \in N\}$. For a continuous walk W, $W(t)$ represents the position of the walk at time $t \in [0, l]$ which is either a node or edge. We note that the condition enforced on continuous walks is there to ensure that time between subsequent nodes is at least the distance between those respective nodes. While in the classic game there is no confusion on distance between nodes in the patrolling game with edge distance we need to distinguish between edge distances and shortest distances. We define the shortest distance between two nodes $j, j \in N$ as $d(j, j')$, the minimum length walk starting at node j and ending at node j'. We note that $d(j, j')$ is not necessarily equal to $D((j, j'))$ as the minimal length walk need not use the edge (j, j').

As with the classic patrolling game a pure patroller strategy is a walk around the graph until time T at which point the game ends. Therefore in the patrolling game with edge distances a pure patroller strategy is a continuous walk W of length T. In the patrolling game with edge distances a pure attacker strategy still chooses a node and commencement time, however this commencement time is now continuous, that is $(j, \tau) \in N \times [0, T]$, representing an attack at node j commencing at time τ (which ends at time $\tau + m$). By the same idea as in the classic patrolling game, any attack commencing after $T - m$ will fail to complete and so we can limit the attacker to choices of $\tau \in [0, T - m]$. The pure patroller strategies are collected in the set $\mathcal{W}(Q, D, T, m)$ and the pure attacker strategies are collected in the set $\mathcal{A}(Q, D, T, m) = N \times [0, T - m]$ (noting that we may omit the parameter space (Q, D, T, m)).

As in the classic patrolling game the patroller wins (and attacker loses) if the patroller is at the chosen node during the attack interval, otherwise the patroller loses (and attacker wins). Therefore the game is zero-sum and hence we only need to define the payoff from the patrollers perspective, thus the payoff for the patroller strategy $W \in \mathcal{W}$ against $(j, \tau) \in \mathcal{A}$ is

$$P(W, (j, \tau)) = \mathbb{I}_{\{j \in W([\tau, \tau+m])\}}.$$

We can assume some ordering of the sets $\mathcal{W}$ and $\mathcal{A}$ by two arbitrary bijections $\beta_1 : \mathcal{W} \to \mathbb{N}$ and $\beta_2 : \mathcal{A} \to \mathbb{N}$ respectively so $W_{(x)} = \beta_1^{-1}(x)$ and $a_{(y)} = \beta_2^{-1}(y)$. Then we can form a pure payoff matrix

$$\mathcal{P} = (P(W_{(x)}, a_{(y)}))_{x \in \{1, \dots, |\mathcal{W}|\}, y \in \{1, \dots, |\mathcal{A}|\}}, \tag{6.1}$$

with a maximizing patroller and minimizing attacker. Like the classic pure patrolling game, the pure patrolling game with edge distances has a Nash equilibrium if and only if there is a pure patrolling strategy which guarantees catching all pure attackers (i.e. there was a row of ones in the payoff matrix). Therefore, as in the classic patrolling game, this prompts us to allow for the mixing of strategies forming the mixed patrolling game with edges distances henceforth called the patrolling game with edge distances.

In the mixed patrolling game with edge distances each player chooses a probability distribution amongst all pure strategies. That is a mixed patroller strategy is $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{|\mathcal{W}|})$ where π_i is the probability of playing $W_{(i)}$ and a mixed attacker strategy is $\boldsymbol{\phi} = (\phi_1, \dots, \phi_{|\mathcal{A}|})$ where ϕ_i is the probability of playing $a_{(i)}$. As $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ are probability distributions we have

$$\boldsymbol{\pi} \in \Pi(Q, D, T, m) = \left\{ \boldsymbol{x} \in [0, 1]^{|\mathcal{W}(Q,D,T,m)|} \;\middle|\; \sum_{i=1}^{|\mathcal{W}(Q,D,T,m)|} x_i = 1 \right\},$$

$$\boldsymbol{\phi} \in \Phi(Q, D, T, m) = \left\{ \boldsymbol{y} \in [0, 1]^{|\mathcal{A}(Q,D,T,m)|} \;\middle|\; \sum_{i=1}^{|\mathcal{A}(Q,D,T,m)|} y_i = 1 \right\},$$

where $\Pi(Q, D, T, m)$ is the set of all mixed patroller strategies and $\Phi(Q, D, T, m)$ is the set of all mixed attacker strategies (again often omitting (Q, D, T, m) when it is clear). For the game $G(Q, D, T, m)$, the (patrollers) payoff for the patroller choosing $\boldsymbol{\pi} \in \Pi$ and the attacker choosing $\boldsymbol{\phi} \in \Phi$ is

$$P(\boldsymbol{\pi}, \boldsymbol{\phi}) = \sum_{i=1}^{|\mathcal{W}|} \sum_{j=1}^{|\mathcal{A}|} \mathcal{P}_{i,j} \pi_i \phi_j = \boldsymbol{\pi} \mathcal{P} \boldsymbol{\phi}^T, \tag{6.2}$$

with the objective of a maximizing patroller and minimizing attacker. In the same fashion as the classic game and it's MiniMax and MaxiMin play variants, a player choosing a strategy will determine a performance according to their strategy leading and the optimal strategy chose by the following player. Thus the performance of a patroller choosing the mixed patroller strategy $\boldsymbol{\pi} \in \Pi$ is given by

$$V_{\boldsymbol{\pi},\bullet}(Q, D, T, m) = \min_{\boldsymbol{\phi} \in \Pi} \boldsymbol{\pi} \mathcal{P} \boldsymbol{\phi}^T = \min_{a \in \mathcal{A}} P(\boldsymbol{\pi}, a),$$

and the performance of an attacker choosing the mixed attacker strategy $\boldsymbol{\phi} \in \Phi$ is given by

$$V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m) = \max_{\boldsymbol{\pi} \in \Pi} \boldsymbol{\pi} \mathcal{P} \boldsymbol{\phi}^T = \max_{W \in \mathcal{W}} P(W, \boldsymbol{\phi}).$$

By theorem 2.2.2 we have the value of the game $G(Q, D, T, m)$ which is played simultaneously is given by

$$V(Q, D, T, m) = \max_{\boldsymbol{\pi} \in \Pi} \min_{\boldsymbol{\phi} \in \Phi} P(\boldsymbol{\pi}, \boldsymbol{\phi}) = \min_{\boldsymbol{\phi} \in \Phi} \max_{\boldsymbol{\pi} \in \Pi} P(\boldsymbol{\pi}, \boldsymbol{\phi}).$$

Hence the value is bounded below by the performance of any mixed patroller strategy and above by any attacker strategy and therefore for all $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\phi} \in \Phi$ we have

$$0 \leq V_{\boldsymbol{\pi},\bullet}(Q, D, T, m) \leq V(Q, D, T, m) \leq V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m) \leq 1.$$

Before presenting results on the game $G(Q, D, T, m)$ we note that as mixed strategies include pure strategies we will drop the term mixed.

6.1.2 Results for patrolling games with edge distances

In this section we look at comparing various patrolling games with edge distances when the edge distance is increased and show that this can only result in a decrease of the games value. We follow this by showing that we can use our solutions to the classic patrolling game (Q, T, m) to solve patrolling games with edge distances (Q, D, T, m) as long as the edge distance for edges used with a non-zero probability in an optimal solution to the classic game are the only edges with a non-unitary distance.

Before presenting any results, we can see from the payoff for any game $G(Q, D, T, m)$ (in equation (6.1) and (6.2)) it is clear that pure patroller strategies/walks should not wait along edges (and may as well wait at nodes) and therefore we can restrict the pure patroller strategy set to

$$\mathcal{W}' = \{W \in \mathcal{W} \mid \forall t \in [0,T] \text{ if } W(t) \in N, W(t+s) \in E \text{ for all } 0 < s < \min_{e \in E} D(e) \\ \text{then } t' = t + D(W(t), W(t')) \text{ where } t' = \min\{s \in (t,T] \mid W(s) \in N\}\}.$$

As any walk $W \in \mathcal{W}'$ moves between adjacent nodes while possibly waiting at nodes we can write W as

$$W(t) = \begin{cases} j_1 & \text{for } t_1 \leq t \leq \nu_1, \\ (j_1, j_2) & \text{for } \nu_1 < t < t_2, \\ j_2 & \text{for } t_2 \leq t \leq t_1 + \nu_2, \\ \vdots & \vdots \\ (j_{k-1}, j_k) & \text{for } t_{k-1} + \nu_{k-1} < t < t_k, \\ j_k & \text{for } t_k \leq t \leq T, \end{cases}$$

for some $k \in \mathbb{N}$, for some $j_i \in N$ for all $i \in \{1, ..., k\}$ where

$$t_i = \begin{cases} 0 & \text{if } i = 1, \\ t_{i-1} + \nu_{i-1} + D(j_{i-1}, j_i) & \text{if } i = 2, 3, ..., k, \end{cases}$$

is the time of arrival at the i^{th} node and $t_k + \nu_k = T$. It is therefore easier to represent any walk $W \in \mathcal{W}'$ in *general move-wait form*

$$\omega = ((j_1, \nu_1), ..., (j_k, \nu_k)) \tag{6.3}$$

where the pair (j_i, ν_i) represents the patroller moving to node $j_i \in N$, such that $(j_{i-1}, j_i) \in E$, arriving at time t_i. We let Ω be the set of all general move-wait form walks equivalent to the set of non-edge waiting pure patroller strategies/walks $\mathcal{W}'$. Then

$$V(Q, D, T, m) = \min_{\boldsymbol{\phi} \in \Phi} \max_{\omega \in \Omega} P(\omega, \boldsymbol{\phi}) = \max_{\omega \in \Omega} \min_{\boldsymbol{\phi} \in \Phi} P(\omega, \boldsymbol{\phi}).$$

We note that this general move,wait form considered for the game $G(Q, D, T, m)$ is similar but not analogous to move,wait form used for evaluating the performance of attacker strategies as done in section 3.2.2.

An intuitive result is that increasing the distance on an edge can only decrease the value of the game as the patroller must spend more time to traverse it if its required in the optimal strategy.

Lemma 6.1.1. *For the patrolling games with edge distance $G(Q, D, T, m)$ and $G(Q, D', T, m)$ if $D'(e) \geq D(e)$ for all $e \in E$, then for all graphs Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q, D, T, m) \geq V(Q, D', T, m).$$

The proof of lemma 6.1.1 follows by using an augmented version of the patrollers strategy from the game $G(Q, D', T, m)$ in the game $G(Q, D, T, m)$ to get a lower bound and by using the exact same attacker strategy from the game $G(Q, D', T, m)$ in the game $G(Q, D, T, m)$.

Proof. First let us show that $V(Q, D, T, m) \geq V(Q, D', T, m)$. Let $\boldsymbol{\pi}^* \in \Pi(Q, D', T, m)$ be an optimal strategy to the game $G(Q, D', T, m)$ playing pure patrolling strategies $W_1, ..., W_p$ with non-zero probability i.e. $\pi^*_{\beta_1(W_x)} > 0$ for all $x \in \{1, ..., p\}$ such that

$$\sum_{x=1}^{p} \pi^*_{\beta_1(W_x)} = 1.$$

Let $\omega_1, ..., \omega_p$ be the general move,wait form of the pure patrolling strategies $W_1, ..., W_p$ and then augment these forming $\widetilde{\omega}_1, ..., \widetilde{\omega}_p$ such that

$$j_i(\widetilde{\omega}_x) = j_i(\omega_x),$$

and

$$\nu_i(\widetilde{\omega}_x) = \begin{cases} \nu_i(\omega_x) & \text{if } i = 1, \\ \nu_i(\omega_x) + D'((j_{i-1}, j_i)) - D((j_{i-1}, j_i)) & \text{if } 2 \leq i \leq k, \end{cases}$$

for all $i \in \{1, ..., k(\omega_x)\}$ for all $x \in \{1, ..., p\}$. That is $\widetilde{\omega}_x$ waits for some excess time, equal to the difference between the edge distances, after using each edge and follows the same nodes and edges as ω_x.

These augmented pure patroller strategies in general move,wait form $\widetilde{\omega}_1, ..., \widetilde{\omega}_1$ have normal form equivalents $\widetilde{W}_1, ..., \widetilde{W}_p \in \mathcal{W}(Q, D, T, m)$ and hence playing $\widetilde{W}_x$ with probability $\pi^*_{\beta_1(W_x)}$ for $x = 1, ..., p$ creates a mixed patroller strategy $\widetilde{\boldsymbol{\pi}} \in \Pi(Q, D, T, m)$. For the created strategy $\widetilde{\boldsymbol{\pi}}$ we have that $P(\widetilde{\boldsymbol{\pi}}, a) \geq P(\boldsymbol{\pi}^*, a)$ for any pure attack $a \in \mathcal{A}(Q, D, T, m)$, as any pure attack caught by ω_x is caught by $\widetilde{\omega}_x$ for all $x \in \{1, .., p\}$. In addition note that $\mathcal{A}(Q, D, T, m) = \mathcal{Q}, \mathcal{D}', \mathcal{T}, \Updownarrow$ and so we have that

$$V(Q, D, T, m) \geq \min_{a \in \mathcal{A}(Q,D,T,m)} P(\widetilde{\boldsymbol{\pi}}, a) \geq \min_{a \in \mathcal{A}(Q,D,T,m)} P(\boldsymbol{\pi}^*, a) = V(Q, D', T, m).$$

Hence we have shown $V(Q, D, T, m) \geq V(Q, D', T, m)$.

□

While we initially assumed the game was of continuous time as we allowed $D : E \to \mathbb{R}$ we will now limit ourselves to using only rational distances on edges, that is have a distance function such that $D : E \to \mathbb{Q}$, a rational game length $T \in \mathbb{Q}$ and a rational attack length $m \in \mathbb{Q}$. Furthermore by taking the lowest common multiple of all denominators in the set $D(E) \cup \{T, m\}$ we can scale the problem up by multiplying by this number to leave a game limited to the integers. So for the remainder of our study on $G(Q, T, m)$ we will assume the distance function is such that $D : E \to \mathbb{N}$, $T \in \mathbb{N}$ and $m \in \mathbb{N}$. In this case the time-horizon is now $\mathcal{J} = \{0, ..., T-1\}$. We then note that since a pure patrollers strategy in general move,wait form is described as list of pairs, as in equation (6.3), (j_i, ν_i) has $\nu_i \in \mathbb{N}$.

In chapter 4, section 4.4 we briefly discussed that some edges have no effect on the value of the classic patrolling game. Here we formally define this idea for the classic patrolling game as *superfluous edges* and show how the distance mapping of these edges is irrelevant to the the value of the patrolling game with distance.

Definition 6.1.2. For the (classic) patrolling game $G = ((N, E), T, m)$ we call an edge $e \in E$ *superfluous* if $V((N, E), T, m) = V((N, E \setminus \{e\}), T, m)$. Furthermore, we call a set of edges $F \subset E$ a *superfluous set* if $V((N, E), T, m) = V((N, E \setminus F), T, m)$.

That is the removal of a superfluous edge does not affect the value of the classic patrolling game, and in other words if there is an optimal patroller strategy which does not use the edge in the game then the edge is superfluous. As an example consider the game $G(K_n, T, m)$,on the complete graph of n nodes, then as any random Hamiltonian strategy $\boldsymbol{\pi}_{\mathrm{rH}}$ is optimal it is easy to see that the removal of any edge results in the new game having the same value, as it remains Hamiltonian. While all edges are initially superfluous this does not mean the set E is a superfluous set. To construct a superfluous set one must find a superfluous edge add it to the set and then proceed to find a superfluous edge in the resultant game with that edge removed. An example of a superfluous set can be found by comparing $G(K_n, T, m)$ to $G(C_n, T, m)$, the patrolling game on the cyclic graph of n nodes. In chapter 2, section 2.2.3 we saw that

$$V(K_n, T, m) = V(C_n, T, m) = \frac{m}{n},$$

for all $n \geq 1$, $m \geq n$ and for all $T \geq m$. Letting E_1 be the set of edges in K_n and E_2 be the set of edges in C_n we have that $F = E_1 \setminus E_2$ is a superfluous set for the game $G(K_n, T, m)$. One can consider the construction of a superfluous set with the same cardinality by the above method, adding in total $\frac{n(n-3)}{2}$ edges to a set G and with appropriate choices it is possible that $G = F$.

When an edge is not superfluous, its removal has a varying effect on the value of the game, depending on the importance of that edge in the patroller's optimal strategy. For example the removal of any edge from the graph in the game $G(C_n, T, m)$ results in the game $G(L_n, T, m)$ which has a widely different value dependent on the attack length m (see figure 2.3.3 for an example of the difference for various attack lengths). We can now consider using the superfluous set/edge for a classic patrolling game to form patrolling game with edge distances which can still utilise the same optimal strategies.

Lemma 6.1.3. *For the (classic) patrolling game $G(Q, T, m)$ $(Q = (N, E))$ with a superfluous set $F \subset E$ and the patrolling game with edge distances $G = (Q, D, T, m)$ such that $D(e) = 1$ for all $e \in E \setminus F$ we have*

$$V(Q, D, T, m) = V(Q, T, m) \quad \forall m \geq 1, \ \forall T \geq m.$$

The proof of lemma 6.1.3 immediately follows from using the optimal strategies of the classic patrolling game in the patrolling game with edges distances.

Proof. First we aim to prove that an optimal policy of the classic game does not need to use the a superfluous edge. As

$$V((N,E),T,m) = V((N,E\setminus F),T,m)$$
$$\iff \max_{\boldsymbol{\pi}\in\Pi((N,E),T,m)} \min_{\boldsymbol{\phi}\in\Phi((N,E),T,m)} P(\boldsymbol{\pi},\boldsymbol{\phi}) = \max_{\boldsymbol{\pi}\in\Pi((N,E\setminus F),T,m)} \min_{\boldsymbol{\phi}\in\Phi((N,E\setminus F),T,m)} P(\boldsymbol{\pi},\boldsymbol{\phi})$$

we have that optimal strategies $\boldsymbol{\pi}^* \in \Pi(N,E\setminus F,T,m) \subset \Pi(N,E\setminus F,T,m)$ and $\boldsymbol{\phi}^* \in \Phi(N,E\setminus F,T,m) = \Phi(N,E),T,m)$ for the game $G((N,E\setminus F,T,m)$ which can be used in $G((N,E),T,m)$ and hence no superfluous edge is used.

The optimal patroller strategy $\boldsymbol{\pi}^*$ is feasible in the game $G(Q,D,T,m)$ as any edge used $e \in E\setminus F$ has $D(e) = 1$. By using the strategy $\boldsymbol{\pi}^* \in \Pi(Q,D,T,m)$ we have

$$\begin{aligned} V(Q,D,T,m) \geq V_{\boldsymbol{\pi}^*,\bullet}(Q,D,T,m) &= \max_{a\in\mathcal{A}(Q,D,T,m)} P(\boldsymbol{\pi}^*,a) \\ &= \max_{a\in\mathcal{A}(Q,T,m)} P(\boldsymbol{\pi}^*,a) = V(Q,T,m). \end{aligned} \tag{6.4}$$

By using lemma 6.1.1 to compare the game $G(Q,T,m) = G(Q,1,T,m)$ to the game $G(Q,D,T,m)$ we get

$$V(Q,D,T,m) \leq V(Q,1,T,m) = V(Q,T,m) \tag{6.5}$$

The lower bound in equation (6.4) and the upper bound in equation (6.5) are equal and thus $V(Q,D,T,m) = V(Q,T,m)$. □

Lemma 6.1.3 immediately gives us the value and hence optimal strategies for games with edge distance when we take a classic game with a known value and change distances on any edge in a superfluous set. For example, lemma 6.1.3 tells us the value of $G(Q,D,T,m)$ for which $Q \in \mathcal{H}$ and for which a Hamiltonian cycle, H exists such that $D((H(i),H(i+1)) = 1$ for all $i = 0,...,|N|-1$. It is possible to get a stronger version of lemma 6.1.3 which compares the distances on edges for various patrolling games with edges distances.

Lemma 6.1.4. *For the game $G(Q,D,T,m)$ and a set $F \subset E$ for which the game $G(Q,D,T,m)$ has an optimal patroller strategy $\boldsymbol{\pi}^* \in \Pi(Q,D,T,m)$ which does not use any edge $e \in F$ then*

$$V(Q,D',T,m) = V(Q,D,T,m) \quad \forall m \geq 1,\ \forall T \geq m,$$

where in game $G(Q,D',T,m)$, D' is any distance function such that $D'(e) = D(e)$ for $e \in E\setminus F$.

The proof of lemma 6.1.4 is left to appendix C.1, since it follows from the same idea as the proof of lemma 6.1.3 in which we think about the set F being 'superfluous'. Lemma 6.1.4 allows us make a stronger statement about Hamiltonian graphs.

Corollary 6.1.5. *For the games $G(Q,D,T,m)$ and $G(Q,D',T,m)$ where $Q=(N,E)\in\mathcal{H}$ and there exists some Hamiltonian cycle H for Q such that $D((H(i),H(i+1)))=D'((H(i),H(i+1)))$ for all $i\in\{0,...,|N|-1\}$ then*

$$V(Q,D',T,m)=V(Q,D,T,m)\quad \forall m\geq 1,\ \forall T\geq m.$$

The proof of corollary 6.1.5 follows immediately by noting that the conditions of lemma 6.1.4 are satisfied by using any $F\subset E\setminus H(\{0,...,|N|-1\})$.

We have seen the value of the patrolling game with edge distances remains the same, when only edges which are superfluous have their distance increased. This is because the patroller need not use those edges. Similarly, we can look the idea of using an attacker strategy which does not use a set of nodes. We start by first looking at removing nodes on the graph in a similar process to that of node-identification from section 2.3.1. However, unlike node-identification this process will remove nodes while retaining the connections (and distances of said connections) provided through the node being removed. We call this process *Sublimation* and define it with our distance on edges. This process is an extension of the smoothing operator (which can only be applied to nodes with degree two) ([47]).

Definition 6.1.6. The graphical operator of *node-sublimation* maps a (simple undirected) graph $Q=(N,E)$ and a distance edge function D to another graph $Q'=(N',E')$ and a distance edge function D' by sublimating a node n written as $\mathcal{Q}^s(Q,D,n)=(Q',D')$. The resultant graph Q' is such that $N'=N\setminus\{n\}$ and $E'=E\setminus\{(i,n)|(i,n)\in E\}\cup\{(i,j)|(i,n),(j,n)\in E,(i,j)\notin E\}$ and the resultant distance edge mapping D' is given by

$$D'((i,j))=\begin{cases}\min(D((i,n))+D((j,n)),D(i,j)) & \text{if } (i,n),(j,n)\in E,(i,j)\in E\\ D((i,n))+D((j,n)) & \text{if } (i,n),(j,n)\in E,(i,j)\notin E\\ D((i,j)) & \text{otherwise.}\end{cases}$$

That is node-sublimation removes all incident edges to n and adds edges between any pair of nodes adjacent to n to form the resultant graph, with the distance mapping on the new edges preserving the distance of a walk between the pair of nodes passing through n. We note that to keep the resultant graph simple and undirected while preserving the distance we only keep the edge with the minimal distance between any two nodes.

Figure 6.1.1 shows an example of how node sublimation can be used to reduce the number of nodes we need consider for a graph, while modifying the distances to maintain the same distances between remaining nodes. In the figure edges $(2,i)$ are removed for $i=1,3,4,5$ and replaced by all combinations of edges between nodes $1,3,4,5$. The distance of these new edges is given by the sum of the two edges used through node 2, as an example the edge $(1,3)$ has a distance of $D'((1,3))=D((1,2))+D((2,3))=2+5=7$. Note that the edge $(4,5)\in Q$ had a distance of 5 and is ‘replaced’ by the new edge of distance 4.

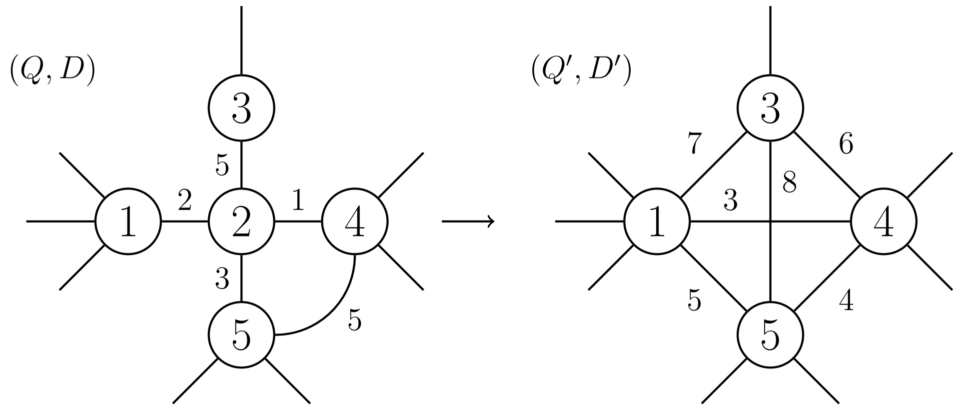


Figure 6.1.1: A graph Q with edge distances D as seen on the edges, showing the process of sublimating node 2 creating the graph Q' with edge distances D'.

As with superfluous edges, we can get useful results when using node-sublimation. These results will immediately give the value of related patrolling games with edges distances.

Lemma 6.1.7. *For the game $G(Q, D, T, m)$ for any graph $Q = (N, E)$, for all $n \in N$, for all D, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V(Q', D', T, m) \geq V(Q, D, T, m),$$

where $(Q', D') = \mathcal{Q}^s(Q, D, n)$. That is $G(Q', D', T, m)$ is the node-sublimated game using node n.

The proof of lemma 6.1.7 follows by considering using the optimal patroller strategy for the game $G(Q, D, T, m)$ augmented for use in the game $G(Q', D', T, m)$.

Proof. Let $\boldsymbol{\pi}^* \in \Pi(Q, D, T, m)$ be an optimal strategy for the game $G(Q, D, T, m)$ playing pure patrolling strategies $W_1, ..., W_p$ with non-zero probability i.e. $\pi^*_{\beta_1(W_x)} > 0$ for all $i \in \{1, ..., p\}$ such that

$$\sum_{x=1}^{p} \pi^*_{\beta_1(W_x)} = 1.$$

Let $\omega_1, ..., \omega_p$ be the general move,wait form of the pure patrolling strategies $W_1, ..., W_p$ and then look at the augmented walks $\widetilde{\omega}_1, ..., \widetilde{\omega}_p$ which 'pass through' the node n to make them feasible in the game $G(Q', D', T, m)$. To do this let us first define a sequence of indices for us to keep by letting $s_0 = 0$ and using the recursive rule

$$s_{i+1} = \begin{cases} s_i + 1 & \text{if } j_{s_i+1}(\omega_x) \neq n, \\ s_i + 2 & \text{if } j_{s_i+1}(\omega_x) = n, \end{cases}$$

to generate a finite sequence $s_1, ..., s_L$ where $L = \{l \mid j_l(\omega_x) \neq n\}$ for a given walk ω_x for $x = 1, ..., p$.

Then we can form augmented walks $\widetilde{\omega}_1, ..., \widetilde{\omega}_p$ such that for all $x = 1, ..., p$ we let $k(\widetilde{\omega}_x) = k(\omega_x) - L$ and if $k(\widetilde{\omega}_x) \neq 0$ we let $j_i(\widetilde{\omega}_x) = j_{s_i}(\omega_x)$ and

$$\nu_i(\widetilde{\omega}_x) = \begin{cases} \nu_{s_i-1}(\omega_x) + \nu_{s_i}(\omega_x) & \text{if } j_{s_i-1}(\omega_x) = n \\ \nu_{s_i}(\omega_x) & \text{otherwise.} \end{cases}$$

Otherwise if $k(\widetilde{\omega}_x) = 0$ we let $\widetilde{\omega}_x$ be any feasible walk in $\mathcal{W}(Q', D', T, m)$.

These augmented pure patroller strategies in general move,wait form $\widetilde{\omega}_1, ..., \widetilde{\omega}_1$ have normal form equivalents $\widetilde{W}_1, ..., \widetilde{W}_p \in \mathcal{W}(Q', D', T, m)$ and hence playing $\widetilde{W}_x$ with probability $\pi^*_{\beta_1(W_x)}$ for $x = 1, ..., p$ creates a mixed patroller strategy $\widetilde{\boldsymbol{\pi}} \in \Pi(Q', D', T, m)$. We have that $P(\widetilde{\boldsymbol{\pi}}, a) = P(\boldsymbol{\pi}^*, a)$ for all $a \in \mathcal{A}(Q', D', T, m) \subset \mathcal{A}(Q, D, T, m)$ and hence

$$\begin{aligned} V(Q', D', T, m) &\geq \min_{a \in \mathcal{A}(Q', D', T, m)} P(\widetilde{\boldsymbol{\pi}}, a) = \min_{a \in \mathcal{A}(Q', D', T, m)} P(\boldsymbol{\pi}^*, a) \\ &\geq \min_{a \in \mathcal{A}(Q, D, T, m)} P(\boldsymbol{\pi}^*, a) = V(Q, D, T, m). \end{aligned}$$

□

We also consider using attacker strategies from the game $G(Q, D, T, m)$ in the game $G(Q', D', T, m)$ which do not attack the node which is chosen for node-sublimation.

Lemma 6.1.8. *For the game $G(Q, D, T, m)$ for any graph Q, for all $n \in N$, for all D, for all $m \geq 1$, for all $T \geq m$ and for any $\boldsymbol{\phi} \in \Phi(Q, D, T, m)$ such that $\varphi_{n,\tau} = 0$ for all $\tau \in \mathcal{T}$ we have*

$$V_{\bullet,\boldsymbol{\phi}}(Q', D', T, m) = V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m),$$

where $(Q', D') = \mathcal{Q}^s(Q, D, n)$. That is $G(Q', D', T, m)$ is the node-sublimated game using node n.

The proof of lemma 6.1.8 follows by considering the best response from the patroller given the attacker chooses $\boldsymbol{\phi}$ in both games $G(Q', D', T, m)$ and $G(Q, D, T, m)$.

Proof. We first show that $V_{\bullet,\boldsymbol{\phi}}(Q', D', T, m) \leq V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m)$, as for any $W' \in \mathcal{W}(Q', D', T, m)$ there exists some $W \in \mathcal{W}(Q, D, T, m)$ with $W(t) = W'(t)$ for all $t \in \{s \in \mathcal{J} \mid W'(t) \in N \setminus \{n\}\}$ and hence for any $W' \in \mathcal{W}(Q', D', T, m)$ there exists some $W \in \mathcal{W}(Q, D, T, m)$ such that $P(W', \boldsymbol{\phi}) = P(W, \boldsymbol{\phi})$. Therefore

$$\begin{aligned} V_{\bullet,\boldsymbol{\phi}}(Q', D', T, m) &= \min_{W' \in \mathcal{W}(Q', D', T, m)} P(W', \boldsymbol{\phi}) \\ &\leq \min_{W \in \mathcal{W}(Q, D, T, m)} P(W, \boldsymbol{\phi}) = V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m). \end{aligned} \tag{6.6}$$

We secondly show that $V_{\bullet,\boldsymbol{\phi}}(Q', D', T, m) \geq V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m)$, as similarly for any $W \in \mathcal{W}(Q, D, T, m)$ there exists some $W' \in \mathcal{W}(Q', D', T, m)$ with $W'(t) = W(t)$

for all $t \in \{s \in \mathcal{J} \mid W(t) \in N \setminus \{n\}\}$ and hence for any $W \in \mathcal{W}(Q, D, T, m)$ there exists some $W' \in \mathcal{W}(Q', D', T, m)$ such that $P(W', \boldsymbol{\phi}) = P(W, \boldsymbol{\phi})$. Therefore

$$\begin{aligned} V_{\bullet,\boldsymbol{\phi}}(Q, D, T, m) &= \min_{W \in \mathcal{W}(Q,D,T,m)} P(W, \boldsymbol{\phi}) \\ &\leq \min_{W' \in \mathcal{W}(Q',D',T,m)} P(W', \boldsymbol{\phi}) = V_{\bullet,\boldsymbol{\phi}}(Q', D', T, m). \end{aligned} \tag{6.7}$$

So with equations (6.6) and (6.7) we conclude the proof. □

Theorem 6.1.9. *For the game $G(Q, D, T, m)$ for any graph Q, for all $n \in N$, for all D, for all $m \geq 1$, for all $T \geq m$ if there exists an optimal attack strategy $\boldsymbol{\phi}^*$ for the game $G(Q, D, T, m)$ such that $\varphi^*_{n,\tau} = 0$ for all $\tau \in \mathcal{T}$ then we have*

$$V(Q', D', T, m) = V(Q, D, T, m),$$

where Q', D' are such that $\mathcal{Q}^s(Q, D, n)$. That is $G(Q', D', T, m)$ is the node-sublimated game using node n.

Proof. By lemma 6.1.7 we know that $V(Q', D', T, m) \geq V(Q, D, T, m)$ and using $\boldsymbol{\phi}^*$ in lemma 6.1.8 we get that

$$V(Q', D', T, m) \leq V_{\bullet,\boldsymbol{\phi}^*}(Q', D', T, m) = V_{\bullet,\boldsymbol{\phi}^*}(Q, D, T, m) = V(Q, D, T, m).$$

Hence we have $V(Q', D', T, m) = V(Q, D, T, m)$. □

Theorem 6.1.9 immediately gives us the value of a patrolling games with edge distances created from a node-sublimation on a node which is not used in the optimal attacker strategy for another patrolling game with edge distances. The intuitive reason behind the lemma is that node-sublimation only reduces the complexity of the patrolling game with edge distances.

In addition successive node-sublimations can be used each using Theorem 6.1.9 to achieve even more reduced complexity models. For notation purposes we let $\mathcal{Q}^s(Q, D, F)$ represent the repeated node-sublimation of the initial graph and distance (Q, D) for each $n \in F$ as long as the order of node-sublimation is irrelevant. A prominent example of which is when for all $n \in F$, the degree of node n is 2.

We can achieve an analogous result to that of the decomposition result for the classic game, as given in lemma 2.3.14. To do so we first define the analogous decomposition strategy.

Definition 6.1.10. For the game $G(Q, D, T, m)$ with a decomposition of Q into $Q_1, ..., Q_R$, we form the subgraph games $G(Q_1, D, T, m), ..., G(Q_R, D, T, m)$ with optimal patroller strategies, $\boldsymbol{\pi}^*_1, ..., \boldsymbol{\pi}^*_R$. A *decomposition* patroller strategy using the decomposition above $\boldsymbol{\pi}_{\text{Dec}}$ is such that $\pi_{\beta_1(W)} = \sum_{i=1}^{R} p_i \pi^*_{i,\beta_1(W)}$, where

$$p_i = \frac{1}{V(Q_i, D, T, m) \sum_{r=1}^{R} \frac{1}{V(Q_r, D, T, m)}},$$

for all $i \in \{1, .., R\}$.

Lemma 6.1.11. *For the game $G(Q,D,T,m)$ for any graph Q with any decomposition into Q_i for $i=1,..,R$ ($Q=\bigcup_{i=1}^{R} Q_i$), for all $m\geq 1$ and for all $T\geq m$ we have*

$$V(Q,D,T,m)\geq \frac{1}{\sum_{i=1}^{R}\frac{1}{V(Q_i,D,T,m)}},$$

where the lower bound on $V(Q,D,T,m)$ is achieved by the patroller choosing a decomposition patroller strategy $\boldsymbol{\pi}_{Dec}$ using the decomposition Q_i for $i=1,..,R$. Moreover if the subgraphs Q_i for $i=1,..,R$ are disjoint and disconnected we have

$$V(Q,D,T,m)=\frac{1}{\sum_{i=1}^{R}\frac{1}{V(Q_i,D,T,m)}}.$$

The proof of lemma 6.1.11 follows by having the patroller choose $\boldsymbol{\pi}_{\mathrm{Dec}}$ with the moreover part following by forming a similarly defined decomposition attacker strategy.

Proof. Considering $\boldsymbol{\pi}_{\mathrm{Dec}}$ we have

$$\begin{aligned}
V(Q,D,T,m) &\geq \min_{a\in\mathcal{A}(Q,D,T,m)} P(\boldsymbol{\pi}_{\mathrm{Dec}},a)\geq \min_{i\in\{1,\dots,R\}}\min_{a\in\mathcal{A}(Q_i,D,T,m)} P(\boldsymbol{\pi}_{\mathrm{Dec}},a)\\
&= \min_{i\in\{1,\dots,R\}}\min_{a\in\mathcal{A}(Q_i,D,T,m)} p_i P(\boldsymbol{\pi}_i^*,a)\\
&= \min_{i\in\{1,\dots,R\}} p_i V(Q_i,D,T,m) = \min_{i\in\{1,\dots,R\}} \frac{1}{\sum_{i=1}^{R}\frac{1}{V(Q_i,D,T,m)}}\\
&= \frac{1}{\sum_{i=1}^{R}\frac{1}{V(Q_i,D,T,m)}}. \qquad (6.8)
\end{aligned}$$

When Q_i are all disjoint and disconnected subgraphs of Q then we can consider the attacker using a decomposed attacker strategy $\boldsymbol{\phi}_{\mathrm{Dec}}$. Let $\boldsymbol{\phi}_i^*\in\Phi(Q_i,D,T,m)$ be optimal in the game $G(Q_i,D,T,m)$ for $i=1,...,R$ then let $\boldsymbol{\phi}_{\mathrm{Dec}}$ be such that $\phi_{\beta_2(a)}=\sum_{i=1}^{R} p_i\phi^*_{i,\beta_2(a)}$ for $i=1,..,R$. Then as $\mathcal{W}(Q,D,T,m)=\bigcup_{i=1}^{R}\mathcal{W}(Q_i,D,T,m)$ we have

$$\begin{aligned}
V(Q,D,T,m) &\leq \max_{W\in\mathcal{W}(Q,D,T,m)} P(W,\boldsymbol{\phi}_{\mathrm{Dec}}) = \max_{i\in\{1,\dots,R\}}\max_{W\in\mathcal{W}(Q_i,D,T,m)} P(W,\boldsymbol{\phi}_{\mathrm{Dec}})\\
&= \max_{i\in\{1,\dots,R\}}\max_{W\in\mathcal{W}(Q_i,D,T,m)} p_i P(W,\boldsymbol{\phi}_i^*)\\
&= \max_{i\in\{1,\dots,R\}} p_i V(Q_i,D,T,m) = \min_{i\in\{1,\dots,R\}} \frac{1}{\sum_{i=1}^{R}\frac{1}{V(Q_i,D,T,m)}}\\
&= \frac{1}{\sum_{i=1}^{R}\frac{1}{V(Q_i,D,T,m)}}. \qquad (6.9)
\end{aligned}$$

Hence by equation (6.8) and (6.9) we have that

$$V(Q, D, T, m) = \frac{1}{\sum_{i=1}^{R} \frac{1}{V(Q_i, D, T, m)}}.$$

$\square$

6.1.3 Value of patrolling games with edge distances

In this section, by using theorem 6.1.9 and lemma 6.1.11 in addition to classic patrolling games with known value and optimal strategy, we can arrive at values for patrolling games with edge distances. In addition we provide solutions to a model of multiple cities connected to a central hub at various distances apart, described and modelled in section 4.3 as $G(S_n^{\boldsymbol{k}}, T, m)$, by using edge distances in the modelling.

Considering that $G(Q, 1, T, m) \equiv G(Q, T, m)$ we can consider node-sublimation of $(Q, 1)$ at any node n giving us $\mathcal{Q}^s(Q, 1, n) = (Q', D')$ for which we have the lower bound

$$V(Q', D', T, m) \geq V(Q, T, m),$$

by relying on the optimal patrolling strategy to the classic game. While node-sublimation is great at reducing the amount of patroller and attacker strategies, it can remove some features and we may have the patroller does strictly better in the game $G(Q', D', T, m)$ compared to the game $G(Q, T, m)$ if every optimal attacker strategy uses the node n which was chosen for the node-sublimation. Moreover, we can consider repeated node-sublimation on a game $G(Q, 1, T, m)$ to still achieve a lower bound on the resultant game. However, as with a single node-sublimation operation, node-sublimating a node n which is used in all optimal attacker strategies makes the lower bound strict as it restricts the attackers choice of positions.

Consider the common scenario of patrolling a border which has been modelled as the line graph L_n, with a distance of $n-1$ between the two diametric nodes 1 and n. Node-sublimation can be repeatedly used to create the graph $L'_n = C_2$, which consists of these two diametric nodes with a single edge $(1, n)$ with $D((1, n)) = n-1$ between them. Solving the resultant patrolling game with edges distances $G(L'_n, n-1, T, m)$ is easy to solve in comparison to classic patrolling $G(L_n, T, m)$. For the game $G(L'_n, n-1, T, m)$ when $m \geq n-1$ we can use theorem 6.1.9, as the nodes which are sublimated to create L'_n from L_n are the internal nodes $(2, .., n-1)$, which are not attacked in the optimal strategy (time-limited diametric attack $\boldsymbol{\phi}_{\text{tdi}}$) of the classic game $G(L_n, T, m)$. Hence for all $n \geq 3$, for all $m \geq n-1$, for all $T \geq m+n-2$ we have

$$V(L'_n, n-1, T, m) = V(L_n, T, m) = \frac{m}{2(n-1)}.$$

For $m < n-1$ the classic patrolling game on the line $G(L_n, T, m)$ required bespoke attacker strategies, which are hard to find and place pure attacks at internal nodes and so theorem 6.1.9 cannot be used to easily get the value of the game $G(L'_n, n-1, T, m)$. However, it is much simpler to get the solution to the game $G(L'_n, n-1, T, m)$ for $m < n-1$ and now hard to find bespoke attacker strategies are needed. Consider for $m < n-1$ the attack strategy $\boldsymbol{\phi} \in \Phi(L'_n, n-1, T, m)$ such that $\varphi_{1,t} = \frac{1}{2}$ and $\varphi_{n,t} = \frac{1}{2}$ for a given fixed arbitrary $t \in \mathcal{T}$, then it is clear that as the distance between the two nodes is greater than m that

$$V(L'_n, n-1, T, m) \leq \frac{1}{2},$$

as any pure patroller can only see one of the pure attacks (either $(1,t)$ or (n,t) exclusively). For the patroller strategy consider a choose and wait style strategy $\boldsymbol{\pi}$ such that $\pi_{\beta_1(W)} = \frac{1}{2}$ if $W(t) = 1$ for all $t \in \mathcal{T}$ or $W(t) = n$ for all $t \in \mathcal{T}$. Then it is clear that

$$V(L'_n, n-1, T, m) \geq \frac{1}{2},$$

as any pure attack must either be at node 1 or n and for any commencement time there is probability of $\frac{1}{2}$ they choice the node the patroller chose. Therefore

$$V(L'_n, n-1, T, m) = \begin{cases} \frac{m}{2(n-1)} & \text{for } m \geq n-1, \\ \frac{1}{2} & \text{for } m < n-1. \end{cases}$$

It is worth noting that while the game $G(L'_n, n-1, T, m)$ is easier to solve than $G(L_n, T, m)$, such a change to the classic game $G(L_n, T, m)$ does not necessarily accurately model the scenario of having to protect a border as it assumes the only locations that may be attacked are the end points and that the border has a impenetrable wall protecting the border which can be crossed at its ends. Perhaps a more realistic scenario modelled by $G(L'_n, n-1, T, m)$ is that of two cities connected by a highway, of distance $n-1$, with police patrolling between them.

As shown above, we can consider node-sublimation in order to make models previously using the classic patrolling game easier to solve by converting them into patrolling games with edge distances, noting that the new model has a restriction on the attacker's choice of locations. As stated in theorem 6.1.9 when these restrictions do not affect the attacker the new model will admit the same answers as the classic model, however in the other case while the value will be higher it is often much easier to find than of that for the classic model.

In chapter 4, section 4.3, we did not managed to solve the classic patrolling game on the generalised star graph $G(S_n^{\boldsymbol{k}}, T, m)$ for all attack lengths m due to the complexity of attacker solutions when the minimal full-node cycle strategy can be improved. We will now show that by using node-sublimation it is possible to get solutions when all internal branch nodes are node-sublimated.

We can use node-sublimation to get solutions for all attacks and in doing so provide a solution to the scenario of a central hub location connected to multiple cities at varying distances from the central hub. We note that such a node-sublimation does mean that the new game no longer models having a central hub and multiple borders which can internally be attacked. However such a node-sublimation is still appropriate for the model of a central hub city connected to various other cities. We will first define this node sublimated graph, a star graph, with an appropriate edge distance function arising from a node-sublimation of a generalised star graph $S_n^{\boldsymbol{k}}$.

Definition 6.1.12. The *distant general star* (graph and edge distance) is the graph $\widetilde{S}_n^{\boldsymbol{k}}(\equiv S_n)$, along with an edge distance function $D_{n,\boldsymbol{k}}$, for some $n \in \mathbb{N}$ and some $\boldsymbol{k} \in \mathbb{N}^n$, in which the node set is $N = \{c, *_1, ..., *_n\}$, the edge set is $E = \{(c, *_i) \mid i = 1, ..., n\}$ and $D((c, *_i)) = k_i + 1$ for $i = 1, ..., n$.

As with the generalised star graph we assume that $\boldsymbol{k}$ is in descending order and so $(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}) = \mathcal{Q}^s(S_n^{\boldsymbol{k}}, 1, F)$, where $F = \{(*_{i,j} \mid i \in \{1, ..., k_j\}$ for all $j \in \{1, .., n\}\}$ is the set of internal branch nodes. That is a distant general star is a node-sublimated generalised star graph. Figure 6.1.2 shows an example of the node-sublimation of generalised star graph to a distant general star graph with the appropriate edge distance shown on the edges.

From theorem 6.1.9 and lemma 4.3.6 we know that for the distant general star $(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}})$ for all $n \geq 3$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $m \geq 2(k_{\max} + 1)$ and for all $T \geq 2k_{\max} + m + 1$ that

$$V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) = V(S_n^{\boldsymbol{k}}, T, m) = \frac{m}{2(n + k_{\text{sum}})}. \tag{6.10}$$

We are able to use theorem 6.1.9 to arrive at this value for $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$ as an optimal strategy for $(S_n^{\boldsymbol{k}}, T, m)$ is the type-centred attacker strategy $\boldsymbol{\phi}_{\text{tc}}$ which doesn't use any potential attacks at nodes in the set of internal branch nodes F. For $m < 2(k_{\max} + 1)$ while we did not find solutions to $G(S_n^{\boldsymbol{k}}, T, m)$ as it would require bespoke attacker strategies, it is much easier to find solutions to $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$. We will see that the optimal solution depends on the distance of each branch $D_{n,\boldsymbol{k}}(c, *_i)$ for $i = 1, ..., n$ compared to the attack length m.

Let use first start by determining a distinct ordering on the branch extensions k_i forming the *descending ordering* $k_{(1)}, ..., k_{(q)}$ such that $k_{(1)} = k_{\max}$ and $k_{(i)} = \max\{k_s \mid k_s \leq k_{(i-1)}, s = 1, ..., n\}$ for $2 \leq i \leq q$ with $k_{(q)} = \min\{k_s \mid s = 1, ..., n\}$. Along with this descending ordering of the branch extensions we define the *repeat count* of the i^{th} order as $b_i = |\{k_s \mid k_s = k_{(i)}\}|$, that is the number of times $k_{(i)}$ is seen in $\boldsymbol{k}$. For example if $\boldsymbol{k} = (4, 2, 6, 4, 3)$ then we would have $k_{(1)} = 6, k_{(2)} = 4, k_{(3)} = 3, k_{(4)} = 2$ and $b_1 = 1, b_2 = 2, b_3 = 1, b_4 = 1$. With our descending order along with the number of times they are repeated we now give the optimal solution to the game $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$ for $2(k_{(i+1)} + 1) \leq m \leq 2(k_{(i)} + 1)$ for each $i = 1, ..., q - 1$.

We now find a lower bound on the value of the game $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$ by considering a decomposition into multiple subgraph games dependent where we leave the graph just consisting of a branch node node $*_j$ is such that $k_j \geq k_{(i)}$. That

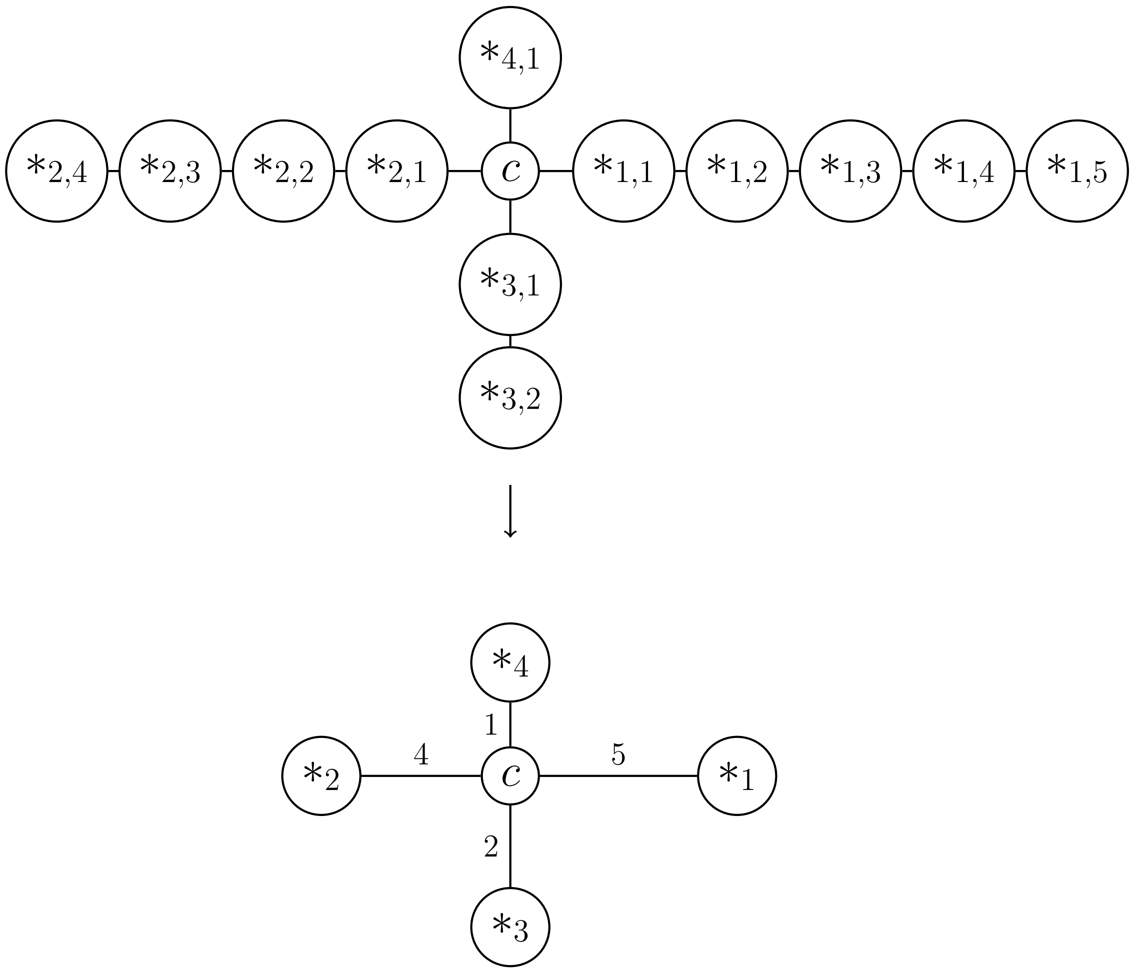


Figure 6.1.2: The generalised star graph $S_4^{4,3,1,0}$ (with a uniform unitary distance function) sublimated to the distant star graph $\widetilde{S}_4^{4,3,1,0}$. The distant star graph has a distance function with $D_{4,(4,3,1,0)}((c, *_i)) = k_i + 1$, with distances seen on edges in the figure.

is for $2(k_{(i+1)}+1) \leq m \leq 2(k_{(i)}+1)$ for each $i = 1, ..., q-1$ we decompose the game $G(\widetilde{S}_{\hat{n}}^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$ into

$$G(\widetilde{S}_n^{\hat{\boldsymbol{k}}}, D_{n,\hat{\boldsymbol{k}}}, T, m),$$

and $\sum_{x=1}^{i} b_x$ lots of the game

$$G((\{1\}, \emptyset), 1, T, m) \equiv G((\{1\}, \emptyset), T, m),$$

where $\hat{n} = n - \sum_{x=1}^{i} b_x$ and $\hat{\boldsymbol{k}} \in \mathbb{N}^{\hat{n}}$ is the vector $\boldsymbol{k}$ with the first $\sum_{x=1}^{i} b_x$ entries truncated (any entry $k_x > k_{(i+1)}$ is therefore removed). By using lemma 6.1.11 and theorem 6.1.9 and knowing that in the game $V(S_n^{\hat{\boldsymbol{k}}}, D_{n,\boldsymbol{k}}, T, m)$ there exists a $\boldsymbol{\phi}$ which does not attack any internal branch nodes we get that for all $n \geq 3$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $2(k_{(i+1)}+1) \leq m \leq 2(k_{(i)}+1)$ for all $i = 1, ..., q-1$ and for all $T \geq m$ we have

$$\begin{aligned} V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) &\geq \frac{1}{V(\widetilde{S}_n^{\hat{\boldsymbol{k}}}, D_{n,\hat{\boldsymbol{k}}}, T, m)^{-1} + \left(\sum_{x=1}^{i} b_x\right) V((\{1\}, \emptyset), T, m)^{-1}} \\ &= \frac{1}{V(\widetilde{S}_n^{\hat{\boldsymbol{k}}}, T, m)^{-1} + \left(\sum_{x=1}^{i} b_x\right) V((\{1\}, \emptyset), T, m)^{-1}} \\ &= \frac{1}{\frac{2\sum_{j=i+1}^{q} b_j(k_j+1)}{m} + \left(\sum_{x=1}^{i} b_x\right)} \\ &= \frac{m}{m\left(\sum_{x=1}^{i} b_x\right) + 2\sum_{j=i+1}^{q} b_j(k_j+1)}. \end{aligned} \tag{6.11}$$

While previous results with some carefully chosen decomposition gives the lower bound as in equation 6.11, to find an equal upper bound we need to develop a new attacker strategy.This attacker strategy will be adapted from the typed time-centre attack strategy (see section 4.3), which is optimal for $m \geq 2(k_{\max}+1)$.

Definition 6.1.13. The *distance from centre* (DFC) attack strategy $\boldsymbol{\phi}_{\text{DFC}}$, in the attack length range of $2(k_{(i+1)}+1) \leq m \leq 2(k_{(i)}+1)$ is given by

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{m\sum_{x=1}^{i} b_x + 2\sum_{x=i+1}^{q} b_x(k_x+1)} & \text{for } j = *_r, \tau \in T_r \text{ for some } r \in \{1, ..., n\}, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$T_r = \begin{cases} \{0, ..., m-1\} & \text{if } m < 2(k_r+1), \\ \{\hat{m} - k_r - 1, ..., \hat{m} + k_r\} & \text{if } m \geq 2(k_r+1), \end{cases}$$

in which $\hat{m} = \left\lfloor \frac{m}{2} \right\rfloor$.

That is the strategy ϕ_{DFC} chooses each branch end $*_r$ with equal probability and then given $*_r$ is chosen the commencement time is chosen uniformly from T_r. Creating such a bespoke strategy adaptation is a lot simpler in the game $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$, compared to the game $G(S_n^{\boldsymbol{k}}, T, m)$, as the set of nodes which can be attacked is much smaller. The DFC attacker strategy ϕ_{DFC} restricts the number of attacks placed at node $*_r$ for any $r \in \{1, ..., n\}$, restricting the effectiveness of the pure patroller who collects all attacks at $*_r$. By using the DFC we are able to get the following bound and hence the know that it is an optimal attacker strategy.

Lemma 6.1.14. *For the game $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$, for all $n \in \mathbb{N}$, $\boldsymbol{k} \in \mathbb{N}^n$, for all $T \geq 2m$, for all $2(k_{(i+1)} + 1) \leq m \leq 2(k_{(i)} + 1)$ we have*

$$V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) \leq \frac{m}{m \sum_{j=1}^{i} b_j + 2 \sum_{j=i+1}^{q} b_j (k_j + 1)}.$$

The proof of lemma 6.1.14 follows from the performance of ϕ_{DFC}, which means we need to consider move wait walks.

Proof. As ϕ_{DFC} is such that $\varphi_{c,\tau} = 0$ for all $\tau \in \mathcal{T}$ we have, by lemma 6.1.8, that

$$V_{\bullet,\phi_{\mathrm{DFC}}}(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) = V_{\bullet,\phi_{\mathrm{DFC}}}(Q, D, T, m),$$

where Q is isomorphic to K_n with $N = \{*_1, ..., *_n\}$ and $E = \{(*_{r'}, *_{r'}) \in N^2 \mid r, r' \in \{1, ..., n\}, r \neq r'\}$ and $D((*_r, *_{r'})) = k_r + k_{r'} + 2$. So we aim to find $V_{\bullet,\phi_{\mathrm{DFC}}}(Q, D, T, m)$ and we have that

$$V_{\bullet,\phi_{\mathrm{DFC}}}(Q, D, T, m) = \max_{W \in \mathcal{W}(Q,D,T,m)} P(W, \phi_{\mathrm{DFC}}) = \max_{\omega \in \Omega(Q,D,T,m)} P(\omega, \phi_{\mathrm{DFC}}).$$

Any move wait walk $\omega \in \Omega(Q, D, T, m)$ we can write it in the form

$$\omega = ((*_{r_1}, \nu_1), ..., (*_{r_q}, \nu_q)),$$

for some choice of indices $r_l \in \{1, ..., n\}$ and $\nu_l \in \{\min T_{r_l}, ..., \max T_{r_l}\}$ for all $l \in \{1, ..., q\}$, for some $q \in \mathbb{N}$. We note that $t_q + \nu_q = T - 1$. The payoff for such a move wait walk against ϕ_{DFC} is

$$\begin{aligned} P(\omega, \phi_{\mathrm{DFC}}) &= \sum_{l=1}^{q} \sum_{t=n_l(\omega)}^{t_l(\omega)+\nu_l} \varphi_{*_{r_l},t} \\ &= \sum_{t=\min T_{r_1}}^{\min(\nu_1, \max T_{r_1})} \frac{1}{\hat{d}} + \sum_{l=2}^{q} \sum_{t=n'_l(\omega)}^{\min(t_l(\omega)+\nu_l, \max T_{r_l})} \frac{1}{\hat{d}} \\ &\leq \sum_{t=\min T_{r_1}}^{\min(\nu_1, \max T_{r_1})} \frac{1}{\hat{d}} + \sum_{l=2}^{q} \sum_{t=n''_l(\omega)}^{\min(t_l(\omega)+\nu_l, \max T_{r_l})} \frac{1}{\hat{d}} \end{aligned} \tag{6.12}$$

where $\hat{d} = m \sum_{j=1}^{i} b_j + 2 \sum_{j=i+1}^{q} b_j(k_j+1)$, $n_l(\omega) = \max(0, l_{*_{r_l}}(\omega)+1, t_l(\omega)-m+1)$, $n'_l(\omega) = \max(\min T_{r_l}, l_{*_{r_l}}(\omega)+1, t_l(\omega)-m+1)$ and $n''_l(\omega) = \max(\min T_{r_l}, t_l(\omega)-m+1)$. We can note that $t_l(\omega)+\nu_l > \max T_{r_l}$ for any $l \in \{2,...,q\}$ for any choices of indices and so we can write equation (6.12) as

$$P(\omega, \boldsymbol{\phi}_{\mathrm{DFC}}) \leq \sum_{t=\min T_{r_1}}^{\min(\nu_1, \max T_{r_1})} \frac{1}{\hat{d}} + \sum_{l=2}^{q} \sum_{t=n''_l(\omega)}^{\max T_{r_l}} \frac{1}{\hat{d}}. \tag{6.13}$$

Hence from equation (6.13) it should be clear that choosing $\nu_l = 0$ for all $l \in \{2,...,q\}$ maximizes the payoff.

We also have that if $t_{l'}(\omega)-m+1 > \min T_{r_{l'}}$ for some minimal $l' \in \{2,...,q\}$ then $t_l(\omega)-m+1 > \max T_{r_l}$ for all $l \in \{l'+1,...,q\}$. Moreover, for $l \in \{2,...,l'-1\}$ each r_l must be such that $m \geq 2(k_{r_l}+1)$ and so we can write equation (6.13) as

$$\begin{aligned}
P(\omega, \boldsymbol{\phi}_{\mathrm{DFC}}) \leq{}& \sum_{t=\min T_{r_1}}^{\min(\nu_1, \max T_{r_1})} \frac{1}{\hat{d}} + \sum_{l=2}^{q} \sum_{t=n''_l(\omega)}^{\max T_{r_l}} \frac{1}{\hat{d}} \\
={}& \frac{\min(\nu_1 - \min T_{r_1}+1, \max T_{r_1} - \min T_{r_1}+1)}{\hat{d}} \\
&+ \frac{\sum_{l=2}^{l'-1}(\max T_{r_l} - \min T_{r_l}+1)}{\hat{d}} + \frac{\max(\max T_{r_{l'}} - t_{l'}(\omega)+m, 0)}{\hat{d}} \\
={}& \frac{\min(\nu_1 - \min T_{r_1}+1, \max T_{r_1} - \min T_{r_1}+1)}{\hat{d}} \\
&+ \frac{\sum_{l=2}^{l'-1} 2(k_{r_l}+1)}{\hat{d}} + \frac{\max(\max T_{r_{l'}} - t_{l'}(\omega)+m, 0)}{\hat{d}}.
\end{aligned} \tag{6.14}$$

From equation (6.14) it is clear that that choosing $\nu_1 = \max T_{r_1}$ maximizes the payoff and in doing so equation (6.14) becomes

$$P(\omega, \boldsymbol{\phi}_{\mathrm{DFC}}) \leq \frac{(\max T_{r_1} - \min T_{r_1}+1) + (\max T_{r_{l'}} - \max T_{r_1} - k_{r_1} - k_{r_{l'}} - 2 + m)_+}{\hat{d}},$$

where $(y)_+ = \max(y, 0)$ is the rectifier function. By considering if r_1 and $r_{l'}$ are such that $m < 2(k_{r_1}+1)$ or $m \geq 2(k_{r_1}+1)$ we get

$$P(\omega, \boldsymbol{\phi}_{\mathrm{DFC}}) \leq \frac{m}{\hat{d}}$$

and hence

$$V_{\bullet, \boldsymbol{\phi}_{\mathrm{DFC}}}(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) \leq \frac{m}{m \sum_{j=1}^{i} b_j + 2 \sum_{j=i+1}^{q} b_j(k_j+1)}. \tag{6.15}$$

The upper bound on the performance of $\boldsymbol{\phi}_{\text{DFC}}$, as in equation (6.15), gives

$$V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) \leq \frac{m}{m \sum_{j=1}^{i} b_j + 2 \sum_{j=i+1}^{q} b_j(k_j+1)}.$$

□

The upper bound in lemma 6.1.14 along with the lower bound given in equation 6.11 give the the value of the game $(\widetilde{S}_n^{\boldsymbol{k}}, D, T, m)$ for $m \geq 2(k_{(q)}+1)$. Then the only region of attack length not yet solved is $1 \leq m \leq 2k_{(q)}+1$, in this region we will again use lemma 6.1.11 decomposing $G(\widetilde{S}_n^{\boldsymbol{k}}, D, T, m)$ into $n+1$ copies of $G((\{1\}, \emptyset), T, m)$. Therefore, for all $n \geq 3$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $1 \leq m \leq 2k_{(q)}+1$ and for all $T \geq m$ we have

$$V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) \geq \frac{1}{n+1}. \tag{6.16}$$

To get an equal upper bound consider that for $1 \leq m \leq 2k_{(q)}+1$ we have that $D((i,j)) \geq m$ for all $i, j \in \{c, *_1, ..., *_n\}$ and so we say they are *independent nodes* as for the same commencement time no pure patrol can catch attacks more than one attack. Forming an attacker strategy $\boldsymbol{\phi}$ such that $\varphi_{j,0} = \frac{1}{n+1}$ for all $j \in \{c, *_1, ..., *_n\}$ means that any pure patroller will only be able to see one of the potential pure attacks. Hence for all $n \geq 3$, for all $\boldsymbol{k} \in \mathbb{N}^n$, for all $1 \leq m \leq 2k_{(q)}+1$ and for all $T \geq m$ we have

$$V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) \leq \frac{1}{n+1}. \tag{6.17}$$

We conclude the results in this section with the following theorem.

Theorem 6.1.15. *For the game $G(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m)$, for all $n \in \mathbb{N}$, for all $\boldsymbol{k} \in \mathbb{N}^n$ and for all $T \geq 2m$ we have*

$$V(\widetilde{S}_n^{\boldsymbol{k}}, D_{n,\boldsymbol{k}}, T, m) = \begin{cases} \frac{m}{2(n+\sum_{j=1}^{n} k_j)} & \text{if } m \geq 2(k_{(1)}+1), \\ \frac{m}{m \sum_{j=1}^{i} b_j + 2 \sum_{j=i+1}^{q} b_j(k_j+1)} & \text{if } 2(k_{(i+1)}+1) \leq m \leq 2(k_{(i)}+1) \\ & \quad \text{for some } i \in \{1, ..., q-1\}, \\ \frac{1}{n+1} & \text{if } m < 2(k_{(q)}+1). \end{cases}$$

Theorem 6.1.15 follows in three parts from the results previously shown, in particular; equation (6.10) for $m \geq 2(k_{(1)}+1)$; equation (6.11) and lemma 6.1.14 for $2(k_{(i+1)}+1) \leq m \leq 2(k_{(i)}+1)$ for some $i \in \{1, ..., q-1\}$; and equations (6.16) and (6.17) for $m < 2(k_{(q)}+1)$. As we can see from theorem 6.1.15 it is much easier to get results for our scenario when we ignore the interior nodes along the branches. While the process of node sublimation removes choices for the attacker strategies it means that we are more easily able to get solutions. As previously mentioned, the use of node-sublimation can be considered in order to make any classic patrolling simpler to solve at the expense of removing locations the attacker can choose.

6.2 A patrolling game with node dependent attack lengths

6.2.1 Introduction to patrolling games with node dependent attack lengths

In this section we extend the classic patrolling game $G(Q,T,m)$ to the *patrolling game with node dependent attack lengths* $G(Q,T,\boldsymbol{m})$ in which the attack length for each node is given in the vector $\boldsymbol{m} \in \mathbb{N}^{|N|}$. By allowing the attack length to be dependent on the node the game $G(Q,T,\boldsymbol{m})$ can be used to model the vunerability of reinforcement of locations. This level of vulnerability can be represented in the model by a low attack length at the node, making it easier to attack, similarly the reinforcement can be represented in the model by a high attack length at the node, making it harder to attack. An example of this could be seen by considering a border protected by a barrier which the attacker must get through in order to cross. While the classic patrolling game would assume the barrier is uniformly hard to cross at all points along the border, now we can allow for different barriers to be placed at each location along the border, ranging from concrete walls to no barrier at all. While the main concern in the classic game is just the connectivity of locations, we must now additionally consider the vulnerability of those connected locations.

The *patrolling game with node dependent attack lengths* $G(Q,T,\boldsymbol{m})$ is parameterized by a 3-tuple $(Q,T,\boldsymbol{m})$ where $Q=(N,E)$ is a simple undirected graph with $N=\{1,...,n\}$ for some $n \in \mathbb{N}$, $T \in \mathbb{N}$ is the game length (with the time-horizon $\mathcal{J}=\{0,...,T-1\}$) and $\boldsymbol{m} \in \mathbb{N}^{|N|}$ is the *attack length vector.* The attack length for the node $j \in N$ is the j^{th} element of $\boldsymbol{m}$, namely m_j. The pure patroller strategies in the game $G(Q,T,\boldsymbol{m})$ remain the same as those in $G(Q,T,m)$ as there is no dependence on the attack length and thus the set of pure patrollers $\mathcal{W}(Q,T,\boldsymbol{m})=\mathcal{W}(Q,T,m)$. However, while the pure attacker strategies could remain the same, we limit the possible commencement time to ensure that it is possible to complete the attack before the end of the game and thus the set of pure attackers $\mathcal{A}(Q,T,\boldsymbol{m})=\{(j,\tau) \mid j \in N, \tau+m_j-1 \leq T-1\}$ (omitting $(Q,T,\boldsymbol{m})$ when the game is clear). As with the classic pure patrolling game the pure patrolling game with node dependent attack lengths is zero-sum and we define the payoff when the patroller chooses $W \in \mathcal{W}$ and the attacker chooses $(j,\tau) \in \mathcal{A}$, in terms of the patroller, as

$$P(W,(j,\tau))=\mathbb{I}_{\{j \in W(\{\tau,\dots,\tau+m_j-1\})\}}.$$

That is 1 if the patroller wins (attacker loses) by catching the attacker and 0 if the patroller loses (attacker wins) by failing to catch the attacker. We can assume some ordering of the sets $\mathcal{W}$ and $\mathcal{A}$ by two arbitrary bijections $\beta_1:\mathcal{W} \to \mathbb{N}$ and $\beta_2:\mathcal{A} \to \mathbb{N}$ respectively so $W_{(x)}=\beta_1^{-1}(x)$ and $a_{(y)}=\beta_2^{-1}(y)$. Then we can form a pure payoff matrix

$$\mathcal{P}=(P(W_{(x)},a_{(y)}))_{x \in \{1,\dots,|\mathcal{W}|\}, y \in \{1,\dots,|\mathcal{A}|\}}, \tag{6.18}$$

with a maximizing patroller and minimizing attacker. Like the classic pure patrolling game, the pure patrolling game with node dependent attack lengths has a Nash equilibrium if and only if there is a pure patrolling strategy which guarantees catching all pure attackers (i.e. there was a row of ones in the payoff matrix). Therefore, as in the classic patrolling game, this prompts us to allow for the mixing of strategies forming the mixed patrolling game with edges distances henceforth called the patrolling game with edge distances.

As the differences in attack lengths between nodes determines their relative vulnerability to each other we will define

$$M_{\min} = \min_{j \in \{1,\dots,|N|\}} m_j,$$

the minimal attack length, for which any node $j \in N$ such that $m_j = M_{\min}$ is a most vulnerable node. Similarly we define

$$M_{\max} = \max_{j \in \{1,\dots,|N|\}} m_j,$$

the maximal attack length, for which any node $j \in N$ such that $m_j = M_{\max}$ is a least vulnerable node. The difference between m_j and $M_{\min}$ for any node $j \in N$ shows the relative security to the most vulnerable node and similarly the difference between m_j and $M_{\max}$ for any node $j \in N$ shows the relative vulnerability to the most secure node.

In the mixed patrolling game with node dependent attack lengths each player chooses a probability distribution amongst all pure strategies. That is a mixed patroller strategy is $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{|\mathcal{W}|})$ where π_i is the probability of playing $W_{(i)}$ and a mixed attacker strategy is $\boldsymbol{\phi} = (\phi_1, \dots, \phi_{|\mathcal{A}|})$ where ϕ_i is the probability of playing $a_{(i)}$. As $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ are probability distributions we have

$$\boldsymbol{\pi} \in \Pi(Q, T, \boldsymbol{m}) = \left\{ \boldsymbol{x} \in [0,1]^{|\mathcal{W}(Q,T,\boldsymbol{m})|} \mid \sum_{i=1}^{|\mathcal{W}(Q,T,\boldsymbol{m})|} x_i = 1 \right\},$$

$$\boldsymbol{\phi} \in \Phi(Q, T, \boldsymbol{m}) = \left\{ \boldsymbol{y} \in [0,1]^{|\mathcal{A}(Q,T,\boldsymbol{m})|} \mid \sum_{i=1}^{|\mathcal{A}(Q,T,\boldsymbol{m})|} y_i = 1 \right\},$$

where $\Pi(Q,T,\boldsymbol{m})$ is the set of all mixed patroller strategies and $\Phi(Q,T,\boldsymbol{m})$ is the set of all mixed attacker strategies (omitting $(Q,T,\boldsymbol{m})$ when the game is clear). For the game $G(Q,D,T,m)$, the (patrollers) payoff for the patroller choosing $\boldsymbol{\pi} \in \Pi$ and the attacker choosing $\boldsymbol{\phi} \in \Phi$ is

$$P(\boldsymbol{\pi}, \boldsymbol{\phi}) = \sum_{i=1}^{|\mathcal{W}|} \sum_{j=1}^{|\mathcal{A}|} \mathcal{P}_{i,j} \pi_i \phi_j = \boldsymbol{\pi} \mathcal{P} \boldsymbol{\phi}^T, \tag{6.19}$$

with the objective of a maximizing patroller and minimizing attacker. In the same fashion as the classic game and its MiniMax and MaxiMin play variants, a player choosing a strategy will determine a performance according to their strategy leading and the optimal strategy chose by the following player. Thus the performance of a patroller choosing the mixed patroller strategy $\boldsymbol{\pi} \in \Pi$ is given by

$$V_{\boldsymbol{\pi},\bullet}(Q,T,\boldsymbol{m}) = \min_{\boldsymbol{\phi}\in\Pi} P(\boldsymbol{\pi},\boldsymbol{\phi}) = \min_{a\in\mathcal{A}} P(\boldsymbol{\pi},a),$$

and the performance of an attacker choosing the mixed attacker strategy $\boldsymbol{\phi} \in \Phi$ is given by

$$V_{\bullet,\boldsymbol{\phi}}(Q,T,\boldsymbol{m}) = \max_{\boldsymbol{\pi}\in\Pi} P(\boldsymbol{\pi},\boldsymbol{\phi}) = \max_{W\in\mathcal{W}} P(W,\boldsymbol{\phi}).$$

By theorem 2.2.2 we have the value of the game $G(Q,T,\boldsymbol{m})$ which is played simultaneously is given by

$$V(Q,T,\boldsymbol{m}) = \max_{\boldsymbol{\pi}\in\Pi}\min_{\boldsymbol{\phi}\in\Phi} P(\boldsymbol{\pi},\boldsymbol{\phi}) = \min_{\boldsymbol{\phi}\in\Phi}\max_{\boldsymbol{\pi}\in\Pi} P(\boldsymbol{\pi},\boldsymbol{\phi}).$$

Hence the value is bounded below by the performance of any mixed patroller strategy and above by any attacker strategy and therefore for all $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\phi} \in \Phi$ we have

$$0 \leq V_{\boldsymbol{\pi},\bullet}(Q,T,\boldsymbol{m}) \leq V(Q,T,\boldsymbol{m}) \leq V_{\bullet,\boldsymbol{\phi}}(Q,T,\boldsymbol{m}) \leq 1.$$

We define the performance of a mixed patroller strategy $\boldsymbol{\pi}$ at a given node j when the attack length at node j is m as

$$V_{\boldsymbol{\pi},\bullet,j,m}(Q,T) = \min_{\tau\in\{0,\dots,T-m+1\}} P(\boldsymbol{\pi},(j,\tau)).$$

Then the performance of the strategy is

$$V_{\boldsymbol{\pi},\bullet}(Q,T,\boldsymbol{m}) = \min_{j\in N} V_{\boldsymbol{\pi},\bullet,j,m_j}(Q,T).$$

Before presenting results on the game $G(Q,T,\boldsymbol{m})$ we note that as mixed strategies include pure strategies we will drop the term mixed.

6.2.2 Results for patrolling games with node dependent attack lengths

In this section we look at comparing various patrolling games with node dependent attack lengths when the attack length vector varies and is compared to the classic patrolling game. In particular we see that when the value of two patrolling games with node dependent attack lengths are equal by looking at the performance of optimal patroller strategies at each node in the graph. We then apply these results to get the value of such games on the star graph and line graph.

We begin by comparing the game $G(Q,T,\boldsymbol{m})$ to two classic games $G(Q,T,M_{\min})$ and $G(Q,T,M_{\max})$.

Lemma 6.2.1. *For the game $G(Q,T,\boldsymbol{m})$ for all graphs Q, for all $\boldsymbol{m} \in \mathbb{N}^n$ and all $T \geq M_{min}$ we have*

$$V(Q,T,M_{min}) \leq V(Q,T,\boldsymbol{m}) \leq V(Q,T,M_{max}).$$

The proof of lemma 6.2.1 follows as increasing the attack length at any node can only remove attacker strategies and does not affect the patroller strategies.

Proof. By the definition we have

$$\Phi(Q, T, M_{\max}) \subset \Phi(Q, T, \boldsymbol{m}) \subset \Phi(Q, T, M_{\min})$$

and

$$\mathcal{W}(Q, T, M_{\max}) = \mathcal{W}(Q, T, \boldsymbol{m}) = \mathcal{W}(Q, T, M_{\min}),$$

therefore

$$V(Q, T, M_{\min}) \leq V(Q, T, \boldsymbol{m}) \leq V(Q, T, M_{\max}).$$

□

The effectiveness of the bounds given by lemma 6.2.1 depends on the range of values in $\boldsymbol{m}$. When $\boldsymbol{m}$ is not extremely varied between nodes the bounds given can be relatively close. Although, we acknowledge that this closeness depends on how the classic patrolling game changes with the attack length. Likewise we can compare two patrolling games with node dependent attack lengths in which all pure attacker strategies for one game are available in the other game.

Lemma 6.2.2. *For the game $G(Q, T, \boldsymbol{m})$ for all graphs Q, for all $\boldsymbol{m} \in \mathbb{N}^{|N|}$ and for all $T \geq M_{min}$ we have*

$$V(Q, T, \widetilde{\boldsymbol{m}}) \leq V(Q, T, \boldsymbol{m}),$$

where $\widetilde{\boldsymbol{m}} \in \mathbb{N}^{|N|}$ is such that $\widetilde{m}_j \geq m_j$ for all $j \in N$.

Proof. By the definition we have

$$\Phi(Q, T, \widetilde{\boldsymbol{m}}) \subset \Phi(Q, T, \boldsymbol{m})$$

and

$$\mathcal{W}(Q, T, \widetilde{\boldsymbol{m}}) = \mathcal{W}(Q, T, \boldsymbol{m}),$$

therefore

$$V(Q, T, \widetilde{\boldsymbol{m}}) \leq V(Q, T, \boldsymbol{m}).$$

□

In order to construct some optimal strategies when the attack length is dependent on the node, we will study the effect on a currently known strategy when the attack length at a node is decreased. Sequentially decreasing attack lengths at given nodes allow us to introduce vulnerabilities into the game from a constant attack length game. For any patroller strategy $\boldsymbol{\pi}$ we know that

$$V(Q, T, \boldsymbol{m}) \geq \min_{j \in N} V_{\boldsymbol{\pi}, \bullet, j, m_j}(Q, T),$$

with equality if $\boldsymbol{\pi}$ is optimal. This bound allows us to consider changing m_j, the attack length of node $j \in N$, to see if there is any change to the lower bound given by $\boldsymbol{\pi}$ in the game $G(Q, T, \boldsymbol{m})$. If changing m_j doesn't result in a change the lower bound and $\boldsymbol{\pi}$ was optimal then it will remain optimal.

Lemma 6.2.3. *Consider a game $G(Q, T, \boldsymbol{m})$ with an optimal strategy $\boldsymbol{\pi}^*$. Then $\boldsymbol{\pi}^*$ is an optimal strategy for the game $G(Q, T, \widetilde{\boldsymbol{m}})$, where $\widetilde{m}_j \leq m_j$ for all $j \in N$ if*

$$\min_{j \in N} V_{\boldsymbol{\pi}^*, \bullet, j, \widetilde{m}_j}(Q, T) = \min_{j \in N} V_{\boldsymbol{\pi}^*, \bullet, j, m_j}(Q, T).$$

Consequently, $V(Q, T, \widetilde{\boldsymbol{m}}) = V(Q, T, \boldsymbol{m})$.

The proof of lemma 6.2.3 follows as $\boldsymbol{\pi}^*$ is feasible in both games.

Proof. If the patroller chooses $\boldsymbol{\pi}^*$ in the game $G(Q, T, \widetilde{\boldsymbol{m}})$ the patroller gets a bound of

$$V(Q, T, \widetilde{\boldsymbol{m}}) \geq \min_{j \in N} V_{\boldsymbol{\pi}^*, \bullet, j, \widetilde{m}_j}(Q, T) = \min_{j \in N} V_{\boldsymbol{\pi}^*, \bullet, j, m_j}(Q, T) = V(Q, T, \boldsymbol{m}).$$

Now as $\widetilde{m}_j \leq m_j$ for all $j \in N$, by lemma 6.2.2 we have the bound

$$V(Q, T, \widetilde{\boldsymbol{m}}) \leq V(Q, T, \boldsymbol{m}).$$

Hence $V(Q, T, \widetilde{\boldsymbol{m}}) = V(Q, T, \boldsymbol{m})$ and $\boldsymbol{\pi}^*$ remains optimal.

□

Lemma 6.2.3 allows us to take known optimal solutions for the classic game and verify that they remain optimal for certain choices of $\boldsymbol{m}$. We now present important graphical structures showcasing such results on the star graph S_n and line graph L_n for particular attack length vectors $\boldsymbol{m}$.

For the star graph S_n we can consider the game where the attack length at all leaf nodes are equal and the centre's attack length is less, but greater than 1. This restriction to the attack lengths at nodes allows us to use lemma 6.2.3.

Corollary 6.2.4. *For the star graph $S_n = (N, E)$ with $N = \{1, ..., n+1\}$ and $E = \{(1, i) \mid \leq i \leq n+1\}$, for all $n \geq 2$, and for all $m \geq 1$, for all $T \geq m$, and for all attack lengths at the centre m_1 such that $2 \leq m_1 \leq m$ we have*

$$V(S_n, T, \boldsymbol{m}) = \frac{m}{2n},$$

where $\boldsymbol{m} = (m_1, m, ..., m)$

Corollary 6.2.4 is proved by considering a random minimal full node cycle $\boldsymbol{\pi}^{S_n}_{\text{RMFNC}}$ which is optimal for the game $G(S_n, T, m)$.

Proof. For $\boldsymbol{\pi}^{S_n}_{\text{RMFNC}}$ we have a performance at node $j \in N$ of

$$V_{\boldsymbol{\pi}^{S_n}_{\text{RMFNC}}, \bullet, j, m_j}(S_n, T) = \begin{cases} \frac{m}{2n} & \text{if } 2 \leq j \leq n, \\ 1 & \text{if } j = 1. \end{cases}$$

As $\boldsymbol{\pi}^{S_n}_{\text{RMFNC}}$ is optimal in $G(S_n, T, (m, ..., m))$ and

$$\min_{j\in N} V_{\boldsymbol{\pi}^{S_n}_{\text{RMFNC}},\bullet,j,m_j}(Q,T) = \min_{j\in N} V_{\boldsymbol{\pi}^{S_n}_{\text{RMFNC}},\bullet,j,m}(Q,T)$$

we have, by lemma 6.2.3, that

$$V(S_n, T, (m_1, m, ..., m)) = V(S_n, T, m) = \frac{m}{2n}.$$

□

For the line graph L_n we have to place a more serve restriction on attack lengths at nodes, to ensure the random Hamiltonian still gives the same performance. That is we form the set of attack length vectors

$$\begin{aligned} M' = &\{\boldsymbol{m} \in \mathbb{N}^n \mid \min(m_1, m_n) \geq n-1, \\ &\min(m_1, m_n) \leq \min(2m_i, m_i, m_i + 2(n-i+1))\, \forall i \in \{2, ..., n-1\}\}, \end{aligned}$$

so that $\boldsymbol{\pi}_{\text{rH}}$ gives the worst performance at an end node 1 or n.

Corollary 6.2.5. *For the line graph, L_n, we have for all $n \geq 2$, for all $\boldsymbol{m} \in M'$ and for all $T \geq \min(m_1, m_n) + n - 2$ we have*

$$V(L_n, T, \boldsymbol{m}) = \frac{\min(m_1, m_n)}{2(n-1)}.$$

Corollary 6.2.5 is proved by considering a random minimal full-node cycle strategy $\boldsymbol{\pi}^{L_n}_{\text{RMFNC}}$.

Proof. For $\boldsymbol{\pi}^{L_n}_{\text{RMFNC}}$ we have a performance at node $j \in N$ of

$$V_{\boldsymbol{\pi}^{L_n}_{\text{RMFNC}},\bullet,j,m_j}(S_n,T) = \frac{\min(2m_j, m_j + 2(j-1), m_j + 2(n-j))}{2(n-1)}$$

As $\boldsymbol{\pi}^{L_n}_{\text{RMFNC}}$ is optimal in $G(L_n, T, M_{\min})$ and

$$\min_{j\in N} V_{\boldsymbol{\pi}^{L_n}_{\text{RMFNC}},\bullet,j,m_j}(Q,T) = \min_{j\in N} V_{\boldsymbol{\pi}^{L_n}_{\text{RMFNC}},\bullet,j,M_{\min}}(Q,T)$$

we have by lemma 6.2.3 that

$$V(L_n, T, \boldsymbol{m}) = \frac{\min(m_1, m_n)}{2(n-1)}.$$

□

That is the optimal strategies of S_n and L_n remain as the random minimal full-node cycle when some decreasing of the attack length at non-leaf nodes occurs. This is because the random minimal full-node cycle provides a better probability of interception for repeated nodes and leaf-node are not repeated, therefore allowing the lowering of the attack length at repeated nodes while retaining the same performance. While powerful, lemma 6.2.3 cannot be relied upon to solve all patrolling games with node dependent attack lengths.

6.2.3 Improvement of patrol strategies when attack lengths decrease

We now turn our attention to that situation, where lemma 6.2.3 is not applicable. That is the case that decreasing $\boldsymbol{m}$ to some $\widetilde{\boldsymbol{m}}$, with $\widetilde{m}_i \leq m_i$ for all $i = 1, ..., n$, gives us a strictly worse value for the game (and changes the optimal patrol strategy). To do so, we will now restrict ourselves to the case when

$$\min_{j \in N} V_{\boldsymbol{\pi}^*, \bullet, j, \widetilde{m}_j}(Q, T) < \min_{j \in N} V_{\boldsymbol{\pi}^*, \bullet, j, m_j}(Q, T). \tag{6.20}$$

An example of this can be seen when we introduce vulnerable nodes into a Hamiltonian graph. In the classic game the optimal strategy for Hamiltonian graphs is a random Hamiltonian which has the same performance at all nodes, ensuring that decreasing the attack length at any given node decreases the value of the lower bound provided on the value by the strategy. In section 6.2.4 we will find solutions to the Hamiltonian graph when some nodes have $m_i = 1$, called *instantaneous win* nodes.

By our assumption, the optimal patroller strategy $\boldsymbol{\pi}^*$ for $G(Q, T, \boldsymbol{m})$ is no longer optimal in $G(Q, T, \widetilde{\boldsymbol{m}})$. So we can look for improvements to $\boldsymbol{\pi}^*$ by using another other strategy. We will use an analogous idea to that developed in chapter 3, section 3.4, noting that the constant m is replaced with a node dependent m_j. As we have seen, choosing a set of strategies to improve a strategy can be tricky and does not always result in an optimal strategy. As mentioned previously this process often requires knowledge of the graphical structure, and furthermore the structure between nodes with low performances are connected under the current strategy. The following are the current tools for improvement.

Suppose that $\boldsymbol{\pi}_h$ is our hybrid strategy made up of some baseline strategy $\boldsymbol{\pi}_0$, along with some $\boldsymbol{\pi}_i$ which are played with probability p_i for $i = 1, ..., l$. Then this hybrid strategy has a performance of

$$V_{\boldsymbol{\pi}_h, \bullet, j, m_j}(Q, T) = \sum_{i=0}^{l} V_{\boldsymbol{\pi}_i, \bullet, j, m_j}(Q, T) p_i,$$

where $p_0 = 1 - \sum_{j=1}^{l} p_j$ and $0 \leq p_j \leq 1$ for $j = 0, ..., l$. In order to say the hybrid strategy is a strict improvement over the baseline strategy we need

$$\min_{j \in N} V_{\boldsymbol{\pi}_h, \bullet, j, m_j}(Q, T) > \min_{j \in N} V_{\boldsymbol{\pi}_0, \bullet, j, m_j}(Q, T).$$

Like the PIP (equation (3.19)) we get the following equation in order to determine the best choice of p_i for $i = 1, ..., l$.

$$\begin{aligned} \text{maximize} \quad & \min_{j \in N} \sum_{i=0}^{l} V_{\boldsymbol{\pi}_i, \bullet, j, m_j}(Q, T) p_i \\ \text{subject to} \quad & \sum_{i=0}^{l} p_i = 1, \\ & p_i \in [0, 1], \qquad\qquad \text{for } i = 0, ..., l. \end{aligned} \tag{6.21}$$

Optimal solutions to the program in equation (6.21) give the best choice of p_i for $i = 1, ..., l$ and the optimal value gives the lower bound provided by the hybrid strategy using the optimal p_i for $i = 1, ..., l$. The program in equation (6.21) can be written as a linear program by implementing the objective function as a constraint with an extra variable for computational implementation. As we are looking at improving the previously optimal strategy $\boldsymbol{\pi}$ (optimal for the game $G(Q, T, \boldsymbol{m})$), the question to now consider is what strategies should be picked to possibly improve it. We can use a host of strategies but the careful selection of them is important to find a good improvement with little analytical computation. To this end we will discuss what sort of strategies should be used.

Let us first define the nodes which have a decreased attack length from $\boldsymbol{m}$ to $\widetilde{\boldsymbol{m}}$ as $N_{m\downarrow} = \{j \in N \mid \widetilde{m}_j < m_j\}$. Next we can define a subset of these nodes which effect the value of the game

$$N_{V\downarrow} = \{j \in N_{m\downarrow} \mid V_{\boldsymbol{\pi}^*,\bullet,j,\widetilde{m}_j}(Q, T) < V(Q, T, \boldsymbol{m}) = V_{\boldsymbol{\pi}^*,\bullet}(Q, T, \boldsymbol{m})\}.$$

Clearly it is the problem of changing m_j to $\widetilde{m}_j$ for $j \in N_{V\downarrow}$ that decreased the value of the game and hence the performance at such nodes must be improved. Exactly how to choose the improvement strategies, $\boldsymbol{\pi}_i$ for $i = 1, ..., l$ for some $l \in \mathbb{N}$ is still a problem, but we know they must improve each node $j \in N_{V\downarrow}$.

A simple, but effective, choice is the use of intercepting patrols to improve these nodes. Each intercepting patrol containing the node j must return to j within $\widetilde{m}_j$ time units, for each $j \in N_{V\downarrow}$. Therefore to find the minimal number of intercepting patrols needed to cover such nodes $j \in N_{V\downarrow}$ we must determine if an intercepting strategy exists for each possible subset of decomposition of $N_{V\downarrow}$. To illustrate such a selection of intercepting patrols we provide example 6.2.6. We will now assume a set of intercepting patrols $\mathcal{W}_{\text{Int},N_{V\downarrow}}$ is a minimal such set. We can choose to use $l = |\mathcal{W}_{\text{Int},N_{V\downarrow}}|$ improvement strategies each playing a distinct intercepting patrol W_i from $\mathcal{W}_{\text{Int},N_{V\downarrow}}$. Let $\boldsymbol{\pi}_i = W_i$ for $i = 1, ..., l$, so they are in mixed strategy notation. The set $N_{m\downarrow} \cap W_i(\mathcal{J})$ are the nodes in $N_{m\downarrow}$ which are also in the i^{th} intercepting patrol strategy W_i, giving us the important nodes to consider. Note that a node may be in more than one intercepting patrol. We can then use the program in equation (6.21) to get the lower bound which follows from using these improvement strategies.

Example 6.2.6. Consider the graph Q as shown in figure 6.2.1, an attack length vector $\widetilde{\boldsymbol{m}} = (4, 2, 4, 3, 1, 4, 2)$, as shown in red on the nodes of the graph Q to form the patrolling game with node dependent attack lengths $G(Q, T, \widetilde{\boldsymbol{m}})$ for some $T \geq 4$.

Say $\boldsymbol{\pi}^*$ is an optimal solution to the game $G(Q, T, 4)$ then as $G(Q, T, 4) \equiv G(Q, T, \boldsymbol{m})$ where $\boldsymbol{m} = (4, 4, 4, 4, 4, 4, 4)$ it is optimal for $G(Q, T, \boldsymbol{m})$ with a value of $V(Q, T, \boldsymbol{m}) = \frac{4}{7}$. For the game $G(Q, T, \widetilde{\boldsymbol{m}})$ the performance of node j using $\boldsymbol{\pi}^*$ is

$$V_{\boldsymbol{\pi}^*,\bullet,j,\widetilde{m}_j}(Q, T) = \frac{\widetilde{m}_j}{7}.$$

As nodes 2, 4, 5 and 7 are made more vulnerable by changing $\boldsymbol{m} = (4,4,4,4,4,4,4)$ to $\widetilde{\boldsymbol{m}} = (4,2,4,3,1,4,2)$ these are the nodes requiring the improvement, i.e $N_{V\downarrow} = \{2,4,5,7\}$.

As $\widetilde{m}_5 = 1$ it is clear that one intercepting patroller strategy W_1 must wait at node 5 for the entire time-horizon. Nodes 2 and 4 can be covered by a single intercepting patrolling strategy W_2 which alternates between nodes 2 and 4 for the time-horizon. Finally this leaves only node 7 for which a single intercepting patroller strategy W_3 can be used, as $\widetilde{m}_7 = 2$ we may as well have W_3 alternate between node 6 and 7 for the time-horizon. Therefore we use W_1, W_2, W_3 as improvement strategies $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3$ which are intercepting and thus

- $V_{\boldsymbol{\pi}_1,\bullet,j,\widetilde{m}_j}(Q,T) = \mathbb{I}_{\{j=5\}}$,
- $V_{\boldsymbol{\pi}_2,\bullet,j,\widetilde{m}_j}(Q,T) = \mathbb{I}_{\{j\in\{2,4\}\}}$ and
- $V_{\boldsymbol{\pi}_3,\bullet,j,\widetilde{m}_j}(Q,T) = \mathbb{I}_{\{j\in\{6,7\}\}}$

So using $\boldsymbol{\pi}_0 = \boldsymbol{\pi}^*$ and $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3$ in equation (6.21) along with $p_0 = 1-p_1-p_2-p_3$ we get the following program.

$$\begin{aligned}\max((1-p_1-p_2-p_3)\frac{4}{7}, (1-p_1-p_2-p_3)\frac{1}{7} + p_1,\\ (1-p_1-p_2-p_3)\frac{2}{7} + p_2, (1-p_1-p_2-p_3)\frac{2}{7} + p_3)\\ \text{s.t } p_1,p_2,p_3 \geq 0 \text{ and } p_1+p_2+p_3 \leq 1.\end{aligned}$$

From this program we can immediately note that $p_2 = p_3$, as both W_2 and W_3 are essentially as effective at improving the performance, as they both contain a node with an attack length of 2. Solving gives $p_1 = \frac{3}{14}$ and $p_2 = p_3 = \frac{1}{7}$ giving a bound of

$$V(Q,T,\widetilde{\boldsymbol{m}}) \geq \frac{18}{49} > \frac{1}{7} = V_{\boldsymbol{\pi}^*,\bullet}(Q,T,\widetilde{\boldsymbol{m}}).$$

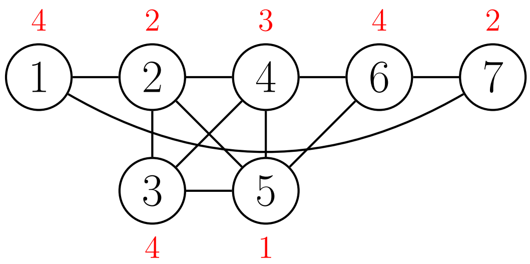


Figure 6.2.1: The graph Q used in example 6.2.6 with attack lengths for nodes from $\widetilde{\boldsymbol{m}}$ shown in red alongside the node.

6.2.4 Hamiltonian graphs with instantaneous win nodes

Having seen the complexity of searching for an improvement on the previously optimal strategy we now look at patrolling games with node dependent attack lengths $G(Q,T,\boldsymbol{m})$ where $Q \in \mathcal{H}$ is a Hamiltonian graph where some nodes are extremely vulnerable and can not be left alone for even one unit of time. We use the improvement idea along with the program in equation (6.21) in order to get a lower bound on the game. Further, we will solve to optimality the game on the Hamiltonian graph when *instantaneous win nodes* are introduced.

Definition 6.2.7. For the game $G(Q,T,\boldsymbol{m})$ we call a node $j \in N$ an *instantaneous win* node if $m_j = 1$. We define the set of instantaneous win nodes as

$$I(Q,T,\boldsymbol{m}) = \{j \in N \mid m_j = 1\}.$$

If a node $j \in N$ is an instantaneous win node then if the patroller is not at the node when the attacker commences their attack then they fail to catch the attacker and hence the patroller loses. Therefore a game in which a node is instantaneous win means the node is extremely vulnerable and can not be left alone for even a moment without the risk of losing. So the game $G(Q,T,\boldsymbol{m})$ where $Q \in \mathcal{H}$ can be seen as the model of a circular perimeter with some unprotected locations, such as those without protective barriers, with some additional connections between nodes. In general these additional connections can affect how the patroller should patrol between vulnerable nodes and while this is extremely complicated and depends on which additional edges are present we can get a result if we assume that all other nodes aside from instantaneous win nodes have the same attack length.

For the game $G(Q,T,\boldsymbol{m})$ where $Q \in \mathcal{H}$ and $\boldsymbol{m}$ is such that $m_j = m$ if $j \notin I(Q,T,\boldsymbol{m})$ and $m_j = 1$ if $j \in I(Q,T,\boldsymbol{m})$ we look at improving the random Hamiltonian patroller strategy using equation (6.21) and improvement strategies which wait at all instantaneous win nodes. Let $\boldsymbol{\pi}_0 = \boldsymbol{\pi}_{\text{rH}}$ and then let $\boldsymbol{\pi}_i$ for $i = 1, ..., |I(Q,T,\boldsymbol{m})|$ each play a distinct pure strategy $W(t) = j$ for all $t \in \mathcal{J}$ for each $j \in I(Q,T,\boldsymbol{m})$. Then by equation (6.21) we have $p_1 = p_2 = ... = p_{|I(Q,T,\boldsymbol{m})|}$ and letting $p \equiv p_1$ and $p_0 = 1 - |I(Q,T,\boldsymbol{m})|p$ we have

$$\begin{aligned} \text{maximize} \quad & \min(1 - |I(Q,T,\boldsymbol{m})|p)\frac{m}{n}, (1 - |I(Q,T,\boldsymbol{m})|p)\frac{1}{n} + p) \\ \text{subject to} \quad & 0 \leq p \leq \tfrac{1}{|I(Q,T,\boldsymbol{m})|} \end{aligned} \tag{6.22}$$

Solving equation (6.22) gives us $p = \frac{m-1}{n+|I(Q,T,\boldsymbol{m})|(m-1)}$ and thus we arrive at the strategy $\boldsymbol{\pi}_{\text{irH}}$ which plays $\boldsymbol{\pi}_{rH}$ with probability $\frac{n}{n+|I(Q,T,\boldsymbol{m})|(m-1)}$ and each $\boldsymbol{\pi}_i$ with probability $\frac{m-1}{n+|I(Q,T,\boldsymbol{m})|(m-1)}$ for $i = 1, ..., |I(Q,T,\boldsymbol{m})|$.

Definition 6.2.8. For the game $G(Q,T,\boldsymbol{m})$ where $Q \in \mathcal{H}$ and $\boldsymbol{m}$ is such that $m_j = m$ if $j \notin I(Q,T,\boldsymbol{m})$ and $m_j = 1$ if $j \in I(Q,T,\boldsymbol{m})$, the improved random Hamiltonian patroller strategy $\boldsymbol{\pi}_{\text{irH}}$, using a Hamiltonian cycle H, is such that

$$\pi_{\beta_1(W)} = \begin{cases} \frac{1}{n+|I(Q,T,\boldsymbol{m})|(m-1)} & \text{if } W \in \{W_0, ..., W_{n-1}\}, \\ \frac{m-1}{n+|I(Q,T,\boldsymbol{m})|(m-1)} & \text{if } W(t) = j \, \forall t \in \mathcal{J}, \\ 0 & \text{otherwise,} \end{cases}$$

where $W_i(t) = H(t+i \mod n)$ for all $t \in \mathcal{J}$ for all $i \in \{0, ..., n-1\}$.

Lemma 6.2.9. *For the game $G(Q, T, \boldsymbol{m})$ where $Q \in \mathcal{H}$ and $\boldsymbol{m}$ is such that $m_j = m$ if $j \notin I(Q, T, \boldsymbol{m})$ and $m_j = 1$ if $j \in I(Q, T, \boldsymbol{m})$, for any $1 \leq m \leq n$ and any $T \geq m$ we have*

$$V(Q, T, \boldsymbol{m}) \geq \frac{m}{n + |I(Q, T, \boldsymbol{m})|(m-1)},$$

achieved by the patroller choosing the improved random Hamiltonian $\boldsymbol{\pi}_{irH}$, using any Hamiltonian cycle H.

The proof of lemma 6.2.9 follows immediately from the optimal value of the program in equation (6.22) by substituting the optimal p. Equivalently by patroller choosing $\boldsymbol{\pi}_{\text{irH}}$.

For classic games on Hamiltonian graphs the position-uniform attacker strategy which chooses $(j, 0)$ with equal probability for all $j \in N$ is optimal. However this can be augmented to give a better upper bound for the attacker in the game $G(Q, T, \boldsymbol{m})$ where $Q \in \mathcal{H}$ and $\boldsymbol{m}$ is such that $m_j = m$ if $j \notin I(Q, T, \boldsymbol{m})$ and $m_j = 1$ if $j \in I(Q, T, \boldsymbol{m})$. To augment the strategy we have m pure attacks happen at each instantaneous win node commencing at all times in the attack interval for a non-instantaneous win node.

Definition 6.2.10. For the game $G(Q, T, \boldsymbol{m})$ where $Q \in \mathcal{H}$ and $\boldsymbol{m}$ is such that $m_j = m$ if $j \notin I(Q, T, \boldsymbol{m})$ and $m_j = 1$ if $j \in I(Q, T, \boldsymbol{m})$, let the *augmented position-uniform* attacker strategy $\boldsymbol{\phi}_{\text{apu}}$ be such that the probability of playing (j, τ) is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{n+|I(Q,T,\boldsymbol{m})|(m-1)} & \text{for } j \notin N \setminus I(Q, T, \boldsymbol{m}), \tau = 0, \\ \frac{1}{n+|I(Q,T,\boldsymbol{m})|(m-1)} & \text{for } j \in I(Q, T, \boldsymbol{m}), \tau \in \{0, ..., m-1\}. \end{cases}$$

Lemma 6.2.11. *For the game $G(Q, T, \boldsymbol{m})$ where $Q \in \mathcal{H}$ and $\boldsymbol{m}$ is such that $m_j = m$ if $j \notin I(Q, T, \boldsymbol{m})$ and $m_j = 1$ if $j \in I(Q, T, \boldsymbol{m})$, for any $1 \leq m \leq n$ and any $T \geq m$ we have*

$$V(Q, T, \boldsymbol{m}) \leq \frac{m}{n + |I(Q, T, \boldsymbol{m})|(m-1)},$$

by the attacker choosing $\boldsymbol{\phi}_{apu}$

The proof of lemma 6.2.11 follows by looking at the performance of $\boldsymbol{\phi}_{\text{apu}}$.

Proof. We first note that all pure attacks occur with the same probability in $\boldsymbol{\phi}_{\text{apu}}$ and so we only need to consider how many pure attacks it is possible for any walk to catch.

Consider any $W \in \mathcal{W}$ then for each $t \in \{0, ..., m-1\}$ we have either $W(t) = j \notin I(Q, T, \boldsymbol{m})$ and then $(j, 0)$ is caught if not previously caught, or $W(t) =$

$j \in I(Q, T, \boldsymbol{m})$ and then (j, t) is caught. Therefore for any $W \in \mathcal{W}$ we have at most one potential attack caught for each time $t \in \{0, ..., m-1\}$. Moreover $W(t) = j'$ for all $t \in \mathcal{J}$ for some $j \in I(Q, T, \boldsymbol{m})$ catches exactly m potential attacks. Therefore

$$V(Q, T, \boldsymbol{m}) \leq V_{\bullet, \phi_{\text{apu}}}(Q, T, \boldsymbol{m}) = \frac{m}{n + |I(Q, T, \boldsymbol{m})|(m-1)}.$$

□

Theorem 6.2.12. *For the game $G(Q, T, \boldsymbol{m})$ for any $Q \in \mathcal{H}$, and $\boldsymbol{m}$ such that $m_j = 1$ for $j \in I$ and $m_j = m$ for $j \in N \setminus I$ for any $m \geq 1$ and any $T \geq m$ we have*

$$V(Q, T, \boldsymbol{m}) = \frac{m}{n + |I|(m-1)}.$$

Theorem 6.2.12 follows from equal lower and upper bounds given in lemmas 6.2.9 and 6.2.11 respectively.

To conclude we note that these strategies of improvement and attacker adaptation are not as effective when the vulnerabilities below $M_{\max}$ are not instantaneous win nodes. This is because, as is usual for patrolling games, finding an attacker strategy is difficult and using the same idea of making more pure attacks distributed in commencement time, as used to from the attacker strategy $\boldsymbol{\phi}_{\text{apu}}$, does not provide a tight bound. We note that theorem 6.2.12 means such games on the graphs C_n and K_n have the same value and the additional edges between them do not affect the performance.

6.3 Multiple players

We now look at introducing more players into the classic patrolling game. When looking at a game with $k \in \mathbb{N}$ patrollers and $l \in \mathbb{N}$ attackers we must decide how we model the interaction between players. Can players collaborate with each other or are they selfish individuals who act individually? For the purpose of this chapter we will assume that the k patroller players form a collaboration and are controlled by a *scheduler* who decides how they patrol the graph. This makes the patrollers act as one entity, picking a pure schedule of walks $\boldsymbol{W} = (W_1, ..., W_k) \in \mathcal{W}(Q, T, m)^k$, where $W_f \in \mathcal{W}(Q, T, m)$ is the f^{th} patrollers pure walk for the schedule $\boldsymbol{W}$ (We often omit (Q, T, m) if these parameters are clear). This models the scenario of a police dispatcher scheduling the patrols of multiple police units. With the ordering of the set $\mathcal{W}(Q, T, m)$ done by some arbitrary bijection $\beta_1 : \mathcal{W} \to \{1, ..., |\mathcal{W}|\}$ so $W_{(x)} = \beta_1^{-1}(x)$ is the x^{th} ordered walk. With this bijection we describe any pure schedule $\boldsymbol{W}$ as the choice of ordered sets $\beta_1(\boldsymbol{W}) = (\beta_1(W_1), ..., \beta_1(W_k)) \in \{1, ..., |\mathcal{W}|\}^k$, meaning the f^{th} pure patroller uses the $\beta_1(W_f)^{\text{th}}$ walk. The scheduler's mixed strategy is a distribution over all such pure schedules, that is some $\boldsymbol{s} \in \zeta$, where ζ is the set of all such mixed scheduler strategies. We denote the probability of playing the pure schedule $(W_1, ..., W_k)$ by $s_{\beta_1(W_1),...,\beta_1(W_k)}$, that is $s_{i_1,....,i_k}$ denotes the probability of playing $(W_{(i_1)}, ..., W_{(i_k)})$. Given a scheduler's strategy of $\boldsymbol{s} \in \zeta$ we can look at each patrollers individual strategy

Definition 6.3.1. For a given $\boldsymbol{s} \in \zeta$ we define the f^{th} patrollers *individualized* strategy $\boldsymbol{\pi}^f(\boldsymbol{s}) \in \Pi(Q, T, m)$ such that the probability of choosing the i^{th} ordered walk $W_{(i)} \in \mathcal{W}(Q, T, m)$ is given by $\pi^f_{\beta_1(W_{(i)})}(\boldsymbol{s}) = s^i_f$ for all $i \in \{1, ..., |\mathcal{W}|\}$ and for all $f \in \{1, ..., k\}$, where

$$s^i_f = \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_{f-1}=1}^{|\mathcal{W}|} \sum_{i_{f+1}=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} s_{i_1,...,i_{f-1},i,i_{f+1},...,i_k}$$

While this modelling seems the most obvious answer for the k patrollers, there are a multitude of scenarios for the l attackers we may wish to consider. We present work on four options for the l attackers:

- Selfish attackers in the game $G_1(k, l, Q, T, m)$.
- Collaborative attackers, who need all attackers to succeed in the game $G_2(k, l, Q, T, m)$.
- Collaborative attackers, who want as many attackers as possible to succeed in the game $G_3(k, l, Q, T, m)$.
- Collaborative attackers, who need one attacker to succeed in the game $G_4(k, l, Q, T, m)$.

While the 5-tuple (k, l, Q, T, m) remains the same for each of the four games, with $k \in \mathbb{N}$ being the number of patrollers, $l \in \mathbb{N}$ being the number of attackers and (Q, T, m) as in the classic game, each game has different attacker strategies and (attacker) payoffs denoted by the games subscript.

The following four subsections contain our work on the four options for the l attackers respectively. In each we define the strategies and payoff for the attackers (and patroller, by noting the game is zero-sum) and find optimal solutions often relying on the classic patrolling game albeit with some dispersal or grouping of the k patrollers and l attackers.

6.3.1 Selfish attackers

In the patrolling game $G_1(k, l, Q, T, m)$ we have k patrollers controlled by a single scheduler playing against l selfish attackers who are individually controlled. In the game $G_1(k, l, Q, T, m)$ the r^{th} attacker picks a pure strategy $(j, \tau) \in \mathcal{A}(Q, T, m)$ (as usual we may omit (Q, T, m) when the parameters are clear) or a mixed strategy $\boldsymbol{\phi} \in \Phi$, where as in the classic game $\mathcal{A}$ is the set of all pure attacks and

Φ is the set of all mixed attacks (distributions over all pure attacks) for $r = 1, ..., l$. We denote the probability that the r^{th} attacker plays the pure strategy (j, τ) by $\varphi_{r,j,\tau}$. As each attacker is selfish their strategy is chosen regardless of others, hence $G_1(k, l, Q, T, m)$ is a $l + 1$ player game in which one player is the scheduler and l players are attackers. When the r^{th} attacker chooses (j_r, τ_r) for $r = 1, ..., l$ and the scheduler plays $\boldsymbol{W}$ we define the pure payoff of the game for the R^{th} *attacker* as

$$P_{a,R}(\boldsymbol{W}, (j_1, \tau_1), ..., (j_l, \tau_l)) = \prod_{f=1}^{k} \mathbb{I}_{\left\{j_R \notin W_f(\{\tau_R, ..., \tau_R+m-1\})\right\}},$$

for each $R \in \{1, ..., l\}$. That is the R^{th} attacker, playing (j_R, τ_R), wins (gets a pure payoff of 1) if and only if they are not caught by any of the k patroller walks used by the scheduler in $\boldsymbol{W}$. Note that the R^{th} attacker's payoff is independent of all other attackers strategies and so we may write $P_{a,R}(\boldsymbol{W}, (j, \tau))$ for the payoff of the R^{th} attacker choosing (j, τ) against $\boldsymbol{W}$. To keep the zero-sum nature of the game we define the scheduler's payoff as

$$P_s(\boldsymbol{W}, (j_1, \tau_1), ..., (j_l, \tau_l)) = l - \sum_{r=1}^{l} P_{a,r}(\boldsymbol{W}, (j_1, \tau_1), ..., (j_l, \tau_l)).$$

That is the scheduler, playing $\boldsymbol{W}$, gets a pure payoff equal to the number of attackers that are caught when the r^{th} attacker plays (j_r, τ_r) for $r = 1, ..., l$.

For the mixed strategies, $\boldsymbol{s} \in \zeta$ for the scheduler and $\boldsymbol{\phi}_r \in \Phi$ for the r^{th} attacker for $r = 1, ..., l$ the payoff for the R^{th} attacker in the game $G_1(k, l, Q, T, m)$ is given by

$$\begin{aligned} &P_{a,R}(\boldsymbol{s}, \boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_l) \\ &= \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} \sum_{j \in N} \sum_{t=0}^{T-m} s_{i_1,....,i_k} \varphi_{R,j,t} P_{a,R}((W_{(i_1)}, ..., W_{(i_k)}), (j, t)). \end{aligned} \tag{6.23}$$

That is the payoff is the probability that the R^{th} attacker is not caught by the scheduler's patrollers. Likewise the payoff for the scheduler is given by

$$\begin{aligned} &P_s(\boldsymbol{s}, \boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_l) \\ &= \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} \sum_{r=1}^{l} \sum_{j \in N} \sum_{t=0}^{T-m} s_{i_1,....,i_k} \varphi_{r,j,t} (1 - P_{a,r}((W_{(i_1)}, ..., W_{(i_k)}), (j, t))). \end{aligned} \tag{6.24}$$

That is the payoff is the expected number of attackers caught by the scheduler. We now define the value of the game as

$$\begin{aligned} V_1(k, l, Q, T, m) &= \max_{\boldsymbol{s} \in \zeta} \min_{\boldsymbol{\phi}_1 \in \Phi} \ldots \min_{\boldsymbol{\phi}_l \in \Phi} P_s(\boldsymbol{s}, (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_l)) \\ &= \min_{\boldsymbol{\phi}_1 \in \Phi} \ldots \min_{\boldsymbol{\phi}_l \in \Phi} \max_{\boldsymbol{s} \in \zeta} P_s(\boldsymbol{s}, (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_l)) \end{aligned} \tag{6.25}$$

We immediately note that from the definition of $G_1(k, l, Q, T, m)$ that when $k = 1$ and $l = 1$ the game is equivalent to the classic patrolling game. I.e. $G_1(1, 1, Q, T, m) \equiv G(Q, T, m)$ for any set of classic game parameters (Q, T, m).

Lemma 6.3.2. *For the game $G_1(k, l, Q, T, m)$ for any $k \in \mathbb{N}$, for any $l \in \mathbb{N}$, for any graph Q, for any $m \geq 1$, for any $T \geq m$ we have*

$$V_1(k, l, Q, T, m) = lV_1(k, 1, Q, T, m).$$

In particular

$$V_1(1, l, Q, T, m) = lV(Q, T, m).$$

The proof of lemma 6.3.2 follows from the fact that the each minimizing attacker does not influence the other attackers.

Proof. We start by writing the payoff (for the scheduler) as a sum for each of the l attackers, who only control one of the l parts of the sum.

$$P_s(\boldsymbol{s}, (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_l)) = \sum_{r=1}^{l} \sum_{j \in N} \sum_{t=0}^{T-m} \varphi_{r,j,t} \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} s_{i_1,...,i_k} (1 - P_{a,r}((W_{(i_1)}, ..., W_{(i_k)}), (j, t)))$$

This allows us to write the value of the game as

$$\begin{aligned} V_1(k, l, Q, T, m) = &\sum_{r=1}^{l} \min_{\boldsymbol{\phi}_r \in \Phi} \max_{\boldsymbol{s} \in \zeta} \sum_{j \in N} \sum_{t=0}^{T-m} \varphi_{r,j,t} \\ &\times \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} s_{i_1,...,i_k} (1 - P_{a,r}((W_{(i_1)}, ..., W_{(i_k)}), (j, t))), \end{aligned}$$

where choosing $\varphi_{r,j,t}$ for each $j \in N$ and each $t \in \mathcal{J}$ is equivalent to choosing $\boldsymbol{\phi}_r \in \Phi$ for each minimization.

By the definition of the payoff (equation (6.24)) and value of the game (equation (6.25))we have

$$V_1(k, l, Q, T, m) = \sum_{r=1}^{l} V_1(k, 1, Q, T, m) = lV_1(k, 1, Q, T, m).$$

For the second assertion of the lemma we note that $V_1(1, 1, Q, T, m) = V(Q, T, m)$ by the equivalent definitions of strategies and value. □

Lemma 6.3.2 allows us to now focus our efforts on investigating how a scheduler can utilise k patrollers in an effort to catch one attacker. That is we now study the game $G_1(k, 1, Q, T, m)$ for the rest of this subsection, focusing on how an optimal scheduler should act. For the game $G_1(k, 1, Q, T, m)$ we can simplify the mixed payoff for the scheduler using $\boldsymbol{s} \in \zeta$ and the attacker using $\boldsymbol{\phi} \in \Phi$ in equation (6.24) to

$$P(\boldsymbol{s}, \boldsymbol{\phi}) = \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} \sum_{j \in N} \sum_{\tau=0}^{T-m} s_{i_1,...,i_k} \varphi_{j,\tau} \mathbb{I}_{\left\{ j \in \bigcup_{f=1}^{k} W_{(i_f)}(\{\tau, ..., \tau+m-1\}) \right\}}.$$

However this representation of the payoff does not explicitly tell us the effectiveness of each patroller the scheduler is able to utilise. To see this better we can write the contribution to the probability that the f^{th} patroller has in catching the attacker using $\boldsymbol{\phi} \in \Phi$ when the scheduler is using $\boldsymbol{s} \in \zeta$ as

$$P(\boldsymbol{s}, \boldsymbol{\phi}, f)$$
$$= \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} \sum_{j \in N} \sum_{\tau=0}^{T-m} s_{i_1,\ldots,i_k} \varphi_{j,\tau} \mathbb{I}_{\{j \in W_{(i_f)}(\{\tau,\ldots,\tau+m-1\})\}} \mathbb{I}_{\left\{j \notin \bigcup_{r=1}^{f-1} W_{(i_r)}(\{\tau,\ldots,\tau+m-1\})\right\}}.$$

That is the f^{th} patroller only contributes if it is the lowest indexed patroller who catches the attacker. We note that $P(\boldsymbol{s}, \boldsymbol{\phi}) = \sum_{f=1}^{k} P(\boldsymbol{s}, \boldsymbol{\phi}, f)$. With this idea of how much each successive patroller contributes we can easily see that the best performance utilises patrollers who only catch distinct pure attacks. From this idea we get the following lemma. For clarity we note the contribution against a pure attack strategy $(j, \tau) \in \mathcal{A}$ is

$$P(\boldsymbol{s}, (j, \tau), f) = \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} s_{i_1,\ldots,i_k} \mathbb{I}_{\left\{j \in W_{(i_f)}(\{\tau,\ldots,\tau+m-1\})\right\}} \mathbb{I}_{\left\{j \notin \bigcup_{r=1}^{f-1} W_{(i_r)}(\{\tau,\ldots,\tau+m-1\})\right\}}.$$

Then we have $P(\boldsymbol{s}, \boldsymbol{\phi}, f) = \sum_{j \in N} \sum_{\tau=0}^{T-m} \varphi_{j,\tau} P(\boldsymbol{s}, (j, \tau), f)$.

Lemma 6.3.3. *For the game $G_1(k, 1, Q, T, m)$ for all $k \in \mathbb{N}$, for all graphs Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V_1(k, 1, Q, T, m) \leq \min(kV(Q, T, m), 1).$$

Furthermore we have equality, $V_1(k, 1, Q, T, m) = kV(Q, T, m)$, if there exists $\boldsymbol{s} \in \zeta$ such that

- *for all $\boldsymbol{W} \in \mathcal{W}^k$ such that $s_{\beta_1(\boldsymbol{W})} > 0$ then for all $(j, \tau) \in \mathcal{A}$ if $j \in W_i(\{\tau, \ldots, \tau+m-1\})$ then $j \notin W_{i'}(\tau, \ldots, \tau+m-1\})$ for all $i' \in \{1, \ldots, k\} \backslash \{i\}$ for all $i \in \{1, \ldots, k\}$ and*
- *the individualized patroller strategy $\boldsymbol{\pi}^f(\boldsymbol{s})$ is such that*

$$V_{\boldsymbol{\pi}^f(\boldsymbol{s}),\bullet}(Q, T, m) = V(Q, T, m), \tag{6.26}$$

  for all $f \in \{1, \ldots, k\}$.

The first condition for equality in lemma 6.3.3 is that for each pure attack there is only one possible pure patroller contributing towards it's capture and the second is that each individualized patroller strategy is optimal in the classic patrolling game with the same (Q, T, m). The proof of the lemma follows from the fact that no individual can contribute more than they could have in the classic patrolling game.

Proof. For the first part of the lemma consider $(\boldsymbol{s}, \boldsymbol{\phi})$ the optimal strategy pair for the game $G_1(k, 1, Q, T, m)$ then for all $f \in \{1, ..., k\}$ we have

$$P(\boldsymbol{s}, \boldsymbol{\phi}, f) \leq \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} \sum_{j \in N} \sum_{\tau=0}^{T-m} s_{i_1,\ldots,i_k} \varphi_{j,\tau} \mathbb{I}_{\{j \in W_{(i_f)}(\{\tau,\ldots,\tau+m-1\})\}} \tag{6.27}$$
$$= \sum_{i_f=1}^{|\mathcal{W}|} s_f^{i_f} \varphi_{j,\tau} \mathbb{I}_{\{j \in W_{(i_f)}(\{\tau,\ldots,\tau+m-1\})\}}$$
$$= P(\boldsymbol{\pi}^f(\boldsymbol{s}), \boldsymbol{\phi}) \leq V(Q, T, m). \tag{6.28}$$

Therefore

$$V_1(k, 1, Q, T, m) = P(\boldsymbol{s}, \boldsymbol{\phi}) = \sum_{f=1}^{k} P(\boldsymbol{s}, \boldsymbol{\phi}, f) \leq kV(Q, T, m),$$

and along with the trivial bound of $V(k, 1, Q, T, m) \leq 1$ we get the first result of the lemma.

For the furthermore part of the lemma we note that the first condition gives us equality in equation (6.27) and the second condition gives us equality in equation (6.28) and hence

$$V_1(k, 1, Q, T, m) = P(\boldsymbol{s}, \boldsymbol{\phi}) = \sum_{f=1}^{k} P(\boldsymbol{s}, \boldsymbol{\phi}, f) = kV(Q, T, m)$$

□

Lemma 6.3.3 shows us that having additional patrollers can at most linearly increase the probability of catching the attacker. This is extremely useful, as it means that if we can find a scheduler who can guarantee the conditions we reach equality. That is we can find a scheduler who can coordinate the patrollers to each collect distinct attacks while each individuals patroller strategy is optimal in the classic game. This essentially comes down to proving the lower bound for a scheduler's strategy, which meet the sufficient condition given in the lemma.

We now present results on the value of the game $G_1(k, 1, Q, T, m)$. First consider the game $G_1(|N|, 1, Q, T, m)$ in which the number of patrollers is equal to the number of nodes, then having each patroller wait at a node means the scheduler will catch every pure attack regardless of m.

Lemma 6.3.4. *For the game $G_1(k, l, Q, T, m)$ for any graph $Q = (N, E)$, for any $k \geq |N|$ and any $l \geq 1$, for any $m \geq 1$ and for any $T \geq m$ we have*

$$V_1(k, l, Q, T, m) = l.$$

The proof of lemma 6.3.4 follows by a pure scheduler $\boldsymbol{W}$ which has the first $|N|$ patrollers wait at distinct nodes for the entire time-horizon. With it proved we will restrict ourselves to looking at games such that $k < |N|$.

Proof. Let $\boldsymbol{W} \in \mathcal{W}^k$ be such that $W_i(t) = \beta^{-1}(i)$ for $1 \leq i \leq |N|$ where $\beta : N \to \{1, ..., |N|\}$ is a bijection. Then for any $(j, \tau) \in \mathcal{A}$ we have $j \in W_i(\{\tau, ..., \tau + m - 1\})$ for some $1 \leq i \leq |N|$ and hence $P_a(\boldsymbol{W}, (j, \tau)) = 0$ and therefore

$$P_s(\boldsymbol{W}, ((j_1, \tau_1), ..., (j_l, \tau_l))) = l.$$

So it is clear that

$$V_1(k, l, Q, T, m) = l.$$

□

When Q is Hamiltonian it is possible to find a scheduler strategy $\boldsymbol{s} \in \zeta$ that satisfies the two conditions for equality in lemma 6.3.3 by considering spreading out the k patrollers who follow a Hamiltonian cycle. Moreover, it is possible to do the same when the classic game omits an optimal solution which is a random minimal full-node cycle. We first demonstrate how to satisfy the conditions when the graph is Hamiltonian.

To ensure each of the k patroller catches distinct pure attacks while they individually follow a classic optimal patrol strategy, we can use the random Hamiltonian cycle with some initial spacing between the k patrollers. To ensure that they only catch distinct pure attacks we can have patroller start m places ahead on the Hamiltonian cycle. An alternative to this is to space the patrollers evenly throughout the Hamiltonian cycle. Note that with both of these we will have a limit of how many patrollers we can place before they start to overlap in which pure attacks they possibly catch. We define the following scheduler strategy, which will give optimality in Hamiltonian graphs.

Definition 6.3.5. The *random spread Hamiltonian* scheduler strategy $\boldsymbol{s}_{\text{rsH}}$ is such that the probability of playing $\boldsymbol{W}$ is

$$s_{\text{rsH},\beta_1}(\boldsymbol{W}) = \begin{cases} \frac{1}{|N|} & \text{if } \boldsymbol{W} \in X, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} X = \{\boldsymbol{W} \in \mathcal{W}^k \, | W_f(t) = H(t + i + (r-1)m \mod |N|) \, \forall t \in \mathcal{J} \, \forall f \in \{1, ..., k\}, \\ \text{for some } i \in \{0, ..., |N| - 1\}, \end{aligned} \tag{6.29}$$

for some Hamiltonian cycle H.

Note that randomization of $\boldsymbol{s}_{\text{rsH}}$ is equally choosing $i \in \{0, ..., |N| - 1\}$ which determines the starting position of the first patroller, namely $H(i)$. The scheduler then starts the r^{th} patroller at the node $H(i + (r-1)m \mod |N|)$ for $r = 2, ..., k$.

Then for the time-horizon each patroller follows the Hamiltonian cycle H. Note that the randomization is only done for the first patroller and all other patrollers follow a definitive pattern, of being m places ahead in the Hamiltonian cycle, given the starting place of the first patroller. They key is that the r^{th} and $r+1^{\text{th}}$ patrollers in each pure scheduler are at least m apart in an effort to never catch the same pure attack (j, τ). However it is possible that there is overlap of attacks but this just means that all pure attacks will be caught by the scheduler.

Theorem 6.3.6. *For the game $G_1(k, 1, Q, T, m)$ for all $k \geq 1$, for all $Q = (N, E) \in \mathcal{H}$, for all $m \geq 1$, for all $T \geq m$ we have*

$$V_1(k, 1, Q, T, m) = \min\left(1, \frac{km}{|N|}\right),$$

achieved by the random spread Hamiltonian scheduler strategy and the position uniform attacker. Moreover for all $l \geq 1$,

$$V_1(k, l, Q, T, m) = \min\left(l, \frac{klm}{|N|}\right),$$

Proof. Let us first consider $G_1(k, 1, Q, T, m)$ when $k \times \frac{m}{|N|} \geq 1$. Then for any $\boldsymbol{W} \in X$ (as in equation (6.29)) we have that for all $(j, \tau) \in \mathcal{A}$ that $j \in W_f(\{\tau, ..., \tau + m - 1\})$ for some $f \in \{1, ..., k\}$ as

$$\bigcup_{f=1}^{k} W_f(\{\tau, ..., \tau + m - 1\}) = N.$$

Therefore for any $(j, \tau) \in \mathcal{A}$ and for any pure scheduler $\boldsymbol{W} \in X$ we have

$$P_s(\boldsymbol{W}, (j, \tau)) = 1.$$

So in this case

$$V_1(k, 1, Q, T, m) = 1.$$

Secondly let us consider $G_1(k, 1, Q, T, m)$ when $k \times \frac{m}{|N|} < 1$, then we will use lemma 6.3.3. Let us show that the first condition of the lemma holds by considering any $\boldsymbol{W} \in X$, then for all $\tau \in \mathcal{T}$ we have that if $j \in W_f(\{\tau, ..., \tau + m - 1\})$ then

$$j \notin W_{f'}(\{\tau, ..., \tau + m - 1\}) = W_i(I'),$$

where $I' = \{\tau + (i' - i)m \mod T, ..., (\tau + (f' - f)m \mod T) + m - 1\}$, for all $f' \in \{1, ..., k\} \setminus \{f\}$ for all $j \in N$ and for all $f \in \{1, ..., k\}$. For the second condition we have that the f^{th} patrollers individualized strategy is $\boldsymbol{\pi}^f(\boldsymbol{s}_{\text{rsH}}) = \boldsymbol{\pi}_{\text{rH}}$ where $\boldsymbol{\pi}_{\text{rH}}$ is the random Hamiltonian strategy using the Hamiltonian cycle H' such that $H'(t) = H(t + (f-1)m)$ for all $t \in \{0, ..., |N| - 1\}$. Now by lemma 2.3.27 we know that $\boldsymbol{\pi}_{\text{rH}}$ is optimal and so

$$V_{\boldsymbol{\pi}_{\text{rH}}, \bullet}(Q, T, m) = V(Q, T, m) = \frac{m}{|N|}.$$

Therefore, as both conditions for equality in lemma 6.3.3 are satisfied, we have

$$V_1(k,1,Q,T,m) = kV(Q,T,m) = \frac{km}{|N|}.$$

Hence

$$V_1(k,1,Q,T,m) = \min\left(1, \frac{km}{|N|}\right),$$

with the final part of the lemma following from lemma 6.3.2.

□

As seen in the proof of theorem 6.3.6 in the case of $\frac{km}{|N|} \geq 1$ there is no need for randomization for the scheduler. Furthermore, in the case of $\frac{km}{|N|} < 1$, full and uniform randomization over $i \in \{0, ..., |N|-1\}$ for the initial starting position of the first patroller is not needed. In the case that the distance between all starting nodes is exactly m, the equal randomization over $i \in \{0, ..., (n-(k-1)m)-1\}$ is sufficient. This is because we can simply relabel of the k patrollers when they change initial starting positions. More alternative spreading ideas can be used, as long as they guarantee starting each patroller at least m nodes along the Hamiltonian cycle. Another possible way to get the result in theorem 6.3.6 is to use the following lemma to provide an upper bound, instead of lemma 6.3.3.

Lemma 6.3.7. *For the game $G_1(k,1,Q,T,m)$ for any $k \geq 1$, for any graph $Q = (N,E)$, for any $m \geq 1$ and for any $T \geq m$ we have*

$$V_1(k,1,Q,T,m) \leq \min\left(1, \frac{\omega^*}{|N|}\right) \leq \min\left(1, \frac{km}{|N|}\right),$$

where ω^* is the maximum number of distinct nodes that k simultaneous walks can visit in m units of time.

Lemma 6.3.7 is analogous result to lemma 2.3.16 the classic patrolling game, in which the attacker choices a node uniformly for a fixed commencement time. As with the classic case, the proof of lemma is done by stating how well the best pure scheduler can do against the position uniform attacker strategy.

Proof. Consider $\boldsymbol{\phi} \in \Phi$ such that $\varphi_{j,0} = \frac{1}{|N|}$ for each $j \in N$. Then during the interval $\{0, ..., m-1\}$ the maximum number of distinct nodes a scheduler can have k patrollers visit is ω^*, thus

$$V_1(k,1,Q,T,m) \leq \max_{\boldsymbol{W} \in \mathcal{W}^k} P(\boldsymbol{W}, \boldsymbol{\phi}) = \frac{\omega^*}{|N|}.$$

The second part of the inequality follows as $\omega^* \leq km$ as at best each of the k patrollers can visit m distinct nodes which are distinct from all other patrollers.

□

Lemma 6.3.7 provides a tight bound with the bound given by the random spread Hamiltonian cycle (see appendix C.2) to give the same result as in theorem 6.3.6. It also clarifies the idea that the best thing the scheduler can do is have patrollers catch distinct attacker.

Simplification and expansion operations (as seen in section 3.3.2) can be used on the graph Q in the patrolling game with multiple patrollers $G_1(k,1,Q,T,m)$. Considering these graphical operators allows us to consider embedded strategies and bounds and we get analogous results as merging nodes in simplification can only possibly help the k patrollers and therefore scheduler and splitting nodes can only help the attacker.

Theorem 6.3.8. *For any graph Q which can be x-simplified to Q^{-x} and y-expanded to Q^{+y}, for all $x \geq 1$, for all $y \geq 1$, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V_1(k,1,Q^{+y},T,m) \leq V_1(k,1,Q,T,m) \leq V_1(k,1,Q^{-x},T,m).$$

Proof. We first show the result for simplification, $V_1(k,1,Q,T,m) \leq V_1(k,1,Q^{-x},T,m)$. For any $\boldsymbol{W} \in \mathcal{W}(Q,T,m)^k$ there exists $\boldsymbol{W'} \in \mathcal{W}(Q^{-x},T,m)^k$ such that

$$P(\boldsymbol{W},\boldsymbol{\phi}) = P(\boldsymbol{W'},\boldsymbol{\phi}),$$

for any attacker strategy $\boldsymbol{\phi} \in \Phi(Q,T,m)$. Hence

$$\begin{aligned} V_1(k,1,Q,T,m) &= \min_{\boldsymbol{\phi} \in \Phi(Q,T,m)} \max_{\boldsymbol{W} \in \mathcal{W}(Q,T,m)^k} P(\boldsymbol{W},\boldsymbol{\phi}) \\ &\leq \min_{\boldsymbol{\phi} \in \Phi(Q,T,m)} \max_{\boldsymbol{W} \in \mathcal{W}(Q^{-x},T,m)^k} P(\boldsymbol{W},\boldsymbol{\phi}) = V(k,1,Q^{-x},T,m) \end{aligned}$$

Secondly we get that $V_1(k,1,Q^{+y},T,m) \leq V_1(k,1,Q,T,m)$ as Q^{+y} can be y-simplified to Q. □

Theorem 6.3.8 lets us get results for non-Hamiltonian graphs, by using a full-node cycle (as seen in section 3.3.4). As in the classic game we remark that the optimality of such strategies we will develop depend on the attack length m. We present the scheduler's strategy which uses a minimal full-node cycle for a graph and then present a similar result to theorem 6.3.6. Recall that a minimal full-node cycle repeats of length $|N|+x$ repeats x nodes and that the choice of such a full-node cycle is arbitrary as it is minimal and therefore the number of repeated nodes are minimal.

Definition 6.3.9. A *random spread minimal full-node cycle* scheduler strategy $\boldsymbol{s}^Q_{\text{RSMFNC}} \in \zeta(k,1,Q,T,m)$l, using a minimal full-node cycle W^Q_{MFNC} of length $|N|+x$, with a Hamiltonian expansion of the graph Q into Q^{+x}, is such that

$$s_{\beta_1(\boldsymbol{W})} = \sum_{\boldsymbol{W}^+ \in \mathcal{W}(k,1,Q^{+x},T,m) \text{s.t.} \boldsymbol{W}^+ = \mathcal{N}^{+x}(\boldsymbol{W})} s_{\text{rsH},\beta_1^+(\boldsymbol{W})},$$

where $\mathcal{N}^{+x}$ is the node mapping of the expansion from Q to Q^{+x} and $s_{\text{rsH},\beta_1^+(\boldsymbol{W}^+)}$ is the probability from the random spread Hamiltonian that $\boldsymbol{W}^+$ is played.

That is $\boldsymbol{s}_{\text{RSMFNC}}$ is the embedded scheduler strategy playable in $G(k,1,Q,T,m)$ from the optimal strategy $\boldsymbol{s}_{\text{rsH}}$ for the game $G(k,1,Q^{+x},T,m)$ which is Hamiltonian.

Lemma 6.3.10. *For the game $G(k,1,Q,T,m)$ for any $k \in \mathbb{N}$, for any graph Q with a minimal full-node cycle W^Q_{MFNC} of length $|N|+x$, for any $m \geq 1$ and for any $T \geq m$ we have*

$$V_1(k,1,Q,T,m) \geq \frac{km}{n+x},$$

achieved by the random spread minimal full-node cycle scheduler strategy $\boldsymbol{s}_{RSMFNC}$.

Proof. We have by theorem 6.3.8

$$V_1(k,1,Q,T,m) \geq V_1(k,1,Q^{+x},T,m),$$

where Q^{+x} is an x-expansion of Q which is Hamiltonian by node-splitting repeated nodes in W^Q_{MFNC}. Then by theorem 6.3.6

$$V_1(k,1,Q^{+x},T,m) = \min\left(1, \frac{km}{n+x}\right),$$

so therefore we obtain the result $V_1(k,1,Q,T,m) \geq \frac{km}{n+x}$. □

Lemma 6.3.10 allows us to get results on the game $G_1(k,1,Q,T,m)$ where the graph Q is a f-partite graph. Recall that when $f=2$ the complete bipartite graph $K_{a,b}$ is not Hamiltonian and when $f \geq 3$ the graph $K_{a_1,\dots,a_f}$ may be Hamiltonian dependent on the size of the partite sets $a_1, \dots, a_f$.

Theorem 6.3.11. *For the game $G_1(k,1,K_{a,b},T,m)$ for any $k \geq 1$, for any $b \geq a$ ($a,b \in \mathbb{N}$), for any $m \geq 2$ and for any $T \geq m+1$ we have*

$$V_1(k,1,K_{a,b},T,m) = \min\left(1, \frac{km}{2b}\right).$$

For the game $G_1(k,1,K_{a_i,\dots,a_l},T,m)$ for any $k \geq 1$, for any $f \geq 3$ with $1 \geq a_1 \geq \dots \geq a_f$ ($a_i \in \mathbb{N}$ for $i \in \{1,\dots f\}$), for any $m \geq 2$ and for any $T \geq m+1$ we have

$$V_1(k,1,K_{a_i,\dots,a_f},T,m) = \begin{cases} \min\left(1, \frac{km}{\sum_{i=1}^{l} a_i}\right) & \text{if } \sum_{i=1}^{f-1} a_i \geq a_f, \\ \min\left(1, \frac{km}{2a_f}\right) & \text{if } \sum_{i=1}^{f-1} a_i < a_f. \end{cases}$$

The proof of theorem 6.3.11 follows from the lemma 6.3.10 and lemma 6.3.3. Therefore, the random spread minimal full-node cycle and the 2-polygonal attack are optimal strategies on a mutlipartite graph.

Proof. For the game $G_1(k, 1, K_{a,b}, T, m)$ on the complete bipartite graph $K_{a,b}$ a minimal full node cycle alternates between the two partite sets and therefore repeats $b - a$ nodes and so by lemma 6.3.10 we have that

$$V_1(k, 1, Q, T, m) \geq \frac{km}{a + b + b - a} = \frac{km}{2b}.$$

Then lemma 6.3.3 and lemma 2.3.28 give us that

$$V_1(k, Q, T, m) \leq \min\left(1, k\frac{m}{2b}\right).$$

Therefore we have the result of equality.

For the game $G_1(k, 1, K_{a_i,\dots,a_l}, T, m)$ on the complete f-partite graph let us first consider the case $\sum\limits_{i=1}^{f-1} a_i \geq a_f$, for which $K_{a_i,\dots,a_l}$ is Hamiltonian and thus is given by theorem 6.3.6. In the second case of $\sum\limits_{i=1}^{f-1} a_i < a_f$, $K_{a_i,\dots,a_l}$ but has a minimal full-node cycle which repeats $a_f - \sum\limits_{i=1}^{f} a_i$ nodes (which are not in largest partite set) and so by lemma 6.3.10 we have that

$$V_1(k, 1, Q, T, m) \geq \frac{km}{2a_f}.$$

Then lemma 6.3.3 and theorem 3.5.7 give us that

$$V_1(k, 1, Q, T, m) \leq \min\left(1, \frac{km}{2a_f}\right).$$

Therefore we have the result of equality in this case. □

Such is the power using the minimal full-node cycle, we can get the solution for the line graph and generalised star graph, when they have a random minimal full-node cycle patroller strategy being optimal in the classic game. This allows us to use lemma 6.3.3 and 6.3.10 to easily get results.

Lemma 6.3.12. *For the game $G_1(k, 1, L_n, T, m)$ for all $k \geq 1$, for all $n \geq 2$, for all $m \geq 2(n-1)$ and for all $T \geq n + m - 1$ we have*

$$V_1(k, 1, L_n, T, m) = \frac{km}{2(n-1)}.$$

For the game $G_1(k, 1, S_n^{\boldsymbol{l}}, T, m)$ for all $k \geq 1$, for all $n \geq 3$, for all $\boldsymbol{f} \in \mathbb{N}^n$, for all $m \geq 2(\max_i f_i + 1)$ and for all $T \geq 2\max_{i\in\{1,\dots,n\}}(f_i) + m + 1$ we have

$$V_1(k, 1, S_n^{\boldsymbol{f}}, T, m) = \frac{km}{2(n + \sum\limits_{i=1}^{n} f_i)}.$$

Having seen the usefulness of having additional patrollers when the classic game has an optimal random minimal full-node cycle, we move onto looking at another scheduler strategy which has patrollers collect distinct attacks. Consider the idea of intercepting patrols and the covering bound (in section 2.3.2). It is easy to adapt the covering strategies for a single patroller to one for the scheduler controlling k patrollers. Let C be a minimal covering set for the game $G(Q,T,m)$ with a covering number of $\mathcal{C}_{Q,T,m} = |C|$. In the classic game these are played with equal probability, to form the covering strategy, however in the game $G(k,1,Q,T,m)$ the k patrollers can be used to have the scheduler simultaneously play k out of $\mathcal{C}_{Q,T,m}$ intercepting patrols in C. Therefore, if $k \geq \mathcal{C}_{Q,T,m}$ the scheduler can guarantee the capture of the attacker and if $k < \mathcal{C}_{Q,T,m}$ then by playing schedules were the k patrollers play distinct patrols in C we can achieve the following result.

Lemma 6.3.13. *For the game $G_1(k,1,Q,T,m)$, for any $k \geq 1$, for any Q, for any $m \geq 1$ and for any $T \geq m$ we have*

- *if $k \geq \mathcal{C}_{Q,T,m}$ then*
$$V_1(k,1,Q,T,m) = 1.$$
- *if $k < \mathcal{C}_{Q,T,m}$ then*
$$V_1(k,1,Q,T,m) \geq \frac{k}{\mathcal{C}_{Q,T,m}}.$$
Moreover if the covering strategy is optimal for the game $G(Q,T,m)$ then
$$V_1(k,1,Q,T,m) = \min\left(1, \frac{k}{\mathcal{C}_{Q,T,m}}\right).$$

Proof. Given a minimal covering set C for the game $G(Q,T,m)$ (such that $|C| = \mathcal{C}_{Q,T,m}$) let $X = \{\boldsymbol{W} \in \mathcal{W}(Q,T,m)^k \mid W_f \in C,\, W_f \neq W_{f'}\, \forall f \neq f'\}$. Then for the first case of $k \geq \mathcal{C}_{Q,T,m}$ let $\boldsymbol{W} \in X$ then

$$\bigcup_{f=1}^{k} W_f(\{\tau, ..., \tau+m-1\}) = N.$$

Therefore for any $(j,\tau) \in \mathcal{A}$ and for any pure scheduler $\boldsymbol{W} \in X$ we have

$$P_s(\boldsymbol{W},(j,\tau)) = 1.$$

So in this case

$$V_1(k,1,Q,T,m) = 1.$$

For the second case of $k < \mathcal{C}_{Q,T,m}$ we can form a scheduler strategy $\boldsymbol{s} \in \zeta(k,1,Q,T,m)$ such that

$$s_{\beta_1(\boldsymbol{W})} = \begin{cases} \frac{1}{\binom{\mathcal{C}_{Q,T,m}}{k}} & \text{if } \boldsymbol{W} \in X, \\ 0 & \text{otherwise.} \end{cases}$$

Then for any $(j,\tau) \in \mathcal{A}(Q,T,m)$ it is such that for the walk $W' \in C$ that $P(W',C) = 1$ and so as W' is an element of the schedule $\boldsymbol{W}$ with probability $\frac{k}{\mathcal{C}_{Q,T,m}}$ we have for any $(j,\tau) \in \mathcal{A}(Q,T,m)$ that

$$P(\boldsymbol{s},(j,\tau)) = \frac{k}{\mathcal{C}_{Q,T,m}}.$$

Hence

$$V_1(k,1,Q,T,m) \geq \frac{k}{\mathcal{C}_{Q,T,m}}.$$

Individual strategies from $\boldsymbol{s}$ are covering strategies for game $G(Q,T,m)$ and hence if they are optimal as assumed we get by lemma 6.3.3 that

$$V_1(k,1,Q,T,m) = \frac{k}{\mathcal{C}_{Q,T,m}}.$$

□

Again this allows us to solve $G_1(k,1,Q,T,m)$ when the covering strategy was optimal for $G(Q,T,m)$. Two such graphs are the line graph and generalised star graph when $m = 2$.

Lemma 6.3.14. *For the game $G(k,1,L_n,T,m)$ for all $k \geq 1$, for all $n \geq 2$, for all $T \geq 2$ we have*

$$V_1(k,1,L_n,T,2) = \frac{k}{\lceil \frac{n}{2} \rceil}.$$

For the game $G(k,1,S_n^{\boldsymbol{f}},T,m)$, for all $k \geq 1$, for all $n \geq 3$, for all $\boldsymbol{f} \in \mathbb{N}^n$, for all $T \geq 2$ we have

$$V_1(k,1,S_n^{\boldsymbol{f}},T,2) = \begin{cases} \frac{k}{1+\sum_{r=1}^{n} \frac{f_r+1}{2}} & \text{if } f_r \text{ is odd for all } r = 1,...,n, \\ \frac{k}{\sum_{r=1}^{n} \lceil \frac{f_r+1}{2} \rceil} & \text{if } f_r \text{ is even for some } r = 1,...,n. \end{cases}$$

We have seen that the adaptation of an optimal patrolling strategy $\boldsymbol{\pi}^* \in \Pi(Q,T,m)$ into an optimal scheduler strategy $\boldsymbol{s}^* \in \zeta(k,1,Q,T,m)$ can be done when patrollers can be coordinated to catch distinct pure attackers. In such cases the value of the game scales linearly with the number of patrollers, up to the natural upper limit of 1, meaning that at some point the inclusion of another patroller means the scheduler is guaranteed to win. Furthermore, the last patroller who contributes an increase to bring the value of the game to 1 may contribute less than others and so. Using such results it is easier to consider how many patrollers are required to meet a certain performance threshold. Patrol strategies in the classic game which do not allow the spreading of patrollers to catch distinct attackers require much more work for the scheduler in order to find the optimal strategy. Such things are to be expected, as we have already seen for low attack lengths improvements to random minimal full-node cycle strategies are needed and improved by using other strategies at weakly performing nodes. However the use of such improvement strategies do not easily allow multiple patrollers to spread out and catch distinct pure attacks. For example consider the game $G(L_n, T, m)$ for $m \in M_5^{L_n}$ an optimal solution plays the random minimal full-node cycle and two intercepting patrols at either end of the line. Augmenting this optimal solution for a scheduler with k patrollers requires more thought, as simply using k patrollers on one intercepting patrol provides no additional benefit. The question remains should one patroller be split between the two intercepting patrols and one play the random minimal full-node cycle? This provides an area for future work in augmenting optimal strategies which required improvement in the game with only a single patroller.

6.3.2 Collaborative attackers, who need all attackers to succeed

In the patrolling game $G_2(k, l, Q, T, m)$ we have k patrollers controlled by a scheduler and l attackers controlled by a *mastermind*. The mastermind, controlling the l attackers, picks a pure mastermind strategy $\boldsymbol{a} = ((j_1, \tau_1), ..., (j_l, \tau_l)) \in \mathcal{A}(Q, T, m)^l$ (as usual omitting (Q, T, m) when clear), in which the mastermind chooses the r^{th} attackers pure strategy (j_r, τ_r) for $r \in \{1, ..., l\}$. While in terms of pure strategies l attackers may coordinate and form a strategy formed that could be formed by a mastermind, there is a true difference at the level of randomization when l individual attackers are collaborating. A mastermind randomizes over the entire collection of strategies in $\mathcal{A}^l$, whereas l selfish attackers can only individually randomize over $\mathcal{A}$ unable to coordinate themselves. Essentially the mastermind is able to sync the l attacker's randomization using a single distribution. Let $\beta_2 : \mathcal{A}^l \to \{1, ..., |\mathcal{A}|^l\}$ be an arbitrary chosen bijection to number the pure mastermind strategies. We denote a mastermind's mixed strategy by $\boldsymbol{c} \in \varsigma$, where ς is the collection of mastermind mixed strategies, and denote the probability that the mastermind plays the pure strategy $\boldsymbol{a} = ((j_1, \tau_1), ..., (j_l, \tau_l))$ by $\varphi_{(j_1,t_1),...,(j_l,t_l)} = c_{\beta_2(\boldsymbol{a})}$. The game $G_2(k, l, Q, T, m)$ is a two player game with the scheduler attempting to win against the mastermind and this setup of strategies will be the same for the this game remaining two game variants $G_3(k, l, Q, T, m)$, $G_4(k, l, Q, T, m)$ (in sections 6.3.3, 6.3.4) while each of $G_i(k, l, Q, T, m)$ for $i = 2, 3, 4$ has a different payoff structure.

In the game $G_2(k, l, Q, T, m)$ with the scheduler choosing $\boldsymbol{W} \in \mathcal{W}^k$ and the mastermind choosing $\boldsymbol{a} \in \mathcal{A}^l$ we define the pure mastermind payoff as

$$\begin{aligned} P_m(\boldsymbol{W}, \boldsymbol{a}) &= \prod_{r=1}^{l} \prod_{f=1}^{k} \mathbb{I}_{\left\{j_r \in W_f(\{\tau_r, ..., \tau_r + m - 1\})\right\}} \\ &= \mathbb{I}_{\left\{j_r \notin W_f(\{\tau_r, ..., \tau_r + m - 1\}) \forall f \in \{1, ..., k\}, r \in \{1, ..., l\}\right\}}. \end{aligned}$$

That is the mastermind wins if and only if none of the l attackers are caught and in order to maintain the zero-sum nature of the game we define the pure payoff for the scheduler as

$$\begin{aligned} P_s(\boldsymbol{W},\boldsymbol{a}) &= 1 - P_m(\boldsymbol{W},\boldsymbol{a}) \\ &= \mathbb{I}_{\left\{\exists f \in \{1,\ldots,k\}, r \in \{1,\ldots,l\} \text{ s.t. } j_r \in W_f(\{\tau_r,\ldots,\tau_r+m-1\})\right\}}. \end{aligned} \tag{6.30}$$

That is the scheduler wins if and only if the mastermind loses by having any of the l attacker caught. To put the pure game in matrix form define $\mathcal{P}_2$ the pure payoff matrix in terms of the scheduler's pure payoff.

For mixed strategies $\boldsymbol{s} \in \zeta$ for the scheduler and $\boldsymbol{c} \in \varsigma$ for the mastermind we define the payoff of the game $G_2(k,l,Q,T,m)$ as

$$\begin{aligned} P(\boldsymbol{s},\boldsymbol{c}) = \boldsymbol{s}\mathcal{P}_2\boldsymbol{c}^T &= \sum_{\boldsymbol{W}\in\mathcal{W}^k} \sum_{\boldsymbol{a}\in\mathcal{A}^l} s_{\beta_1(\boldsymbol{W})} c_{\beta_2(\boldsymbol{a})} P_s(\boldsymbol{W},\boldsymbol{a}) \\ &= \sum_{i_1=1}^{|\mathcal{W}|} \cdots \sum_{i_k=1}^{|\mathcal{W}|} \sum_{j_1\in N} \sum_{t_1=0}^{T-m} \cdots \sum_{j_l\in N} \sum_{t_l=0}^{T-m} \xi, \end{aligned} \tag{6.31}$$

where $\xi = s_{i_1,\ldots,i_k} \varphi_{(j_1,t_1),\ldots,(j_l,t_l)} P_s((W_{(i_1)},\ldots,W_{(i_k)}),((j_1,\tau_1),\ldots,(j_l,\tau_l)))$. That is the payoff is the probability that the mastermind loses, or in other words the probability that any l attacker is caught. We now define the value of the game as

$$V_2(k,l,Q,T,m) = \max_{\boldsymbol{s}\in\zeta} \min_{\boldsymbol{c}\in\varsigma} P_s(\boldsymbol{s},\boldsymbol{c}) = \min_{\boldsymbol{c}\in\varsigma} \max_{\boldsymbol{s}\in\zeta} P_s(\boldsymbol{s},\boldsymbol{c}). \tag{6.32}$$

From the fact that the mastermind loses if any of their l attackers are caught in the game $G_2(k,l,Q,T,m)$, we can easily show that any mastermind pure strategy should have l identical attackers and that only these such strategies can be played with a non-zero probability in a mixed mastermind strategy.

Lemma 6.3.15. *For the game $G_2(k,l,Q,T,m)$ for any $k \in \mathbb{N}$, for any $l \in \mathbb{N}$, for any graph Q, for any $T \geq m$, for any $m \in \mathbb{N}$ then there exists some mastermind's strategy $\boldsymbol{c} \in \varsigma$ which is optimal and is such that*

$$\varphi_{\boldsymbol{a}} = 0 \quad \forall\, \boldsymbol{a} \in \mathcal{A}^l \setminus \{\boldsymbol{a} \in \mathcal{A}^l \mid (j_1,\tau_1) = \ldots. = (j_l,\tau_l)\}.$$

Furthermore if $V_2(k,l,Q,T,m) < 1$ such mastermind strategies are the only ones which can be optimal.

The proof of lemma 6.3.15 follows by showing that the mastermind can improve the performance of a strategy which doesn't have the l attackers use the same pure attack.

Proof. Consider, for the sake of contradiction, that there only exists optimal mastermind strategies such that the condition is not met. Take such a mastermind strategy $\boldsymbol{c}^*$ playing $\boldsymbol{a} \in \mathcal{A}^l$ with probability $\varphi^*_{\boldsymbol{a}}$. Then there is some $\boldsymbol{a}$ in which $(j_{r'},\tau_{r'}) \neq (j_{r''},\tau_{r''})$ for some attacker indices $r', r'' \in \{1,\ldots,l\}$ such that $\varphi^*_{\boldsymbol{a}} \neq 0$. Then we can construct another mastermind strategy $\boldsymbol{c} \in \varsigma$ such that the attacker with index r'' performs the same attack as the attacker with index r'. That is $\boldsymbol{c}$ plays $(j_1, \tau_1), ..., (j_l, \tau_l)$ with probability

$$\varphi_{(j_1,\tau_1),...,(j_l,\tau_l)} = \begin{cases} \sum\limits_{(j_{r''},\tau_{r''})\in\mathcal{A}} \varphi^*_{(j_1,t_1),...,(j_l,t_l)} & \text{if } (j_{r'}, \tau_{r'}) = (j_{r''}, \tau_{r''}), \\ 0 & \text{if } (j_{r'}, \tau_{r'}) \neq (j_{r''}, \tau_{r''}), \end{cases}$$

for all $((j_1, \tau_1), ..., (j_l, \tau_l)) \in \mathcal{A}^l$. Then we can show that for any pure scheduler strategy $\boldsymbol{W} \in \mathcal{W}^k$ that $P(\boldsymbol{W}, \boldsymbol{c}) \leq P(\boldsymbol{W}, \boldsymbol{c}^*)$. To see this consider

$$\begin{aligned} P(\boldsymbol{W}, \boldsymbol{c}) &= \sum_{\boldsymbol{a}\in\mathcal{A}^l} \varphi_{\boldsymbol{a}} \mathbb{I}_{\{\exists x\in\{1,...,l\} \text{ s.t. } j_x\in \bigcup\limits_{f=1}^{k} W_f(\{\tau_x,....,\tau_x+m-1\})\}} \\ &= \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t. } \exists x\in\{1,...,l\} \text{ s.t. } j_x\in \bigcup\limits_{f=1}^{k} W_f(\{\tau_x,....,\tau_x+m-1\})} \varphi_{\boldsymbol{a}} \\ &= \sum_{z=1}^{4} \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_z} \varphi_{\boldsymbol{a}} = \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_1} \varphi_{\boldsymbol{a}} + \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_4} \varphi_{\boldsymbol{a}} \\ &= \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_1 \text{ and } (j_{r'},\tau_{r'})=(j_{r''},\tau_{r''})} \left(\sum_{(j_{r''},\tau_{r''})\in\mathcal{A}} \varphi^*_{(j_1,t_1),...,(j_l,t_l)} \right) \\ &\quad + \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_4 \text{ and } (j_{r'},\tau_{r'})=(j_{r''},\tau_{r''})} \left(\sum_{(j_{r''},\tau_{r''})\in\mathcal{A}} \varphi^*_{(j_1,t_1),...,(j_l,t_l)} \right) \\ &= \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_1} \varphi^*_{\boldsymbol{a}} + \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_4} \varphi^*_{\boldsymbol{a}} \\ &\leq \sum_{z=1}^{4} \sum_{\boldsymbol{a}\in\mathcal{A}^l \text{ s.t } \mathbb{E}_z} \varphi^*_{\boldsymbol{a}} = P(\boldsymbol{W}, \boldsymbol{c}^*), \end{aligned} \tag{6.33}$$

where:

- $\mathbb{E}_1$ is the event that $\exists x \in \{1, ..., l\}$ s.t. $j_x \in \bigcup\limits_{f=1}^{k} W_f(\{\tau_x, ..., \tau_x + m - 1\})$ and $j_{r'} \notin \bigcup\limits_{f=1}^{k} W_f(\{\tau_{r'}, ..., \tau_{r'} + m - 1\})$, $j_{r''} \notin \bigcup\limits_{f=1}^{k} W_f(\{\tau_{r''}, ..., \tau_{r''} + m - 1\})$. That is $\mathbb{E}_1$ is the event that an attacker is caught by a patroller but both the r' and r'' indexed attackers are not caught,

- $\mathbb{E}_2$ is the event that $\exists x \in \{1, ..., l\}$ s.t. $j_x \in \bigcup\limits_{f=1}^{k} W_f(\{\tau_x, ..., \tau_x + m - 1\})$ and $j_{r'} \in \bigcup\limits_{f=1}^{k} W_f(\{\tau_{r'}, ..., \tau_{r'} + m - 1\})$, $j_{r''} \notin \bigcup\limits_{f=1}^{k} W_f(\{\tau_{r''}, ..., \tau_{r''} + m - 1\})$. That is $\mathbb{E}_2$ is the event that the r' indexed attacker is caught but the r'' indexed attacker is not caught,

- $\mathbb{E}_3$ is the event that $\exists x \in \{1, ..., l\}$ s.t. $j_x \in \bigcup\limits_{f=1}^{k} W_f(\{\tau_x, ..., \tau_x + m - 1\})$ and $j_{r'} \notin \bigcup\limits_{f=1}^{k} W_f(\{\tau_{r'}, ..., \tau_{r'} + m - 1\})$, $j_{r''} \in \bigcup\limits_{f=1}^{k} W_f(\{\tau_{r''}, ..., \tau_{r''} + m - 1\})$.

That is $\mathbb{E}_3$ is the event that the r'' indexed attacker is caught but the r' indexed attacker is not caught, and

- $\mathbb{E}_4$ is the event that $\exists x \in \{1, ..., l\}$ s.t. $j_x \in \bigcup_{f=1}^{k} W_f(\{\tau_x, ..., \tau_x + m - 1\})$ and $j_{r'} \in \bigcup_{f=1}^{k} W_f(\{\tau_{r'}, ..., \tau_{r'} + m - 1\})$, $j_{r''} \in \bigcup_{f=1}^{k} W_f(\{\tau_{r''}, ..., \tau_{r''} + m - 1\})$. That is $\mathbb{E}_4$ is the event that both the r' indexed attacker and the r'' indexed attacker are caught.

This process can be repeated for any indices r', r'' until no such indices exist and thus we reach a contradiction and thus have the first part of the lemma. For the furthermore part notice that we have strict inequality in equation (6.33) as $V_2(k, l, Q, T, m) < 1$ and thus it is possible to have the events $\mathbb{E}_2$ and $\mathbb{E}_3$ occur. In contrast when $V_2(k, l, Q, T, m) = 1$ these events do not take place and there is equality in equation (6.33).

□

Let us now assume that $V_2(k, l, Q, T, m) < 1$, so the scheduler can not be guaranteed to catch any of the mastermind's attackers, then we know that we can restrict the set of mastermind strategies to place all attackers together in time and space. Therefore, we restrict the set of pure attacks for the mastermind's mixed strategy, with the mastermind choosing a distribution among the *restricted* pure mastermind set

$$\mathcal{A}^l_{\text{Res}} = \{\boldsymbol{a} \in \mathcal{A}^l \mid (j_1, \tau_1) = ... = (j_l, \tau_l)\}.$$

A distribution among $\mathcal{A}^l_{\text{Res}}$ is equivalent to a distribution among $\mathcal{A}$ as all attackers follow the same pure attack under any realisation of the distribution. Knowing this, the mastermind can utilize the known optimal distribution among $\mathcal{A}$ for the game with one attacker $G_2(k, 1, Q, T, m)$ in the game with multiple attackers $G_2(k, l, Q, T, m)$.

Lemma 6.3.16. *For the game $G_2(k, l, Q, T, m)$ for any $k \in \mathbb{N}$, for any $l \in \mathbb{N}$, for any graph Q, for any $m \in \mathbb{N}$ and for any $T \geq m$ we have*

$$V_2(k, l, Q, T, m) = V_2(k, 1, Q, T, m).$$

Moreover, if $\boldsymbol{\phi}^* \in \Phi(Q, T, m)$ is the optimal strategy for the game $G_2(k, 1, Q, T, m)$ then $\boldsymbol{c} \in \varsigma(k, l, Q, T, m)$ such that

$$\varphi_{\boldsymbol{a}} = \begin{cases} \varphi^*_{j_1, t_1} & \text{if } \boldsymbol{a} = ((j_1, t_1), ..., (j_l, t_l)) \in \mathcal{A}^l_{Res}, \\ 0 & \text{otherwise,} \end{cases}$$

is optimal for the game $G_2(k, l, Q, T, m)$, where $\varphi^*_{j_1, t_1}$ is the probability of playing $(j_1, t_1) \in \mathcal{A}$ in $\boldsymbol{\phi}^*$.

Proof. Let $\boldsymbol{c}$ be as described in the lemma we seek to show it is optimal. By the definition of $\boldsymbol{c}$ we know that $P(\boldsymbol{s},\boldsymbol{c}) = P(\boldsymbol{s},\boldsymbol{\phi}^*)$ for any $\boldsymbol{s} \in \zeta$ and therefore $V_2(k,l,Q,T,m) \leq P(\boldsymbol{s}^*,\boldsymbol{c}) = P(\boldsymbol{s}^*,\boldsymbol{\phi}^*) = V_2(k,1,Q,T,m)$. However it is clear from the pure scheduler payoff, in equation (6.30), that it is non-decreasing in the number of attackers and hence so is the mixed payoff and hence so is the value

$$V_2(k,l,Q,T,m) \geq V_2(k,1,Q,T,m).$$

Therefore, we have

$$V_2(k,l,Q,T,m) = P(\boldsymbol{s}^*,\boldsymbol{c}) = P(\boldsymbol{s}^*,\boldsymbol{\phi}^*) = V_2(k,1,Q,T,m).$$

Moreover $\boldsymbol{c}$ is optimal for the game $G_2(k,l,Q,T,m)$.

□

We note the equivalence of the games with one attacker and multiple patrollers, that is $G_2(k,1,Q,T,m) \equiv G_1(k,1,Q,T,m)$, and so

$$V_2(k,1,Q,T,m) = V_1(k,1,Q,T,m).$$

Therefore we can use the work done in section 6.3.1 to get results for the game $G_2(k,1,Q,T,m)$ and therefore by using lemma 6.3.16 get results on the game $G_2(k,l,Q,T,m)$.

6.3.3 Collaborative attackers, who want as many attackers to succeed as possible

In the patrolling game $G_3(k,l,Q,T,m)$ we have k patrollers who are controlled by a scheduler and l attackers controlled by a mastermind. The scheduler and mastermind have the same strategies as in the game $G_2(k,l,Q,T,m)$, however in the game $G_3(k,l,Q,T,m)$ we model the mastermind attempting to make as many attackers succeed as possible. So for the scheduler choosing $\boldsymbol{W} \in \mathcal{W}^k$ against the mastermind choosing $\boldsymbol{a} \in \mathcal{A}^l$ in the game $G_3(k,l,Q,T,m)$ we define the pure mastermind payoff as

$$P_m(\boldsymbol{W},\boldsymbol{a}) = \sum_{r=1}^{l} \left(\prod_{f=1}^{k} \mathbb{I}_{\left\{j_r \notin W_f(\{\tau_r,\ldots,\tau_r+m-1\})\right\}} \right).$$

That is the mastermind gets a payoff equal to the number of attackers who succeed in their attacks and in order to maintain the zero-sum nature of the game we define the pure scheduler payoff as

$$P_s(\boldsymbol{W},\boldsymbol{a}) = l - \sum_{r=1}^{l} \left(\prod_{f=1}^{k} \mathbb{I}_{\left\{j_r \notin W_f(\{\tau_r,\ldots,\tau_r+m-1\})\right\}} \right) = \sum_{r=1}^{l} \mathbb{I}_{\left\{j_r \in \bigcup_{f=1}^{k} W_f(\{\tau_r,\ldots,\tau_r+m-1\})\right\}}.$$

That is the scheduler gets a payoff equal to the number of attackers caught. Using the scheduler's pure payoff we define the matrix form for the game $G_3(k,l,Q,T,m)$ as $\mathcal{P}_3$. The mixed payoff (in terms of the scheduler) is given by

$$\begin{aligned} P(\boldsymbol{s},\boldsymbol{c}) = \boldsymbol{s}\mathcal{P}_3\boldsymbol{c}^T &= \sum_{\boldsymbol{W}\in\mathcal{W}^k}\sum_{\boldsymbol{a}\in\mathcal{A}^l} s_{\beta_1(\boldsymbol{W})}c_{\beta_2(\boldsymbol{a})}P_s(\boldsymbol{W},\boldsymbol{a}) \\ &= \sum_{i_1=1}^{|\mathcal{W}|}\cdots\sum_{i_k=1}^{|\mathcal{W}|}\sum_{j_1\in N}\sum_{t_1=0}^{T-m}\cdots\sum_{j_l\in N}\sum_{t_l=0}^{T-m}\xi, \end{aligned}$$

where $\xi = s_{i_1,\dots,i_k}\varphi_{(j_1,t_1),\dots,(j_l,t_l)}P_s((W_{(i_1)},...,W_{(i_k)}),((j_1,\tau_1),...,(j_l,\tau_l)))$. So $P(\boldsymbol{s},\boldsymbol{c})$ is the expected number of attackers caught by a scheduler using $\boldsymbol{s}\in\zeta$ against a mastermind using $\boldsymbol{c}\in\varsigma$. We now define the value of the game as

$$V_3(k,l,Q,T,m) = \max_{\boldsymbol{s}\in\zeta}\min_{\boldsymbol{c}\in\varsigma}P(\boldsymbol{s},\boldsymbol{c}) = \min_{\boldsymbol{c}\in\varsigma}\max_{\boldsymbol{s}\in\zeta}P(\boldsymbol{s},\boldsymbol{c}).$$

From the pure mastermind payoff we can see that each attacker being successful is independent of the choice for all other attackers by the mastermind. That is $P(\boldsymbol{W},\boldsymbol{a}) = \sum_{r=1}^{l}P(\boldsymbol{W},a_r)$ where

$$P(\boldsymbol{W},a_r) = \left(\prod_{f=1}^{k}\mathbb{I}_{\{j_r\notin W_f(\{\tau_r,\dots,\tau_r+m-1\})\}}\right).$$

This allows us to consider how each attacker should be distributed without having to consider other attackers.

Lemma 6.3.17. *For the game $G_3(k,l,Q,T,m)$ for any $k\in\mathbb{N}$, for any $l\in\mathbb{N}$, for any graph Q, for any $m\in\mathbb{N}$ and for any $T\geq m$ we have*

$$V_3(k,l,Q,T,m) = lV_3(k,1,Q,T,m).$$

Proof. We start by obtaining a lower bound and then an upper bound for the game by considering how individuals can act.

We know that $P(\boldsymbol{W},\boldsymbol{a}) = \sum_{r=1}^{l}P(\boldsymbol{W},a_r)$ and so

$$P(\boldsymbol{s},\boldsymbol{c}) = \sum_{\boldsymbol{W}\in\mathcal{W}^k}\sum_{r=1}^{l}\sum_{(j,\tau)\in\mathcal{A}} s_{\beta_1(\boldsymbol{W})}\varphi_{r,j,\tau}P(\boldsymbol{W},a_r) = \sum_{r=1}^{l}P(\boldsymbol{s},\boldsymbol{\phi}_r),$$

where $\boldsymbol{\phi}_r$ is the r^{th} attackers individual distribution with a probability of playing (j,τ) given by

$$\varphi_{r,j,\tau} = \sum_{\boldsymbol{a}\in\{((j_1,\tau_1),\dots,(j_l,\tau_l))\in\mathcal{A}^l|j_r=j,\tau_r=\tau\}}\varphi_{\boldsymbol{a}}.$$

Therefore as choosing $\boldsymbol{c} \in \varsigma$ is equivalent to choosing $\boldsymbol{\phi}_r \in \Phi$ for $r = 1, ..., l$ we have

$$\begin{aligned}
V_3(k,l,Q,T,m) = \max_{\boldsymbol{s}\in\zeta}\min_{\boldsymbol{c}\in\varsigma} P_s(\boldsymbol{s},\boldsymbol{c}) &= \max_{\boldsymbol{s}\in\zeta}\min_{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_l\in\Phi}\sum_{r=1}^{l} P(\boldsymbol{s},\boldsymbol{\phi}_r)\\
&= \max_{\boldsymbol{s}\in\zeta}\sum_{r=1}^{l}\min_{\boldsymbol{\phi}_r\in\Phi} P(\boldsymbol{s},\boldsymbol{\phi}_r) = l\max_{\boldsymbol{s}\in\zeta}\min_{\boldsymbol{\phi}\in\Phi} P(\boldsymbol{s},\boldsymbol{\phi})\\
&= lV_3(k,1,Q,T,m).
\end{aligned}$$

□

We note the equivalence of the games with one attacker and multiple patrollers, that is $G_3(k,1,Q,T,m) \equiv G_1(k,1,Q,T,m)$, and so

$$V_3(k,1,Q,T,m) = V_1(k,1,Q,T,m).$$

Therefore we can use the work done in section 6.3.1 to get results for the game $G_3(k,1,Q,T,m)$ and therefore by using lemma 6.3.17 get results on the game $G_3(k,l,Q,T,m)$.

6.3.4 Collaborative attackers, who need one attacker to succeed

In the patrolling game $G_4(k,l,Q,T,m)$ we have k patrollers who are controlled by a scheduler and l attackers controlled by a mastermind. The scheduler and mastermind have the same strategies as in the game $G_2(k,l,Q,T,m)$, however in the game $G_4(k,l,Q,T,m)$ we model any attacker succeeding results in a win for the mastermind. So for the scheduler choosing $\boldsymbol{W} \in \mathcal{W}^k$ against the mastermind choosing $\boldsymbol{a} \in \mathcal{A}^l$ in the game $G_4(k,l,Q,T,m)$ we define the pure mastermind payoff as

$$\begin{aligned}
P_m(\boldsymbol{W},\boldsymbol{a}) &= 1 - \prod_{r=1}^{l}\prod_{f=1}^{k}\mathbb{I}_{\left\{j_r\in W_f(\{\tau_r,\ldots,\tau_r+m-1\})\right\}}\\
&= \mathbb{I}_{\left\{\exists r\in\{1,\ldots,l\}\text{ s.t. } j_r\notin W_f(\{\tau_r,\ldots,\tau_r+m-1\})\forall f\in\{1,\ldots,k\}\right\}}.
\end{aligned}$$

That is the mastermind wins if any attacker succeeds and in order to maintain the zero-sum nature of the game we define the pure scheduler payoff as

$$\begin{aligned}
P_s(\boldsymbol{W},\boldsymbol{a}) &= \prod_{r=1}^{l}\prod_{f=1}^{k}\mathbb{I}_{\left\{j_r\in W_f(\{\tau_r,\ldots,\tau_r+m-1\})\right\}}\\
&= \mathbb{I}_{\left\{\exists r\in\{1,\ldots,l\}\text{ s.t. } j_r\notin W_f(\{\tau_r,\ldots,\tau_r+m-1\})\forall f\in\{1,\ldots,k\}\right\}}.
\end{aligned}$$

That is the scheduler wins if they catch all attackers. Unlike all the other patrolling games seen in this thesis we note that it is possible to get a mastermind strategy which can guarantee a win in $G_4(k,l,Q,T,m)$ for certain parameters as well as it being possible to get a scheduler strategy which is a guaranteed win for other parameters. We define the matrix form of the game $G_4(k,l,Q,T,m)$ as $\mathcal{P}_4$. The game has a mixed payoff (in terms of the scheduler) is given by

$$\begin{aligned} P(\boldsymbol{s},\boldsymbol{c}) = \boldsymbol{s}\mathcal{P}_4\boldsymbol{c}^T &= \sum_{\boldsymbol{W}\in\mathcal{W}^k}\sum_{\boldsymbol{a}\in\mathcal{A}^l} s_{\beta_1(\boldsymbol{W})}c_{\beta_2(\boldsymbol{a})}P_s(\boldsymbol{W},\boldsymbol{a}) \\ &= \sum_{i_1=1}^{|\mathcal{W}|}\cdots\sum_{i_k=1}^{|\mathcal{W}|}\sum_{j_1\in N}\sum_{t_1=0}^{T-m}\cdots\sum_{j_l\in N}\sum_{t_l=0}^{T-m}\xi, \end{aligned}$$

where $\xi = s_{i_1,\ldots,i_k}\varphi_{(j_1,t_1),\ldots,(j_l,t_l)}P_s((W_{(i_1)},...,W_{(i_k)}),((j_1,\tau_1),...,(j_l,\tau_l)))$. So $P(\boldsymbol{s},\boldsymbol{c})$ is the probability that the scheduler wins the game, by catching all attackers, using $\boldsymbol{s}\in\zeta$ against a mastermind using $\boldsymbol{c}\in\varsigma$. We now define the value of the game as

$$V_4(k,l,Q,T,m) = \max_{\boldsymbol{s}\in\zeta}\min_{\boldsymbol{c}\in\varsigma}P(\boldsymbol{s},\boldsymbol{c}) = \min_{\boldsymbol{c}\in\varsigma}\max_{\boldsymbol{s}\in\zeta}P(\boldsymbol{s},\boldsymbol{c}).$$

In the game $G_4(k,l,Q,T,m)$ as the scheduler must catch all attackers in order to win we can get a result by relying on the independence of nodes at least m apart. Consider a set of nodes $L\subset N$ such that $d(j,j')\geq m$ for all $j,j'\in L$, then a mastermind strategy can be formed by having each of the l attackers choose nodes in L, such that they are always placed at distinct nodes if $|L|\geq l$, for a fixed commencement time. Intuitively the maximal cardinality of all such sets should be used to form the mastermind strategy to give the l attackers the best chance at using distinct nodes. The maximal cardinality of such a set is given by $\mathcal{L}_{Q,T,m}$.

Definition 6.3.18. For a maximal cardinality independent set L (with $|L| = \mathcal{L}_{Q,T,m}$) we define *independence mastermind strategy* $\boldsymbol{c}_{\text{Ind}}$ which chooses to play the pure strategy $\boldsymbol{a} = ((j_1,\tau_1),...,(j_l,\tau_l))\in\mathcal{A}^l$ with probability

$$\varphi_{\boldsymbol{a}} = \begin{cases} \frac{1}{|A_I|} & \text{for } \boldsymbol{a}\in A_I, \\ 0 & \text{otherwise.} \end{cases}$$

In which

$$A_I = \begin{cases} \begin{array}{l}\{\boldsymbol{a}\in\mathcal{A}^k \mid j_r\in L,\ \tau_r=0\quad\forall r\in\{1,...,l\}, \\ \qquad j_r\neq j_{r'}\,\forall r'\in\{1,...,l\}\setminus\{r\}\quad\forall r\in\{1,...,l\}\}\end{array} & \text{if } l\leq\mathcal{L}_{Q,T,m}, \\ \begin{array}{l}\{\boldsymbol{a}\in\mathcal{A}^k \mid j_r\in L,\ \tau_r=0\,\forall r\in\{1,...,l\}, \\ \qquad j_r\neq j_{r'}\,\forall r'\in\{1,...,\mathcal{L}_{Q,T,m}\}\setminus\{r\}\,\forall r\in\{1,...,\mathcal{L}_{Q,T,m}\}\}\end{array} & \text{if } l>\mathcal{L}_{Q,T,m}. \end{cases}$$

That is A_I is the set of pure mastermind strategies placing $\min(l,\mathcal{L}_{Q,T,m})$ attackers at distinct nodes in the independence set L and the remaining attacker able to choose any node. In the independence mastermind strategy pure strategies in the set A_I are equally picked. The performance of the independence mastermind strategy $\boldsymbol{c}_{\text{Ind}}$ gives an upper bound on the value of the game.

Lemma 6.3.19. *For the game $G_4(k,l,Q,T,m)$ for all $k \in \mathbb{N}$, for all $l \in \mathbb{N}$, for all graphs Q, for all $m \geq 1$ and for all $T \geq m$ we have*

$$V_4(k,l,Q,T,m) \leq \begin{cases} \frac{\binom{k}{l}}{\binom{\mathcal{I}_{Q,T,m}}{l}} & \text{if } l \leq \mathcal{L}_{Q,T,m}, \\ \mathbb{I}_{\{k \geq \mathcal{I}_{Q,T,m}\}} & \text{if } l > \mathcal{L}_{Q,T,m}, \end{cases}$$

achieved by the independence mastermind strategy $\boldsymbol{c}_{Ind}$. Moreover, if $\mathcal{I}_{Q,T,m} \geq l > k$ or if $l > \mathcal{I}_{Q,T,m} > k$ then

$$V_4(k,l,Q,T,m) = 0.$$

The proof of lemma 6.3.19 follows by evaluating the performance of the independence mastermind strategy and then noting when the strategy guarantees a win for the mastermind.

Proof. For any pure scheduler strategy $\boldsymbol{W} \in \mathcal{W}^k$ and any pure mastermind strategy $\boldsymbol{a} \in \mathcal{A}_I$ where L is the maximal independence set used we have that any patroller who catches an attacker who chose $(j,0) \in L \times \{0\}$ does not catch any other attackers $(j,0) \in (L \setminus \{j\}) \times \{0\}$ as $d(j,j') \geq m$ for any $j,j' \in L$. Hence for any pure scheduler strategy $\boldsymbol{W} \in \mathcal{W}^k$ against any pure mastermind strategy $\boldsymbol{a} \in \mathcal{A}_I$ the most nodes $j \in L$ which attackers can be caught at is k. This can be achieved by choosing $\boldsymbol{W'} \in \mathcal{W}^k$ such that $W_i(t) = j_q$ for all $t \in \mathcal{J}$ for $i = 1,...,k$, where $j_q \in L$ are such that $j_q \neq j_{q'}$ for all $q \in \{1,...,\mathcal{L}_{Q,T,m}\}$, $q' \in \{1,...,\mathcal{L}_{Q,T,m}\} \setminus \{q\}$. Knowing that a scheduler can at most see attacks at k distinct nodes we will need to get results in two cases, as if $l > \mathcal{L}_{Q,T,m}$ then not all attackers are at distinct nodes and if $l \leq \mathcal{L}_{Q,T,m}$ then all attackers are at distinct nodes.

- If $l > \mathcal{L}_{Q,T,m}$ then not all attackers are at distinct nodes and hence

$$P(\boldsymbol{W'}, \boldsymbol{c}_{\text{Ind}}) = \begin{cases} 1 & \text{if } k \geq \mathcal{L}_{Q,T,m}, \\ 0 & \text{if } k < \mathcal{L}_{Q,T,m}. \end{cases}$$

  That is, for the case that $k \geq \mathcal{L}_{Q,T,m}$ we have that $\boldsymbol{W'}$ places a patroller at each node in L and hence catches all attackers and in the case that $k < \mathcal{L}_{Q,T,m}$ there are not enough patrollers to do so and hence not all attackers are caught.

- If $l \leq \mathcal{I}_{Q,T,m}$ then all attackers are at distinct nodes and hence

$$P(\boldsymbol{W'}, \boldsymbol{c}_{\text{Ind}}) = \begin{cases} \frac{\binom{k}{l}}{\binom{\mathcal{I}_{Q,T,m}}{l}} & \text{if } k \geq l, \\ 0 & \text{if } k < l. \end{cases}$$

  That is, for the case that $k \geq l$ we have that $\boldsymbol{W'}$ has a

$$\frac{\binom{k}{l}}{\binom{\mathcal{I}_{Q,T,m}}{l}}$$

chance of placing its k patrollers at the same nodes randomly chosen for the l attacker by the mastermind strategy $\boldsymbol{c}_{\text{Ind}}$ and in the case that $k < l$ there are not enough patrollers to catch all attackers.

Hence,

$$V_4(k,l,Q,T,m) \leq P(\boldsymbol{W}', \boldsymbol{c}_{\text{Ind}}) = \begin{cases} \frac{\binom{k}{l}}{\binom{\mathcal{I}_{Q,T,m}}{l}} & \text{if } l \leq \mathcal{L}_{Q,T,m}, \\ \mathbb{I}_{\{k \geq \mathcal{I}_{Q,T,m}\}} & \text{if } l > \mathcal{L}_{Q,T,m}, \end{cases}$$

and along with the trivial lower bound of $V_4(k,l,Q,T,m) \geq 0$ we get that the moreover result of the lemma. $\square$

From lemma 6.3.19 it is clear that we are not going to get some useful result which decomposes the game with multiple attackers into games with a single attacker, as the joint coordination of the attackers in the mastermind's strategy can guarantee a loss which is not possible with less attackers. This means, while the game $G_4(k,1,Q,T,m) \equiv G_1(k,1,Q,T,m)$, so $V_4(k,1,Q,T,m) = V_1(k,1,Q,T,m)$ and our work subsection 6.3.1 is able to get results for $G_4(k,1,Q,T,m)$ we can not use this to get results for $G_4(k,l,Q,T,m)$ for $l > 1$. In the patrolling game $G_4(k,1,Q,T,m)$ the amount of attackers the mastermind has at their disposal is crucial to working out the performance of mastermind strategies.

6.4 Conclusion

In this chapter we have defined three extensions to the patrolling game: the patrolling game with edge distances $G(Q,D,T,m)$, the patrolling game with node dependent attack lengths $G(Q,T,\boldsymbol{m})$ and multiple player patrolling game $G_i(k,l,Q,T,m)$ for some $i \in \{1,2,3,4\}$. For the multiple player patrolling game we look at four variants depending on if the attackers where in collaboration and if so the payoff given to their mastermind.

For the patrolling game with edge distances $G(Q,D,T,m)$ we found that the value is the same as the game $G(Q,D',T,m)$ if there is some optimal patroller strategy which only uses edges such that the distance according to the mappings D and D' are the same. We also look at node-sublimation as a way to allow the removal of nodes while retaining the distance between nodes, using such a graphical operation we found that if an optimal attacker strategy did not use nodes which undergo node-sublimation then the value of the game remained the same. Using these results we were able to develop optimal strategies for the scenario of a central hub with multiple cities at various distances, which is sublimated version of the generalised star graph $S_n^{\boldsymbol{k}}$ defined in chapter 4, section 4.3. In general the idea of node-sublimation makes developing strategies for the game much easier at the cost of removing potential nodes to attack. Therefore when using node-sublimation we must take in consideration what it is we are modelling in order to decided if node-sublimation is a valid idea.

For the patrolling game with node dependent attack lengths $G(Q, T, \boldsymbol{m})$ we found that varying some attack lengths m_j does not affect the value of the game unless the node j is a minimally performing node under the optimal strategy. In which case the value only changes if all currently minimal performing nodes have their attack lengths changed and moreover if the value is not changed then the optimal patroller strategy remains optimal. As the value of the game is determined by the performance at nodes we look at using the Patrol Improvement Program(PIP) idea, as seen in chapter 3, section 3.4, to find improvements on patrolling strategies. The PIP is then used to find an improvement which is optimal for games on Hamiltonian graphs where m_j is constant aside from at a set of nodes which are instantaneous win nodes.

For the multiple player patrolling game we assume k patrollers are controlled by a scheduler, but define four variants depending on how the l attackers act and receive payoff. The first game variant $G_1(k, l, Q, T, m)$ models l selfish attackers, for this variant we showed that the value of the game depends entirely on $G_1(k, 1, Q, T, m)$ and that all l attackers use the same optimal strategy. Reducing the game to $G_1(k, 1, Q, T, m)$ means we only have to consider how the scheduler should act against one attacker, allowing us to get a variety of bounds on the game. In particular, we find the value of the game $G_1(k, 1, Q, T, m)$ when $Q \in \mathcal{H}$ or $Q \in \mathcal{KP}_f$ for any $f \geq 2$ and find a value for a range of attack lengths when $Q = L_n$ for some $n \geq 3$. We leave to future work the value of games in which the classic game does not omit optimal random full-node cycle strategies. The remaining three game variants $G_i(k, l, Q, T, m)$ for $i = 2, 3, 4$ models l collaborative attackers controlled by a mastermind. In both $G_2(k, l, Q, T, m)$, which models all attackers must succeed to win, and $G_3(k, l, Q, T, m)$, which models the want for the most attackers to succeed we were able to reduce the value of the game to the value of the game $G_2(k, 1, Q, T, m)$ and $G_3(k, 1, Q, T, m)$ respectively. Following this reduction we then noted that by definition $G_i(k, 1, Q, T, m) \equiv G_1(k, 1, Q, T, m)$ for $i = 2, 3$ so we rely on previous work in the section to find the values of such games for certain parameters. For the final game variant $G_4(k, l, Q, T, m)$, which models one attacker must succeed to win, we look at why it is not possible to get a reduction to the game $G_4(k, 1, Q, T, m)$ as it is possible that there is a strategy for the mastermind that guarantees a loss for the scheduler.

Chapter 7

Conclusion

7.1 Thesis summary

This thesis provided an in-depth look into 'Allocating patrolling resources to effectively thwart intelligent attackers', by furthering techniques and strategies for patrolling games. This allowed for various classes of patrolling games to be solved before extending those ideas to allow for more realistic scenarios to be modelled.

In chapter 2 the patrolling game $G(Q, T, m)$ was introduced following the work produced in [16] and [107]. Within this chapter the value of a game $V(Q, T, m)$ is defined along with a discussion of the performance of (mixed) patroller and attacker strategies, $\boldsymbol{\pi} \in \Pi(Q, T, m)$ and $\boldsymbol{\phi} \in \Phi(Q, T, m)$ respectively. In particular, the performance of $\boldsymbol{\pi}$ gives a lower bound on the value and the performance of $\boldsymbol{\phi}$ gives an upper bound on the value. Though it is possible to solve a patrolling game for particular a particular 3-tuple (Q, T, m), (graph Q, game length T, attack length m) by the use of a linear program, this is not useful for solving classes of patrolling games (patrolling games with certain general parameters). Patrolling games are often solved by finding tight lower and upper bounds on the value with the corresponding strategies being the optimal strategies. Within the work discussed, specifically referring to the diametric attacker strategy $\boldsymbol{\phi}_{\text{di}}$, we find an issue with the performance of the strategy, with this issue cascading into the solution for patrolling games when $Q = L_n$. However, we develop the time-limited diametric attacker strategy $\boldsymbol{\phi}_{\text{tdi}}$, which has the suggested performance and is able to replace $\boldsymbol{\phi}_{\text{di}}$ as the optimal attacker strategy for a class of patrolling games when $Q = L_n$. Prior work was summarized by a list of attacker and patroller strategies along with their respective bounds and for which classes of patrolling games they were optimal.

The main work of this thesis is provided in chapter 3, which proposes, introduces and describes new techniques and strategies which can be used in order to find the value of patrolling games. The techniques developed aid the calculation of strategy performances by reducing the computation required by removing pure walks which cannot be the best response to an attacker strategy. These reductions were accumulated in theorem 3.2.13. We saw that the notion of repeated node-identification and node-splitting form simplification $\mathcal{Q}^{-k}$ and expansion maps $\mathcal{Q}^{+l}$ respectively, and in particular that a patrolling strategy $\boldsymbol{\pi}' \in \Pi(\mathcal{Q}^{+l}(Q), T, m)$ can be embedded to make a patrolling strategy $\boldsymbol{\pi} \in \Pi(Q, T, m)$ and the lower bound on the value provided by the performance of $\boldsymbol{\pi}$. Importantly we saw that it is possible to perform an expansion to make any graph Q into a Hamiltonian graph $Q' \in \mathcal{H}$ and therefore embed the random Hamiltonian patroller strategy $\boldsymbol{\pi}_{\mathrm{rH}}$ into any patrolling game. We defined a minimal full-node cycle W^Q_{MFNC} as a closed walk which visits every node with minimal length and use this to create the random minimal full-node cycle patroller strategy $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$. The random minimal full-node cycle strategy $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$ is a crucially important patroller strategy as it ensures that all nodes receive at least one visit within a constant time interval, equal to the length of W^Q_{MFNC}. Moreover $\boldsymbol{\pi}^Q_{\mathrm{RMFNC}}$ is optimal for many classes of patrolling games and in particular is optimal when Q is an extended star graph, which was seen in chapter 4. To further improve the performance of patrolling strategies we developed the patrol improvement program(PIP) which can be used to find the best hybrid patroller strategy given a finite set of patrolling strategies. This chapter on general techniques and strategies was concluded by finding the solution to $G(Q,T,m)$ when $Q \in \mathcal{KP}_k$ for some $k \geq 2$. In particular we noted that the solution to $G(Q,T,m)$ when $Q = (N,E)$ is a non-complete k-partite graph is the same as the solution to $G(Q',T,m)$ where Q' is the complete version of graph Q (that is has all possible edges).

Chapter 4 applied the new techniques and strategies seen in chapter 3 to patrolling games on three new graphs which all extend the star graph: elongated star graphs $S^k_n \in \mathcal{SE}$, generalised star graphs $S^{\boldsymbol{k}}_n \in \mathcal{SG}$ and a graph made by connecting the centre node of multiple star graphs. The graph S^k_n is important as it models the scenario of a border with multiple rooms at one end of the border. Solving the game $G(S^k_n,T,m)$ for all graphical parameters $n \geq 3$ and $k \geq 1$ however required the decomposition of the set of attack lengths into 6 regions, $M^{S^k_n}_i$ for $i = 0,...,5$, in which each have different optimal strategies. Aside from the regions $M^{S^k_n}_0$ and $M^{S^k_n}_3$, solving $G(S^k_n,T,m)$ required the creation of different bespoke attacker strategies. In particular the time-centred attacker strategy $\boldsymbol{\phi}_{\mathrm{tc}}$ was created to have a performance equal to that for the random minimal full-node cycle $\boldsymbol{\pi}^{S^k_n}_{\mathrm{RMFNC}}$. For the regions $M^{S^k_n}_i$ for $i = 4,5$ we saw that $\boldsymbol{\pi}^{S^k_n}_{\mathrm{RMFNC}}$ performed weakly at a variety of nodes and improved the patroller strategy using PIP, which led to a further decomposition of these regions. In particular for one of these regions, namely $M^{S^k_n}_{5,1}$, we found four attacker strategies which are optimal, or near optimal, dependent on the parameter ρ. In general for lower attack lengths the creation of bespoke attacker strategies requires some thought and heavily depend on the attack length, this can be seen thoughout our work and in [107]. We then extended the elongated star graph to the generalised star graph $S^{\boldsymbol{k}}_n \in \mathcal{SG}$ which can be used to model multiple borders of varying lengths with a single hub location or multiple cities at varying distances from a given central hub. We provided solutions for $G(S^{\boldsymbol{k}}_n,T,m)$ for attack lengths which did not require the creation of a multitude of bespoke attacker strategies. However, in chapter 6 we saw that in the case when the scenario you are trying to model is that of multiple cities at varying distances from a central hub, we can solve such a scenario by modelling with distances on edges rather than having intermediate nodes along the connection between the hub and the cities.

In chapter 5 we focus on the patrolling game $G(Q,T,m)$ when Q is a tree (or forest). For the game $G(Q,T,2)$ we proved that the covering patrolling strategy $\boldsymbol{\pi}_{\text{Cov}}$ and the independent attacker strategy $\boldsymbol{\phi}_{\text{Ind}}$ are optimal. This was done by an algorithm which generates a minimal covering set and maximal independent set and ensures they have equal cardinality and hence the performance of $\boldsymbol{\pi}_{\text{Cov}}$ and $\boldsymbol{\phi}_{\text{Ind}}$ are equal and hence $V(Q,T,2) = \frac{1}{\mathcal{C}_{Q,T,2}} = \frac{1}{\mathcal{I}_{Q,T,2}}$. However as both $\mathcal{C}_{Q,T,2}$ and $\mathcal{I}_{Q,T,2}$ are not found explicitly for every tree, the value of the game is not explicit either and depends on the exact structure of the tree Q. While this algorithm may not terminate when Q is a not a tree (due to at some point having no leaf nodes), if it does terminate then we achieve the same result for the value. In addition, we conjectured about the optimality of the random minimal full-node cycle strategy $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ for the game $G(Q,T,m)$ when $Q=(N,E)$ is a tree. In particular, in conjecture 5.3.2 we suggest that if $m \geq |N|-1$ then $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$ is optimal and so

$$V(Q,T,m) = \frac{m}{2(|N|-1)}. \tag{7.1}$$

We have seen empirical evidence throughout this thesis that conjecture 5.3.2 holds for all currently studied patrolling games meeting the criteria. We also provided an intuitive reasoning in the form of a 'proof' of conjecture 5.3.2 in which we make the assumption that we can restrict the set of mixed patroller strategies we must consider using. Following this, a computer algorithm was presented which generates all trees for a given amount of nodes $|N|$. We then checked the conjecture by solving multiple patrolling games, for all parameters, by the use of a linear program. This was done exhaustively for all patrolling games such that $|N| \leq 8$ for which the conjecture held.

Unlike previous chapters chapter 6 does not attempt to solve the patrolling game $G(Q,T,m)$ for some set of parameters, but instead introduces three different extensions to the patrolling game. These extensions are: patrolling games with edge distances $G(Q,D,T,m)$, which introduced a distance edge map $D: E \to \mathbb{N}$ assigning each edge a distance with pure patroller strategies now becoming walks with a differing number of nodes visited; patrolling games with node-dependent attack lengths $G(Q,T,\boldsymbol{m})$, which changed the constant attack length m into $\boldsymbol{m}$, where m_j is the attack length at node $j \in N$; multi player patrolling games with k patrollers controlled by a single scheduler and l attackers $G_i(k,l,Q,T,m)$ for $i=1,2,3,4$, where for $i=1$ the attackers are uncoordinated and selfish and for $i=2,3,4$ the attackers are coordinated by a mastermind but each game has a different payoff for the mastermind. Each extension was introduced, stating the strategies and payoff for players before developing some techniques and strategies and solving some games. In particular, we solve the game on the distant general star graph $G(\widetilde{S}^{\boldsymbol{k}}_n, D_{n,\boldsymbol{k}}, T, m)$ which models the scenario of a central hub with cities at varying distances from the hub. We also found that we are able to reduce the problem of solving multi player patrolling games $G_2(k,l,Q,T,m)$ and $G_3(k,l,Q,T,m)$ into solving $G_1(k,1,Q,T,m)$. However, $G_4(k,l,Q,T,m)$ differs as for certain parameters the game is a guaranteed attacker win, which is not possible in any other patrolling game (as even one attacker succeeding in the game means the mastermind wins). This results in the game $G_4(k,l,Q,T,m)$ being more interesting to study and more importantly how intercepting patrol strategies are important as they guarantee that no attacker can succeed on a set of nodes.

Overall, this thesis covers specific in-depth aspects of the patrolling game $G(Q, T, m)$, beginning with describing the game and then proposing new techniques and strategies, most notably $\boldsymbol{\pi}^{Q}_{\text{RMFNC}}$. These techniques and strategies were then applied to a variety of patrolling games with different graphical structures. This thesis also explored general solutions for the large class of graphical structures, namely when Q is a tree. Furthermore, the patrolling game was extended to increase the various scenarios it can model.

7.2 Future work

Future work leading from this thesis could use the ideas in chapter 4, section 4.3.4, to determine how the to create patroller and attacker strategies for the game $G(S_n^{\boldsymbol{k}}, T, m)$, namely by identifying what full-node cycles are optimal when $2 < m < 2(k_{\max} + 1)$. In doing so we continue to find out how best to patrol multiple borders linked by a central hub location. In addition, work could consider finding bespoke attacker strategies for the game $G(S_n^k, T, m)$ when $m \in M_{5,1}^{S_n^k}$ and $\rho \in \{1, 3\}$ in order to achieve optimal strategies rather than near optimal strategies.

As mentioned in chapter 5, in our reasoning behind making conjecture 5.3.2 we assume that it is possible to ignore all pure patroller strategies which do not repeat closed walks for all of the time-horizon. Proving that such pure patroller strategies are not optimal would then allow us to prove conjecture 5.3.2. This would consequently have a large impact on the future study of patrolling games on trees as we would then know the value when $m \geq |N| - 1$ and would only have to consider games with $m < |N| - 1$.

Another avenue of future work is to consider how to convert our techniques and strategies to be applicable in the periodic patrolling game $G^p(Q, T, m)$. That is, adapt our techniques and strategies to deal with an innate restriction of $\mathcal{W}$ to $\mathcal{W}^p$ for which $(W(0), W(T-1)) \in E$, meaning that the patroller is forced to choose a base location and have a shift (of length T) which starts and ends at the base.

The extension of the patrolling game to one with multiple patrollers and multiple attackers, where only one attacker needs to succeed in order for the attackers to win in $G_4(Q, T, m)$, presents an interesting avenue in considering how additional patrollers should be utilised. In particular, how they can be used to ‘remove nodes’ from consideration for the other patrollers. Further to our current extensions we could consider introducing a ‘toll’ to use to certain edges for the patrollers or rewarding the attacker’s success dependent on the location they choose to attack. While this thesis focuses on an intelligent attacker, models with random attackers arriving at nodes according to a Poisson process have been introduced in [94], which heuristically find ‘optimal’ strategies and value. The work done in

[99] extended this patrolling problem to have distances and multiple patrollers for which we could compare our results in chapter 6 to. In addition, it would be interesting to study extensions in which the Poisson rate at nodes change throughout the game and comparing them to results for the patrolling game. In particular one could consider how successful attacks in such a model increase the rate of future attacks introducing vulnerabilities in a similar fashion to the game $G(Q, T, \boldsymbol{m})$.

Tangential work would look at applying similar techniques and strategies seen throughout this thesis to other related areas. Of particular interest would be the applications of techniques to the hide and seek ([5]) game and rendezvous game ([93]) on graphs.

Bibliography

[1] Steve Alpern. The search game with mobile hider on the circle. *Differential Games and Control Theory*, pages 181–200, 1974.

[2] Steve Alpern. Infiltration games on arbitrary graphs. *Journal of Mathematical Analysis and Applications*, 163(1):286–288, 1992.

[3] Steve Alpern. The Rendezvous Search Problem. *SIAM Journal on Control and Optimization*, 33(3):673–683, 1995.

[4] Steve Alpern. Rendezvous search on labeled networks. *Naval Research Logistics*, 49(3):256–274, 2002.

[5] Steve Alpern. Hide-and-seek games on a tree to which Eulerian networks are attached. *Networks*, 52(3):162–166, 2008.

[6] Steve Alpern, Vic Baston, and Skander Essegaier. Rendezvous search on a graph. *Journal of Applied Probability*, 36(1):223–231, 1999.

[7] Steve Alpern, Vic Baston, and Shmuel Gal. Network search games with immobile hider, without a designated searcher starting point. *International Journal of Game Theory*, 37(2):281–302, 2008.

[8] Steve Alpern and Robbert Fokkink. Accumulation games on graphs. *Networks*, 64(1):40–47, 2014.

[9] Steve Alpern, Robbert Fokkink, Shmuel Gal, and Marco Timmer. On Search Games That Include Ambush. *SIAM Journal on Control and Optimization*, 51(6):4544–4556, 2013.

[10] Steve Alpern, Robbert Fokkink, Roy Lindelauf, and Geert-Jan Olsder. The "Princess and Monster" Game on an Interval. *SIAM Journal on Control and Optimization*, 47(3):1178–1190, 2008.

[11] Steve Alpern, Robbert Fokkink, Marco Timmer, and Jerome Casas. Ambush frequency should increase over time during optimal predator search for prey. *Journal of the Royal Society Interface*, 8(64):1665–1672, 2011.

[12] Steve Alpern and Shmuel Gal. A Mixed-Strategy Minimax Theorem without Compactness. *SIAM Journal on Control and Optimization*, 26(6):1357–1361, 1988.

[13] Steve Alpern and Shmuel Gal. *The Theory of Search Games and Rendezvous*. International Series in Operations Research & Management Science. Kluwer Academic Publishers, Boston, 2003.

[14] Steve Alpern, Thomas Lidbetter, Alec Morton, and Katerina Papadaki. Patrolling a pipeline Book section Patrolling a Pipeline. *Proceedings. Lecture Notes in Computer Science*, 9996:129–138, 2016.

[15] Steve Alpern, Thomas Lidbetter, and Katerina Papadaki. Optimizing periodic patrols against short attacks on the line and other networks. *European Journal of Operational Research*, 273(3):1065–1073, 2019.

[16] Steve Alpern, Alec Morton, and Katerina Papadaki. Patrolling Games. *Operations Research*, 59(5):1246–1257, 2011.

[17] Edaward J. Anderson. Mazes: Search games on unknown networks. *Networks*, 11(4):393–397, 1981.

[18] Edaward J. Anderson and Miguel A. Aramendia. The search game on a network with immobile hider. *Networks*, 20(7):817–844, 1990.

[19] Edward J. Anderson and Richard R. Weber. The rendezvous problem on discrete locations. *Journal of Applied Probability*, 27(4):839–851, 1990.

[20] John M. Auger. An infiltration game on k arcs. *Naval Research Logistics*, 38(4):511–529, 1991.

[21] Rudolf Avenhaus and Marc Kilgour. Efficient distributions of arms-control inspection effort. *Naval Research Logistics*, 51(1):1–27, 2004.

[22] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184-185:78–123, 2012.

[23] Vic Baston. Note: Two rendezvous search problems on the line. *Naval Research Logistics*, 46(3):335–340, 1999.

[24] Vic Baston and F. A. Bostock. A generalized inspection game. *Naval Research Logistics*, 38(2):171–182, 1991.

[25] Vic Baston and Andrey Y. Garnaev. A fast infiltration game on n arcs. *Naval Research Logistics*, 43(4):481–489, 1996.

[26] Anatole Beck and Donald J. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429, 1970.

[27] Michael G.H. Bell, Urszula Kanturska, Jan-Dirk Schmöcker, and Achille Fonzone. Attacker-defender models and road network vulnerability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1872):1893–1906, 2008.

[28] Richard Bellman. An Optimal Search. *SIAM Review*, 5(3):274–274, 1963.

[29] Nelson M. Blachman. Prolegomena to optimum discrete search procedures. *Naval Research Logistics Quarterly*, 6(4):273–281, 1959.

[30] Nelson M. Blachman and Frank Proschan. Optimum Search for Objects Having Unknown Arrival Times. *Operations Research*, 7(5):625–638, 1959.

[31] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 2005.

[32] Pauline Boss, Lorraine Beaulieu, Elizabeth Wieling, William Turner, and Shulaika Lacruz. Healing loss, ambiguity, and trauma: A community-based intervention with families of union workers missing after the 9/11 attack in New York City. *Journal of Marital and Family Therapy*, 29(4):455–467, 2003.

[33] Scott Shorey Brown. Optimal Search for a Moving Target in Discrete Time and Space. *Operations Research*, 28(6):1275–1289, 1980.

[34] Franz Thomas Bruss and James B. Robertson. A survey of the linear-search problem. *Mathematical Scientist*, 13(2):75–89, 1988.

[35] Abraham Charnes and William W. Cooper. The Theory of Search: Optimum Distribution of Search Effort. *Management Science*, 5(1):44–50, 1958.

[36] Abraham Charnes and Roger G. Schroeder. On some stocxastic tactical antisubmarine games. *Naval Research Logistics Quarterly*, 14(3):291–311, 1967.

[37] Marek Chrobak, Claire Kenyon, John Noga, and Neal E Young. Incremental Medians via Online Bidding. *Algorithmica*, 50(4):455–478, 2008.

[38] Marek Chrobak and Claire Kenyon-Mathieu. SIGACT news online algorithms column 10: competitiveness via doubling. *SIGACT News*, 37(4):115–126, 2006.

[39] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*, pages 151–158, New York, New York, USA, 1971. ACM Press.

[40] José Correa, Tobias Harks, Vincent J.C. Kreuzen, and Jannik Matuschke. Fare evasion in transit networks. *Operations Research*, 65(1):165–183, 2017.

[41] Thomas L. Corwin. Search for a moving target under minimax estimation. *Journal of Applied Probability*, 18(1):167–180, 1981.

[42] Arnon Dagan and Shmuel Gal. Network search games, with arbitrary searcher starting point. *Networks*, 52(3):156–161, 2008.

[43] John M. Danskin. On Koopman's Addition Theorem in Search Theory. *Institute of Naval Studies, Cambridge, Massachusetts*, 13, 1964.

[44] Jacques de Guenin. Optimum Distribution of Effort: An Extension of the Koopman Basic Theory. *Operations Research*, 9(1):1–7, 1961.

[45] Francesco M. Delle Fave, Matthew Brown, Chao Zhang, Eric Shieh, Albert Xin Jiang, Heather Rosoff, Milind Tambe, and John P. Sullivan. Security games in the field: Deployments on a transit system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8758:103–126, 2014.

[46] Anders Dessmark, Pierre Fraigniaud, Dariusz R. Kowalski, and Andrzej Pelc. Deterministic Rendezvous in Graphs. *Algorithmica*, 46(1):69–96, 2006.

[47] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2005.

[48] Irit Dinur and Samuel Safra. On the hardness of approximating vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

[49] James M. Dobbie. Search theory: A sequential approach. *Naval Research Logistics Quarterly*, 10(1):323–334, 1963.

[50] James M. Dobbie. A Two-Cell Model of Search for a Moving Target. *Operations Research*, 22(1):79–92, 1974.

[51] Melvin Dresher. A Sampling Inspection Problem in Arms Control Agreements: A Game-Theoretic Analysis. Technical Report February, Rand Corp Santa Monica CA, 1962.

[52] Lester E. Dubins. A discrete evasion game. *Contributions to the Theory of Games*, 3:231–255, 1957.

[53] James N. Eagle and Alan R. Washburn. Cumulative search-evasion games. *Naval Research Logistics*, 38(4):495–510, 1991.

[54] Jack Edmonds and Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.

[55] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai):2589–2595, 2015.

[56] Sidney I. Firstman and Brian Gluss. Search Rules for Automatic Fault Detection. *Rand Corporation, Santa Monica, California. RM-2514*, 15, 1960.

[57] Wallace Franck. An Optimal Search Problem. *SIAM Review*, 7(4):503–512, oct 1965.

[58] Shmuel Gal. Minimax Solutions for Linear Search Problems. *SIAM Journal on Applied Mathematics*, 27(1):17–30, 1974.

[59] Shmuel Gal. Search Games with Mobile and Immobile Hider. *SIAM Journal on Control and Optimization*, 17(1):99–122, 1979.

[60] Shmuel. Gal. *Search games*. Academic Press New York, 1980.

[61] Shmuel Gal. Rendezvous Search on the Line. *Operations Research*, 47(6):974–976, 1999.

[62] Shmuel Gal. On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory*, 29(4):533–542, 2001.

[63] Shmuel Gal and Dan Chazan. On the Optimality of the Exponential Functions for Some Minimax Problems. *SIAM Journal on Applied Mathematics*, 30(2):324–348, 1976.

[64] Shumel. Gal and Edward J. Anderson. Search in a Maze. *Probability in the Engineering and Informational Sciences*, 4(3):311–318, 1990.

[65] Andrey Y. Garnaev. A remark on the princess and monster search game. *International Journal of Game Theory*, 20(3):269–276, 1992.

[66] Andrey Y. Garnaev. A remark on a helicopter and submarine game. *Naval Research Logistics (NRL)*, 40(5):745–753, 1993.

[67] Andrey Y. Garnaev, G. Garnaeva, and P. Goutal. On the infiltration game. *International Journal of Game Theory*, 26(2):215–221, 1997.

[68] Tristan Garrec. Continuous patrolling and hiding games. *European Journal of Operational Research*, 277(1):42–51, 2019.

[69] Edgar. N. Gilbert. Optimal Search Strategies. *Journal of the Society for Industrial and Applied Mathematics*, 7(4):413–424, 1959.

[70] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, 2011.

[71] Brian Gluss. An Optimum Policy for Detecting a Fault in a Complex System. *Operations Research*, 7(4):468–477, 1959.

[72] Brian Gluss. Approximately optimal one-dimensional search policies in which search costs vary through time. *Naval Research Logistics Quarterly*, 8(3):277–283, 1961.

[73] Constantine Gumacos. Analysis of an Optimum Sync Search Procedure. *IEEE Transactions on Communications*, 11(1):89–99, 1963.

[74] William B. Haskell, Debarun Kar, Fei Fang, Milind Tambe, Sam Cheung, and Lt Elizabeth Denicola. Robust protection of fisheries with COmPASS. *Proceedings of the National Conference on Artificial Intelligence*, 4:2978–2983, 2014.

[75] Ryusuke Hohzaki. An inspection game with multiple inspectees. *European Journal of Operational Research*, 178(3):894–906, 2007.

[76] Ryusuke Hohzaki. An inspection game with smuggler's decision on the amount of contraband. *Journal of the Operations Research Society of Japan*, 54(1):25–45, 2011.

[77] Ryusuke Hohzaki. A smuggling game with the secrecy of smuggler's information. *Journal of the Operations Research Society of Japan*, 55(1):23–47, 2012.

[78] Rufus Isaacs. *Differential Games*. Wiley, 1965.

[79] Albert Xin Jiang, Zhengyu Yin, Chao Zhang, Milind Tambe, and Sarit Kraus. Game-theoretic randomization for security patrolling with dynamic execution uncertainty. *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013*, 1(Aamas):207–214, 2013.

[80] Triantafyllos Karatrantos. Daesh : The archetype of hybrid terrorist organizations. 2019.

[81] Offer Kella. Star search—A different show. *Israel Journal of Mathematics*, 81(1-2):145–159, 1993.

[82] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*, 9781107096:156–176, 2011.

[83] Kensaku Kikuta and William H. Ruckle. Accumulation Games, Part 1: Noisy Search. *Journal of Optimization Theory and Applications*, 94(2):395–408, 1997.

[84] Kensaku Kikuta and William H. Ruckle. Continuous accumulation games on discrete locations. *Naval Research Logistics*, 49(1):60–77, 2002.

[85] Murali Kodialam and T. V. Lakshman. Detecting network intrusions via sampling: A game theoretic approach. *Proceedings - IEEE INFOCOM*, 3(C):1880–1889, 2003.

[86] Bernard O Koopman. The Theory of Search. *Operations Research*, 5(5):613–626, 1957.

[87] Joseph Varghese Kureethara. Hamiltonian Cycle in Complete Multipartite Graphs. *Annals of Pure and Applied Mathematics*, 13(2):223–228, 2017.

[88] Steven P. Lalley. A one-dimensional infiltration game. *Naval Research Logistics (NRL)*, 35(5):441–446, 1988.

[89] Steven P. Lalley and Herbert Robbins. Stochastic search in a convex region. *Probability Theory and Related Fields*, 77(1):99–116, 1988.

[90] Eric Langford. A continuous submarine versus submarine game. *Naval Research Logistics Quarterly*, 20(3):405–417, 1973.

[91] K. T. Lee. A firing game with time lag. *Journal of Optimization Theory and Applications*, 41(4):547–558, 1983.

[92] K. T. Lee. An evasion game with a destination. *Journal of Optimization Theory and Applications*, 46(3):359–372, 1985.

[93] Pierre Leone and Steve Alpern. Search-and-Rescue Rendezvous. 2016.

[94] Kyle Y. Lin, Michael P. Atkinson, Timothy H. Chung, and Kevin D. Glazebrook. A Graph Patrol Problem with Random Attack Times. *Operations Research*, 61(3):694–710, 2013.

[95] Kyle Y. Lin, Michael P. Atkinson, and Kevin D. Glazebrook. Optimal patrol to uncover threats in time when detection is imperfect. *Naval Research Logistics*, 61(8):557–576, 2014.

[96] Alejandro López-Ortiz and Sven Schuierer. The ultimate strategy to search on m rays? *Theoretical Computer Science*, 261(2):267–295, 2001.

[97] Alejandro López-Ortiz and Sven Schuierer. On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. *Theoretical Computer Science*, 310(1-3):527–537, 2004.

[98] David Matula. A Periodic Optimal Search. *American Mathematical Monthly*, 71(1):15–21, 1964.

[99] Richard G. McGrath and Kyle Y. Lin. Robust patrol strategies against attacks at dispersed heterogeneous locations. *International Journal of Operational Research*, 30(3):340–359, 2017.

[100] Richard D. McKelvey and Thomas R. Palfrey. Quantal Response Equilibria for Normal Form Games. *Games and Economic Behavior*, 10(1):6–38, 1995.

[101] J J. Meinardi. A sequentially compounded search game. *Theory of Games: Techniques and Applications*, 1964.

[102] Giuseppe Moscarini and Randall Wright. Introduction to Search Theory and Applications. *Journal of Economic Theory*, 145(4):1319–1324, 2010.

[103] Teruhisa Nakai. A sequential evasion-search game. *Mathematica Japonica*, 28:315–326, 1983.

[104] Teruhisa Nakai. A sequential evasion-search game with a goal. *Journal of the Operations Research Society of Japan*, 29(2):113–122, 1986.

[105] John Nash. Non-Cooperative Games. *The Annals of Mathematics*, 54(2):286, 1951.

[106] Oystein Ore. Note on Hamilton Circuits. *The American Mathematical Monthly*, 67(1):55, 1960.

[107] Katerina Papadaki, Steve Alpern, Thomas Lidbetter, and Alec Morton. Patrolling a Border. *Operations Research*, 64(6):1256–1269, 2016.

[108] Praveen Paruchuri, JP Pearce, and Janusz Marecki. Efficient Algorithms to Solve Bayesian Stackelberg Games for Security Applications. In *Aaai*, volume 1, pages 2–5, New York, New York, USA, 2008. ACM Press.

[109] Ljiljana Pavlovic. Search game on an odd number of arcs with immobile hider. *Yugoslav Journal of Operations Research*, 3(1):11–19, 1993.

[110] Federico Perea and Justo Puerto. Revisiting a game theoretic framework for the robust railway network design against intentional attacks. *European Journal of Operational Research*, 226(2):286–292, 2013.

[111] Sharon E. Perlman, Stephen Friedman, Sandro Galea, Hemanth P. Nair, Monika Ers-Sarnyai, Steven D. Stellman, Jeffrey Hon, and Carolyn M. Greene. Short-term and medium-term health effects of 9/11. *The Lancet*, 378(9794):925–934, 2011.

[112] James Pita, Manish Jain, Janusz Marecki, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Deployed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles international airport. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*, 9781107096:67–87, 2011.

[113] James Pita, Milind Tambe, Chris Kiekintveld, Shane Cullen, and Erin Steigerwald. GUARDS - Game Theoretic Security Allocation on a National Scale Categories and Subject Descriptors. *International Conference on Autonomous Agents and Multiagent Systems*, pages 37–44, 2011.

[114] Stephen M. Pollock. Sequential search and detection. Technical report, Massachusetts Instute of Technology Cambridge Operations Research Center, 1964.

[115] Stephen M. Pollock. A Simple Model of Search for a Moving Target. *Operations Research*, 18(5):883–903, 1970.

[116] Edward Posner. Optimal search procedures. *IEEE Transactions on Information Theory*, 9(3):157–160, 1963.

[117] Henry R. Richardson and Joseph H. Discenza. United States Coast Guard Computer-Assisted Search Planning System (CASP). *Naval Research Logistics Quarterly*, 27(4):659–680, 1980.

[118] William H. Ruckle. Ambushing Random Walks II: Continuous Models. *Operations Research*, 29(1):108–120, 1981.

[119] William H. Ruckle. Pursuit on a cyclic graph — The symmetric stochastic case. *International Journal of Game Theory*, 10(2):91–99, 1981.

[120] William H. Ruckle, Robert Fennell, Paul T. Holmes, and Charles Fennemore. Ambushing Random Walks I: Finite Models. *Operations Research*, 24(2):314–324, 1976.

[121] William H. Ruckle and John R. Reay. Ambushing Random Walks III: More Continuous Models. *Operations Research*, 29(1):121–129, 1981.

[122] M Sakaguchi. A sequential game of multi-opportunity infiltration. *Scientiae Mathematicae Japonicae*, 39:157–166, 1994.

[123] Javier Salmeron, Kevin Wood, and Ross Baldick. Analysis of Electric Grid Security Under Terrorist Threat. *IEEE Transactions on Power Systems*, 19(2):905–912, 2004.

[124] Maria P. Scaparra and Richard L. Church. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers and Operations Research*, 35(6):1905–1923, 2008.

[125] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. PROTECT: A deployed game theoretic system to protect the ports of the United States. *11th International Conference on Autonomous Agents and Multiagent Systems 2012, AAMAS 2012: Innovative Applications Track*, 1:1–8, 2012.

[126] Richard D. Smallwood and Edward J. Sondik. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Operations Research*, 21(5):1071–1088, 1973.

[127] Lawrence D. Stone. *Theory of optimal search*, volume 118. Elsevier, 1976.

[128] Milind Tambe. A Game Theoretic Approach for Allocation of Limited Security Resources Milind Tambe , University of Southern California Fernando Ordonez , University of Southern California. Technical report, University of Southern California, 2007.

[129] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.

[130] Marlin U. Thomas and Yair Nisgav. An infiltration game with time dependent payoff. *Naval Research Logistics Quarterly*, 23(2):297–302, 1976.

[131] Alexi Thompson. Terrorism and tourism in developed versus developing countries. *Tourism Economics*, 17(3):693–700, 2011.

[132] M Timmer. *Rendezvous on an interval and a search game on a star*. PhD thesis, Delft University of Technology, 2008.

[133] Ingrid Torjesen and Anne Gulland. Manchester doctors describe aftermath of bomb blast as NHS continues to treat casualties. *BMJ (Clinical research ed.)*, 357:j2628, 2017.

[134] Jason Tsai, Shyamsunder Rathi, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. IRIS – A tool for strategic security allocation in transportation networks. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*, 9781107096:88–106, 2011.

[135] Albert W. Tucker. Solving a Matrix Game by Linear Programming. *IBM Journal of Research and Development*, 4(5):507–517, 1960.

[136] Gabriel Valiente. *Algorithms on trees and graphs*. Springer Science & Business Media, 2013.

[137] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

[138] Bernhard von Stengel and Ralph Werchner. Complexity of searching an immobile hider in a graph. *Discrete Applied Mathematics*, 78(1-3):235–249, 1997.

[139] Willem A. Wagenaar. Generation of random sequences by human subjects: A critical survey of literature. *Psychological Bulletin*, 77(1):65–72, 1972.

[140] Alan R. Washburn. Search-Evasion Game in a Fixed Region. *Operations Research*, 28(6):1290–1298, 1980.

[141] Alan R. Washburn. Search for a Moving Target: The FAB Algorithm. *Operations Research*, 31(4):739–751, 1983.

[142] D. J. Wilson. Isaacs' princess and monster game on the circle. *Journal of Optimization Theory and Applications*, 9(4):265–288, 1972.

[143] Rong Yang, Benjamin Ford, Milind Tambe, and Andrew Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. *13th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2014*, 1(Aamas):453–460, 2014.

[144] Rong Yang, Christopher Kiekintveld, Fernando Ordóñez, Milind Tambe, and Richard John. Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence*, 195:440–469, 2013.

[145] Rong Yang, Fernando Ordonez, and Milind Tambe. Computing optimal strategy against quantal response in security games. *11th International Conference on Autonomous Agents and Multiagent Systems 2012, AAMAS 2012: Innovative Applications Track*, 2:584–591, 2012.

[146] Zhengyu Yin, Albert Xin Jiang, Milind Tambe, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, and John P. Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4):59–72, 2012.

[147] Samuel Zahl. An Allocation Problem with Applications to Operations Research and Statistics. *Operations Research*, 11(3):426–441, 1963.

[148] Mikhail Il'ich Zelikin. On a differential game with incomplete information. In *Doklady Akademii Nauk*, volume 202, pages 998–1000. Russian Academy of Sciences, 1972.

[149] Noemi Zoroa, Maria-Jose Fernandez Saez, and Procopio Zoroa. Search and Ambush Games with Capacities. *Journal of Optimization Theory and Applications*, 123(2):431–450, 2004.

Appendices

Appendix A

Patrolling games

A.1 Proof of the complete k-partite patrolling game value

For completeness we present the proof of theorem 3.5.7 which is analogous to the proof of that for a tripartite graph (lemma 3.5.5 and 3.5.6) which was shown in section 3.5

Proof. We begin by proving that $K_{a_1,\dots,a_k}$ (with $1 \le a_1 \le \dots \le a_k$) is Hamiltonian iff $\sum_{i=1}^{k-1} a_i \ge a_k$. To show that $\sum_{i=1}^{k-1} a_i \ge a_k \implies K_{a_1,\dots,a_k}$ is Hamiltonian we use Ore's theorem. Ore's theorem allows us to get the sufficient condition that for all $1 \le j \le k$ that $2\sum_{i \ne j} a_i \ge \sum_{i=1}^{k} a_i$. All of these inequalities are satisfied either by $1 \le a_1 \le \dots \le a_k$ (if $j \ne k$) or by the assumption of $\sum_{i=1}^{k-1} a_i \ge a_k$.

Next we show that $K_{a_1,\dots,a_k}$ being Hamiltonian $\implies \sum_{i=1}^{k-1} a_i \ge a_k$. As the graph is Hamiltonian it must exhibit a Hamiltonian cycle of the form

$$\{*_1, i_1, \dots, *_{a_k}, i_{a_k}, \sim_1, \sim_2, \dots, \sim_r\},$$

where $*$ and $\sim$ are listing of nodes in sets other than A_k and $i \in A_k$. As we know the cycle is of length $\sum_{i=1}^{k} a_i$ and that no individual node is repeated, it is clear that the number of nodes not in the set A_k is at least as many in A_k and hence $\sum_{i=1}^{k-1} a_i \ge a_k$.

Now we've proven a necessary and sufficient condition for $K_{a_1,\dots,a_k}$ to be Hamiltonian we can simply apply lemma 2.3.27 to get the result when $\sum_{i=1}^{k-1} a_i \ge a_k$.

We now focus on the graph when it is not Hamiltonian, that is the case of $\sum_{i=1}^{k-1} a_i < a_k$. In this case the patroller can use a random full-node cycle strategy on the minimal full-node cycle. A repetition of $a_k - \sum_{i=1}^{k-1} a_i$ is needed, giving a minimal full-node cycle of length $2a_k$ and hence , by theorem 3.3.26 arrive at the lower bound of bound

$$V(K_{a_1,\ldots,a_k}, T, m) \geq \frac{m}{2a_k}.$$

To find an equal upper bound we can use a 2-polygonal attack using the set A_k, then by lemma 3.3.16 we get an upper bound of

$$V(K_{a_1,\ldots,a_k}, T, m) \leq \frac{m}{2a_k}.$$

Hence we get the result for the value of the complete tripartite graph for $T \geq m+1$.

If m is even then we can reduce this condition to $T \geq m$ by considering a uniform attacker strategy $\boldsymbol{\phi}_{u,A_k}$ which is such that the probability of choosing the pure attack (j,τ) is

$$\varphi_{j,\tau} = \begin{cases} \frac{1}{a_k} & \text{if } j \in A_k, \tau = 0, \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that at most any pure patroller can only visit $\frac{m}{2}$ distinct nodes form A_k in the time period $\{0, \ldots, m-1\}$ and hence

$$V(K_{a_1,\ldots,a_k}, T, m) \leq \frac{m}{2a_k}.$$

Therefore when m is even $\boldsymbol{\phi}_{u,A_k}$ is optimal for $T \geq m$.

For the two furthermore parts of theorem we will first consider the addition of an edge (j,j') such that $j,j' \in A_i$ for some $i \in \{1, \ldots, k-1\}$, as this does not effect if the graph is Hamiltonian or not then we are left to only consider if it changes for the case of $\sum_{i=1}^{k-1} a_i < a_k$. In this case the upper bounds generated by lemma 3.3.16 and attacker strategy $\boldsymbol{\phi}_{u,A_k}$ still hold as there is still a distance of 2 between all nodes in A_k. Therefore along with equation (2.18) for adding edges to a graph the lower bound still holds. Hence there is no affect on our value or optimal strategies.

For the second furthermore part we consider the removal of an edge $(j,j') \in E$ such that there is no change in the length of the minimal node cycle. Then if the graph was Hamiltonian the graph is still Hamiltonian and so we are only left to consider if it changes the case of $\sum_{i=1}^{k-1} a_i < a_k$. In this case the lower bound generated using theorem 3.3.26 still holds. By considering adding the removed edge back into the graph we get by equation (2.18) that the upper bound still holds. Hence there is no affect on our value or optimal strategies. □

A.2 Proof of polygonal bound

For completeness we present the proof of lemma 3.3.16.

Proof. Let $D = \{v_1, ..., v_{|D|}\}$, then we divide the proof into two cases, case 1: if $d \geq m$ and case 2: if $d < m$. Noting that the attack structure is node-symmetric and non-increasing on $N_A = \{v_1, ..., v_{|D|}\}$ we can use corollary 3.2.17 and lemma 3.2.14 to reduce the problem of finding the performance of $\boldsymbol{\phi}_{\text{tdi}}$ to

$$V_{\bullet,\boldsymbol{\phi}_{\text{poly}}}(Q, T, m) = V_{\bullet,\boldsymbol{\phi}_{\text{poly}}}(Q, d + m - 1, m) = \max_{\omega \in \Omega_{\uparrow}} P(\omega, \boldsymbol{\phi}_{\text{poly}})$$

Case 1: In the case of $d \geq m$ we know that by lemma 3.2.16 that $\Omega_{\uparrow} = \{\omega_1, \omega_2, ..., \omega_{|D|}\}$ in which $\omega_i = ((v_i, d + m - 1))$ for all $i = 1, ..., |D|$. As the nodes in D are node-symmetric we can without loss of generality assume that $\omega_1 = ((v_1, d + m - 1))$ is the only walk to consider for the maximum. Therefore, the performance of $\boldsymbol{\phi}_{\text{poly}}$ is

$$V_{\bullet,\boldsymbol{\phi}_{\text{poly}}}(Q, d + m - 1, m) = P(\omega_1, \boldsymbol{\phi}_{\text{poly}}) = \sum_{t=0}^{d-1} \frac{1}{|D|d} = \frac{1}{|D|}.$$

Hence $V(Q, T, m) \leq \frac{1}{|D|}$ when $d \geq m$.

Case 2: In the case of $d < m$ we can again look at what elements are in $\Omega_{\uparrow}$. The sequence of nodes must be such $j_1 \in D$ and $j_i \in D \setminus \{j_{i-1}\}$ for $i \geq 2$ and therefore $\Omega_{\uparrow} = \{\omega(x) \mid x \in \{0, ..., d+m-1\}\}$ where $\omega(x)$ is such that $j_1 \in D$ and $j_i \in D \setminus \{j_{i-1}\}$ for $i \geq 2$ and $\nu_i(\omega(x)) = x\mathbb{I}_{\{i=1\}}$ with the number of nodes visited $k = \left\lfloor \frac{2d+m-2-x}{d} \right\rfloor$. Then the time of visits to nodes is given by

$$t_i(\omega(x)) = \begin{cases} 0 & \text{if } i = 1, \\ x + (i-1)d & \text{otherwise.} \end{cases}$$

and

$$n_i(\omega(x)) = \begin{cases} 0 & \text{if } i = 1, \\ \max(x + d - m + 1, 0) & \text{if } i = 2, \\ \max(x + (i-1)d - m + 1, l_i(\omega(x)) + 1, 0) & \text{otherwise,} \end{cases}$$

in which $l_i(\omega)$ is the last visit time to node j_i. Therefore the payoff for responding to $\boldsymbol{\phi}_{\text{poly}}$ with $\omega(x)$ is

$$P(\omega(x), \boldsymbol{\phi}_{\text{poly}}) = \sum_{t=0}^{\min(x,d-1)} \frac{1}{|D|d} + \sum_{t=\max(x+d-m+1,0)}^{\min(x+d,d-1)} \frac{1}{|D|d} \tag{A.1}$$

$$+ \sum_{i=3}^{k} \sum_{t=n_i(\omega(x))}^{\min(x+(i-1)d,d-1)} \frac{1}{|D|d}. \tag{A.2}$$

It is clear from equation (A.2) that the choice of the node sequence should be such that the last time a node is visited is maximized, therefore as all nodes in D are node-symmetric we can assume without loss of generality the node sequence is such that $j_i = v_{i \mod |D|+1}$. Then equation (A.2) becomes

$$P(\omega(x), \boldsymbol{\phi}_{\text{poly}}) = \sum_{t=0}^{\min(x,d-1)} \frac{1}{|D|d} + \sum_{i=2}^{|D|} \sum_{t=\max(x+(i-1)d-m+1,0)}^{d-1} \frac{1}{|D|d} \tag{A.3}$$

$$+ \sum_{t=\max(x+|D|d-m+1,x+1,0)}^{d-1} \frac{1}{|D|}. \tag{A.4}$$

Thus to find the performance of $\boldsymbol{\phi}_{\text{poly}}$ we seek to maximize the payoff in equation (A.4) by choosing x. It is clear from the equation that the choice of $x^* = d-1$ maximizes it and therefore the performance of $\boldsymbol{\phi}_{\text{poly}}$ is

$$V_{\bullet,\boldsymbol{\phi}_{\text{poly}}}(Q, d+m-1, m) = P(\omega(x^*), \boldsymbol{\phi}_{\text{poly}}) \tag{A.5}$$

$$= \sum_{t=0}^{d-1} \frac{1}{|D|d} + \sum_{i=2}^{|D|} \sum_{t=\max(id-m,0)}^{d-1} \frac{1}{|D|d} + 0 \tag{A.6}$$

$$= \frac{d}{|D|d} + \frac{\min(m-d, (|D|-1)d)}{|D|d} = \min\left(\frac{m}{|D|d}, 1\right). \tag{A.7}$$

Hence $V(Q,T,m) \leq \min\left(\frac{m}{|D|d}, 1\right)$ when $d \geq m$.

□

Appendix B

Star graphs

B.1 Proof of the adjusted combinatorial bound

For completeness we present the proof of lemma 4.2.13.

Proof. First consider the case of $m \in M_{5,0}^{S_n^k}$, with the adjusted combinatorial hybrid strategy $\boldsymbol{\pi}_{\text{AdjCombHyb}}$. The PIP is,

$$\begin{array}{ll} \text{maximize} & \min_{j \in N} \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, j}(S_n^k, T, m) p_i \\ \text{s.t} & p_i \in [0,1], i = 0,1,2, \\ & p_0 + p_1 + p_2 = 1. \end{array}$$

We can now simplify the objective function as we have either $j \in L$ or $j \in S$ (as $M = \emptyset$) and we know for the two sets that for any choice of p_1 and p_2,

- for all $j \in L$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, j}(S_n^k, T, m) p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, 1}(S_n^k, T, m) p_i$,
- for all $j \in S$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, j}(S_n^k, T, m) p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, *_1}(S_n^k, T, m) p_i$.

Moreover $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i, \bullet, *_k}(S_n^k, T, m) p_i$ is equal for any $k \in \{1, ..., n-1\}$ and so we need only consider the nodes 1 and $*_1$ in the PIP. Hence the PIP, along with the performances (in equations (4.15) and (4.17)) and reduction of $p_0 = 1 - p_1 - p_2$, becomes

$$\begin{array}{ll} \text{maximize} & \min \left(\frac{m}{2(n+k)}(1 - p_1 - p_2) + p_1, \frac{m}{2(n+k)}(1 - p_1 - p_2) + \frac{m}{2(n-1)} p_2 \right) \\ \text{s.t} & p_i \in [0,1], i = 1,2, \\ & p_1 + p_2 \leq 1. \end{array}$$

From the objective function of the PIP we know that it is maximized when $p_1 = \frac{m}{2(n-1)} p_2$ so we get the optimal solution that $p_1 = \frac{m}{m+2(n-1)}$ and $p_2 = \frac{2(n-1)}{m+2(n-1)}$

as $\frac{m}{2(n+k)} \leq \frac{1}{2}$ when $m \in M_{5,0}^{S_n^k}$. The optimal value gives the bound given in the lemma.

Similarly in the case of $m \in M_{5,1}^{S_n^k}$, we have a simplification of the objective function of the PIP as for any choice of p_1 and p_2,

- for all $j \in L$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,1}(S_n^k, T, m)p_i$,
- for all $j \in M$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i = \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,\hat{m}+2}(S_n^k, T, m)p_i = \frac{m}{n+k}p_0$,
- for all $j \in S$, $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,j}(S_n^k, T, m)p_i \geq \sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_1}(S_n^k, T, m)p_i$.

Moreover $\sum_{i=0}^{2} V_{\boldsymbol{\pi}_i,\bullet,*_k}(S_n^k, T, m)p_i$ is equal for any $k \in \{1, ..., n-1\}$ and so we need only consider the nodes 1, $\hat{m}+2$ and $*_1$ in the PIP. Hence the PIP, along with the performances (in equations (4.15) and (4.17)) and reduction of $p_0 = 1 - p_1 - p_2$, becomes

$$\begin{array}{ll} \text{maximize} & \min\left(\frac{m}{2(n+k)}p_0 + p_1, \frac{m}{n+k}p_0, \frac{m}{2(n+k)}p_0 + \frac{m}{2(n-1)}p_2\right) \\ \text{s.t} & p_i \in [0,1], i = 1, 2, \\ & p_1 + p_2 \leq 1. \end{array}$$

From the objective function of the PIP we know that it is maximized when $p_1 = \frac{m}{2(n-1)}p_2$, and $(1-(1+\frac{2(n-1)}{m})p_1)\frac{m}{2(n+k)} + p_1 = (1-(1+\frac{2(n-1)}{m})p_1)\frac{m}{n+k}$. Hence the optimal solution has $p_1 = \frac{m}{2(n+k)+m+2(n-1)}$, $p_2 = \frac{2(n-1)}{2(n+k)+m+2(n-1)}$ and $p_0 = \frac{2(n+k)}{m+n+k}$. The optimal value gives the bound as given in the lemma. $\square$

B.2 Proof of reduced time-centred attacker bound

For completeness we present the proof of lemma 4.2.15.

Proof. We aim to calculate $V_{\bullet,\phi_{\text{rtc}}}(S_n^k, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet,\phi_{\text{rtc}}}(S_n^k, T, m) = V_{\bullet,\phi_{\text{rtc}}}(S_n^k, 2m-1, m) = \max_{W \in \mathcal{W}(S_n^k, 2m-1, m)} P(W, \boldsymbol{\phi}_{\text{rtc}}),$$

so we only need to consider pure walks for $2m - 1$ units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet,\phi_{\text{rtc}}}(S_n^k, 2m-1, m) = \max_{\omega \in \Omega'''(S_n^k, 2m-1, m)} P(\omega, \boldsymbol{\phi}_{\text{rtc}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_n^k, 2m-1, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ such that the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$, where $N_A = \{1\} \cup \{*_l \mid l \in \{1, ..., n-1\}\}$,
- $\nu_1 \in \{\hat{m}, \hat{m}+1\}$ if $j_1 \in \{*_l \mid l \in \{1, ..., n-1\}\}$, $\nu_1 \in \{0, ..., m-1\}$ if $j_1 = 1$, $\nu_i = 0$ for all $i \in \{2, ..., q\}$ and
- $\nu_1 + \sum_{i=1}^{q-1}(d(j_i, j_{i+1}, N_A) + \nu_{i+1}) \equiv t_q + \nu_q = 2m-2.$

That is a move-wait walk such that nodes belong to those which have a non-zero probability of catching the attacker at if travelled to, with no waiting aside from at the initial node and that the arrival at the final node plus the final waiting match the end of the time-horizon.

For any such walk $\omega \in \Omega'''(S_n^k, 2m-1, m)$ the payoff is given by

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{\text{rtc}}) &= \sum_{i=1}^{q} \sum_{t} = n_i(\omega)^{t_i(\omega)+\nu_i} \varphi_{j_i,t} \\ &= \sum_{t=f(1)}^{\min(\nu_1, g(1))} \frac{1}{m+2(n-1)} + \sum_{i=2}^{q} \sum_{t=n'_i(\omega)}^{g(i)} \frac{1}{m+2(n-1)}, \end{aligned}$$

where $f(i) = \mathbb{I}_{\{j_i \neq 1\}}\hat{m}$, $g(i) = \mathbb{I}_{\{j_i=1\}}(m-1) + \mathbb{I}_{\{j_i \neq 1\}}(\hat{m}+1)$, $n'_i(\omega) = \max(f(i), l_i(\omega)+1, t_i(\omega)-m+1)$. By ignoring the last time a node is visited we get the following inequality

$$P(\omega, \boldsymbol{\phi}_{\text{rtc}}) \leq \sum_{t=f(1)}^{\min(\nu_1, g(1))} \frac{1}{m+2(n-1)} + \sum_{i=2}^{q} \sum_{t=n''_i(\omega)}^{g(i)} \frac{1}{m+2(n-1)}, \tag{B.1}$$

where $n''_i(\omega) = \max(f(i), t_i(\omega) - m + 1)$.

For any $i' \in \{2, ..., q\}$ such that $n''_{i'}(\omega) = t_{i'}(\omega) - m + 1$ we have for all $i \in \{i'+1, ..., q\}$ that $f(i) > n''_i(\omega)$ as $t_{i'+1}(\omega) = t_{i'}(\omega) + 2 + \mathbb{I}_{\{j_{i'}=1 \text{ or } j_{i'+1}=1\}}k > g(i'+1)$. Equation (B.1) therefore becomes

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{\text{rtc}}) \leq & \frac{\min(\nu_1 - f(1) + 1, g(1) - f(1) + 1)}{m+2(n-1)} + \frac{\sum_{i=2}^{i'-1}(g(i) - f(i) + 1)}{m+2(n-1)} \\ & + \frac{\max(g(i') - (t_{i'}(\omega) - m + 1) + 1, 0)}{m+2(n-1)} \end{aligned} \tag{B.2}$$

Equation (B.2) is clearly maximized by having $\nu_1 = g(1)$. In addition we know that any $i \in \{2, ..., l' - 1\}$ must be such that $j_i \neq 1$. Therefore equation (B.2) becomes

$$\begin{aligned}P(\omega, \boldsymbol{\phi}_{\text{rtc}}) \leq& \frac{g(1) - f(1) + 1 + 2(i' - 2)}{m + 2(n-1)} \\ &+ \frac{\max(g(i') + m - g(1) - 2(l' - 1) - \mathbb{I}_{\{j_1 = 1\}}k - \mathbb{I}_{\{j_{l'} = k\}}, 0)}{m + 2(n-1)} \\ =& \frac{g(1) - f(1) + 1 + \max(g(i') - g(1) + m - 2 - \mathbb{I}_{\{j_1 = 1\}}k - \mathbb{I}_{\{j_{l'} = k\}})}{m + 2(n-1)}.\end{aligned}$$

By considering if j_1 and $j_{l'}$ are node 1 we have

$$P(\omega, \boldsymbol{\phi}_{\text{rtc}}) \leq \frac{m}{m + 2(n-1)}.$$

Therefore

$$V_{\bullet, \boldsymbol{\phi}_{\text{rtc}}}(S_n^k, 2m - 1, m) \leq \frac{m}{m + 2(n-1)}. \tag{B.3}$$

The upper bound on the performance of $\boldsymbol{\phi}_{\text{rtc}}$, as in equation (B.3), gives

$$V(S_n^k, T, m) \leq \frac{m}{m + 2(n-1)}.$$

$\square$

B.3 Proof of the bespoke attacker bounds

B.3.1 ρ=3

For completeness we present the proof of lemma 4.2.19.

Proof. We aim to calculate $V_{\bullet, \boldsymbol{\phi}_{3-\text{stc}}}(S_n^k, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet, \boldsymbol{\phi}_{3-\text{stc}}}(S_n^k, T, m) = V_{\bullet, \boldsymbol{\phi}_{3-\text{stc}}}(S_n^k, 2m, m) = \max_{W \in \mathcal{W}(S_n^k, 2m, m)} P(W, \boldsymbol{\phi}_{3-\text{stc}}),$$

so we only need to consider pure walks for $2m$ units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet, \boldsymbol{\phi}_{3-\text{stc}}}(S_n^k, 2m, m) = \max_{\omega \in \Omega'''(S_n^k, 2m, m)} P(\omega, \boldsymbol{\phi}_{3-\text{stc}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_n^k, 2m, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ such that the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$, where $N_A = \{1\} \cup \{\frac{m+1}{2}+1\} \cup N_1 \cup N_2$ where $N_1 = \{*_l \mid l \in \{1, ..., n-1\}\}$ and $N_2 = \{k+1, k-1, ... \frac{m+1}{2}+2\}$,

- $$\nu_1 \in \begin{cases} \{0, 1, ..., m\} & \text{if } j_1 = 1, \\ \{\frac{m+1}{2} - 1\} & \text{if } j_1 = \frac{m+1}{2} + 1, \\ \{\frac{m+1}{2} - 1, \frac{m+1}{2}\} & \text{if } j_1 \in N_1 \cup N_2, \end{cases}$$

  and $\nu_i = 0$ for all $i \in \{2, ..., q\}$ and

- $\nu_1 + \sum_{i=1}^{q-1} (d(j_i, j_{i+1}, N_A) + \nu_{i+1}) \equiv t_q + \nu_q = 2m - 1.$

We will define the beginning of the 'single attacks' as $f(i) = (\frac{m+1}{2} - 1)\mathbb{I}_{\{j_i \neq 1\}}$ and the beginning of 'addition attacks' as

$$f'(i) = \begin{cases} f(i) & \text{if } j_i \in N_1 \cup N_2, \\ f(i) + 1 & \text{if } j_i = 1, \\ T & \text{if } j_i = \frac{m+1}{2} + 1, \end{cases}$$

where T is chosen to indicate that no additional attacks take place at the node $\frac{m+1}{2} + 1$. Similarly we define the end of 'single attacks' as

$$g(i) = \begin{cases} \frac{m+1}{2} & \text{if } j_i \in N_1 \cup N_2, \\ \frac{m+1}{2} - 1 & \text{if } j_i = \frac{m+1}{2} + 1, \\ m & \text{if } j_i = 1, \end{cases}$$

and the end of 'additional attacks' as

$$g'(i) = \begin{cases} g(i) & \text{if } j_i \in N_1 \cup N_2, \\ g(i) - 1 & \text{if } j_i = 1, \\ 0 & \text{if } j_i = \frac{m+1}{2} + 1, \end{cases}$$

where 0 is chosen to indicate that no additional attacks take place at the node $\frac{m+1}{2} + 1$.

Then for any such walk $\omega \in \Omega'''(S_n^k, 2m, m)$ the payoff is given by

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{3-\text{stc}}) &= \sum_{i=1}^{q} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_i,t} \\ &= \sum_{i=1}^{q} \left(\sum_{t=b_i(\omega)}^{\min(t_i+\nu_i, g(i))} \frac{1}{\hat{d}} + \sum_{t=b'_i(\omega)}^{\min(t_i+\nu_i, g'(i))} \frac{1}{\hat{d}} \right) \\ &= \left(\sum_{t=f(1)}^{\min(\nu_1, g(1))} \frac{1}{\hat{d}} + \sum_{t=f'(1)}^{\min(\nu_1, g'(1))} \frac{1}{\hat{d}} \right) \quad + \sum_{i=2}^{q} \left(\sum_{t=b_i(\omega)}^{g(i)} \frac{1}{\hat{d}} + \sum_{t=b'_i(\omega)}^{g'(i)} \frac{1}{\hat{d}} \right) \\ &\leq \left(\sum_{t=f(1)}^{\min(\nu_1, g(1))} \frac{1}{\hat{d}} + \sum_{t=f'(1)}^{\min(\nu_1, g'(1))} \frac{1}{\hat{d}} \right) + \sum_{i=2}^{q} \left(\sum_{t=c_i(\omega)}^{g(i)} \frac{1}{\hat{d}} + \sum_{t=c'_i(\omega)}^{g'(i)} \frac{1}{\hat{d}} \right) \end{aligned} \tag{B.4}$$

where $\hat{d} = 2(n+k)+m+2(n-1)$, $b_i(\omega) = \max(f(i), l_i(\omega)+1, t_i(\omega)-m+1)$ and $b'_i(\omega) = \max(f'(i), l_i(\omega)+1, t_i, t_i(\omega)-m+1)$ with $c_i(\omega) = \max(f(i), t_i(\omega)-m+1)$ and $c'_i(\omega) = \max(f'(i), t_i(\omega)-m+1)$ giving the inequality by ignoring the last visit time.

For any considered move wait walk ω, for the inequality in equation (B.4), if $j_{i'} \in N_2 \cup \{\frac{m+1}{2}+1\}$ for some $i' \in \{1, ..., q\}$ then ω' can be formed such that $j_i \neq 1$ for all $i \in \{i'+1, ..., q\}$ with $P(\omega', \boldsymbol{\phi}_{3-\text{stc}}) \geq P(\omega, \boldsymbol{\phi}_{3-\text{stc}})$. Moreover these j_i for $i \in \{i'+1, ..., q\}$ can be chosen such that they move towards the centre c before visiting nodes in N_1. In addition as visiting nodes in N_1 and N_2 are equivalent we now need only consider three possible walks by considering the starting positions.

For the first walk ω_1 we consider starting at node 1. So for the purpose of performance evaluation we have that

$$j_i \in \begin{cases} \{1\} & \text{if } i = 1, \\ \{\frac{m+1}{2}+1\} & \text{if } i = 2, \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

Note that we do not need to know the exact node sequence as all nodes bar node 1 and $\frac{m+1}{2}+1$ have the same $f(i)$,$f'(i)$,$g(i)$ and $g'(i)$. By equation (B.4) we have

$$\begin{aligned} P(\omega_1, \boldsymbol{\phi}_{3-\text{stc}}) &\leq \frac{\min(m+1, \nu_1+1) + \min(m-1, \nu_1)}{\hat{d}} \\ &\quad + \frac{\mathbb{I}_{\{\nu_1 \leq m-2\}} + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq m-(2+x)\}}}{\hat{d}} \\ &= \frac{\min(m+1, \nu_1+1) + \min(m-1, \nu_1)}{\hat{d}} \\ &\quad + \frac{\mathbb{I}_{\{\nu_1 \leq m-2\}} + 2\mathbb{I}_{\{\nu_1 \leq m-2\}}(m-1-\nu_1)}{\hat{d}} \\ &= \begin{cases} \frac{2m}{\hat{d}} & \text{if } \nu_1 = m, \\ \frac{2m-1}{\hat{d}} & \text{if } \nu_1 = m-1, \\ \frac{2m}{\hat{d}} & \text{if } \nu_1 \leq m-2, \end{cases} \leq \frac{2m}{\hat{d}}. \end{aligned}$$

For the second walk ω_2 we consider starting at node $\frac{m+1}{2}+1$. Note that in this case $\nu_1 = \frac{m+1}{2} - 1$ and that

$$j_i \in \begin{cases} \{\frac{m+1}{2}+1\} & \text{if } i = 1 \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

By equation (B.4) we have

$$\begin{aligned} P(\omega_2, \boldsymbol{\phi}_{3-\text{stc}}) &\leq \frac{1+4+\sum_{x=0}^{\infty} 2\mathbb{I}_{\{m \geq (3+x)\}}}{\hat{d}} = \frac{5+2\mathbb{I}_{\{m \geq 3\}}(m-2)}{\hat{d}} \\ &= \frac{2m+1}{\hat{d}}. \end{aligned}$$

For the third and final walk ω_3 we consider starting at any node in $N_1 \cup N_2$ and therefore $j_i \in N_1 \cup N_2$ for all $i \in \{1, ..., q\}$, with a choice for ν_1 of $\frac{m+1}{2} - 1$ or $\frac{m+1}{2}$. By equation (B.4) we have

$$\begin{aligned} P(\omega_3, \boldsymbol{\phi}_{3-\mathrm{stc}}) &\leq \frac{2\min(2, 2+\nu_1 - \frac{m+1}{2}) + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq \frac{m+1}{2}+m-(3+x)\}}}{\hat{d}} \\ &= \frac{2\min(2, 2+\nu_1 - \frac{m+1}{2}) + 2\mathbb{I}_{\{\nu_1 \leq \frac{m+1}{2}+m-3\}}(\frac{m+1}{2} + m - 2 - \nu_1)}{\hat{d}} \\ &= \frac{2m}{\hat{d}}. \end{aligned}$$

Therefore in any case we have that $P(\omega, \boldsymbol{\phi}_{3-\mathrm{stc}}) \leq \frac{2m+1}{\hat{d}}$ and so we have

$$V_{\bullet, \boldsymbol{\phi}_{3-\mathrm{stc}}}(S_n^k, 2m, m) \leq \frac{2m+1}{\hat{d}} = \frac{2m+1}{2(n+k)+m+2(n-1)}. \tag{B.5}$$

The upper bound on the performance of $\boldsymbol{\phi}_{3-\mathrm{stc}}$, as in equation (B.5), gives

$$V(S_n^k, T, m) \leq \frac{2m+1}{2(n+k)+m+2(n-1)}.$$

□

B.3.2 ρ=2

For completeness we present the proof of lemma 4.2.21.

Proof. We aim to calculate $V_{\bullet, \boldsymbol{\phi}_{2-\mathrm{stc}}}(S_n^k, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet, \boldsymbol{\phi}_{2-\mathrm{stc}}}(S_n^k, T, m) = V_{\bullet, \boldsymbol{\phi}_{2-\mathrm{stc}}}(S_n^k, 2m-3, m) = \max_{W \in \mathcal{W}(S_n^k, 2m-3, m)} P(W, \boldsymbol{\phi}_{2-\mathrm{stc}}),$$

so we only need to consider pure walks for $2m - 3$ units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet, \boldsymbol{\phi}_{2-\mathrm{stc}}}(S_n^k, 2m-3, m) = \max_{\omega \in \Omega'''(S_n^k, 2m-3, m)} P(\omega, \boldsymbol{\phi}_{2-\mathrm{stc}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_n^k, 2m-3, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ such that the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$, where $N_A = \{1\} \cup \{\frac{m+4}{2}\} \cup N_1 \cup N_2$ where $N_1 = \{*_l \mid l \in \{1, ..., n-1\}\}$ and $N_2 = \{k+1, k-1, ..., \frac{m+4}{2}+1\}$,
- $$\nu_1 \in \begin{cases} \{0, 1, ..., m-2\} & \text{if } j_1 = 1, \\ \{\frac{m-2}{2}\} & \text{if } j_1 = \frac{m+4}{2}, \\ \{\frac{m-4}{2}, \frac{m-2}{2}, \frac{m}{2}\} & \text{if } j_1 \in N_1 \cup N_2, \end{cases}$$

  and $\nu_i = 0$ for all $i \in \{2, ..., q\}$ and
- $\nu_1 + \sum_{i=1}^{q-1} (d(j_i, j_{i+1}, N_A) + \nu_{i+1}) \equiv t_q + \nu_q = 2m - 4.$

Then for any such walk $\omega \in \Omega'''(S_n^k, 2m-3, m)$ the payoff is given by

$$P(\omega, \boldsymbol{\phi}_{1-\text{stc}}) = \sum_{i=1}^{q} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_i,t} = \sum_{i=1}^{q} S(i), \tag{B.6}$$

where the function $S(i)$ is given by for $i = 2$

$$S(1) = \begin{cases} \sum_{t=0}^{\min(\nu_1, m-2)} \frac{2\mathbb{I}_{\{t=0 \mod 2\}}}{\hat{d}} & \text{if } j_1 = 1, \\ 1 & \text{if } j_1 = \frac{m+4}{2}, \\ 1 + \mathbb{I}_{\{\nu_1 = \frac{m}{2}\}} & \text{if } j_1 \in N_1 \cup N_2, \end{cases}$$

and for $i \geq 2$

$$S(i) = \begin{cases} \sum_{t=\max(0, l_i(\omega)+1, t_i(\omega)-m+1)}^{m-2} \frac{2\mathbb{I}_{\{t=0 \mod 2\}}}{\hat{d}} & \text{if } j_i = 1, \\ \mathbb{I}_{\{\max(l_i(\omega)+1, t_i(\omega)-m+1) \leq \frac{m-2}{2}\}} & \text{if } j_i = \frac{m+4}{2}, \\ \mathbb{I}_{\{\max(l_i(\omega)+1, t_i(\omega)-m+1) \leq \frac{m}{2}\}} + \mathbb{I}_{\{\max(l_i(\omega)+1, t_i(\omega)-m+1) \leq \frac{m-4}{2}\}} & \text{if } j_i \in N_1 \cup N_2, \end{cases}$$

where the denominator $\hat{d} = n + k + \frac{m}{2} + n - 1$.

For any considered move wait walk ω, for the inequality in equation (B.6), if $j_{i'} \neq 1$ for some $i' \in \{1, ..., q\}$ then ω' can be formed such that $j_i \neq 1$ for all $i \in \{i'+1, ..., q\}$ with $P(\omega', \boldsymbol{\phi}_{2-\text{stc}}) \geq P(\omega, \boldsymbol{\phi}_{2-\text{stc}})$. Moreover these j_i for $i \in \{i'+1, ..., q\}$ can be chosen such that they move towards the centre c before visiting nodes in N_1. In addition as visiting nodes in N_1 and N_2 are equivalent we now need only consider three possible walks by considering the starting positions.

For the first walk ω_1 we consider starting at node 1. So for the purpose of performance evaluation we have that

$$j_i \in \begin{cases} \{1\} & \text{if } i = 1, \\ \{\frac{m+4}{2}\} & \text{if } i = 2, \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

Note that we do not need to know the exact node sequence as all nodes bar nodes $1, \frac{m+4}{2}$, as $S(i)$ is the same function for all $i \geq 2$ when $j_i \in N_1 \cup N_2$. By equation (B.6) we have

$$\begin{aligned} P(\omega_1, \boldsymbol{\phi}_{1-\mathrm{stc}}) &\leq \frac{\min(2(\lfloor \frac{\nu_1}{2} \rfloor + 1), m) + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq m-(3+2x)\}}}{\hat{d}} \\ &= \frac{\min(2(\lfloor \frac{\nu_1}{2} \rfloor + 1), m) + 2\mathbb{I}_{\{\nu_1 \leq m-3\}}(\lfloor \frac{m-3-\nu_1}{2} \rfloor + 1)}{\hat{d}} \\ &= \frac{m}{\hat{d}}. \end{aligned}$$

For the second walk ω_2 we consider starting at node $\frac{m+4}{2}$ and so $\nu_1 = \frac{m-2}{2}$ and

$$j_i \in \begin{cases} \{\frac{m+4}{2}\} & \text{if } i = 1, \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

By equation (B.6) we have

$$\begin{aligned} P(\omega_2, \boldsymbol{\phi}_{1-\mathrm{stc}}) &\leq \frac{1 + 1 + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{m \geq (3+2x)\}}}{\hat{d}} = \frac{2 + 2\mathbb{I}_{\{m \geq 3\}}(\lfloor \frac{m-3}{2} \rfloor + 1)}{\hat{d}} \\ &= \frac{2 + 2(\frac{m-4}{2} + 1)}{\hat{d}} = \frac{m}{\hat{d}}. \end{aligned}$$

For the third and final walk ω_3 we consider starting at any node in $N_1 \cup N_2$ and therefore $j_i \in N_1 \cup N_2$ for all $i \in \{1, ..., q\}$, with a choice for ν_1 of $\frac{m-4}{2}, \frac{m-2}{2}, \frac{m}{2}$. By equation (B.6) we have

$$\begin{aligned} P(\omega_3, \boldsymbol{\phi}_{2-\mathrm{stc}}) &\leq \frac{\min\left(\left\lfloor \frac{\nu_1 - \frac{m-4}{2}}{2} \right\rfloor + 1, 2\right) + 1 + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq \frac{3m}{2} - (5+2x)\}}}{\hat{d}} \\ &= \frac{\min\left(\left\lfloor \frac{\nu_1 - \frac{m-4}{2}}{2} \right\rfloor + 1, 2\right) + 1 + 2\mathbb{I}_{\{\nu_1 \leq \frac{3m}{2} - 5\}}(\left\lfloor \frac{\frac{3m}{2} - 5 - \nu_1}{2} \right\rfloor + 1)}{\hat{d}} \\ &= \begin{cases} \frac{m}{\hat{d}} & \text{if } \nu_1 = \frac{m-4}{2}, \\ \frac{m}{\hat{d}} & \text{if } \nu_1 = \frac{m-2}{2}, \\ \frac{m-1}{\hat{d}} & \text{if } \nu_1 = \frac{m}{2}, \end{cases} \leq \frac{m}{\hat{d}}. \end{aligned}$$

Therefore in any case we have that $P(\omega, \boldsymbol{\phi}_{2-\mathrm{stc}}) \leq \frac{m}{\hat{d}}$ and so we have

$$V_{\bullet, \boldsymbol{\phi}_{2-\mathrm{stc}}}(S_n^k, 2m, m) \leq \frac{m}{\hat{d}} = \frac{2m}{2(n+k) + m + 2(n-1)}. \tag{B.7}$$

The upper bound on the performance of $\boldsymbol{\phi}_{2-\mathrm{stc}}$, as in equation (B.7), gives

$$V(S_n^k, T, m) \leq \frac{2m}{2(n+k) + m + 2(n-1)}.$$

□

B.3.3 ρ=1

For completeness we present the proof of lemma 4.2.23.

Proof. We aim to calculate $V_{\bullet,\phi_{1-\mathrm{stc}}}(S_n^k, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet,\phi_{1-\mathrm{stc}}}(S_n^k, T, m) = V_{\bullet,\phi_{1-\mathrm{stc}}}(S_n^k, 2m, m) = \max_{W \in \mathcal{W}(S_n^k, 2m, m)} P(W, \phi_{1-\mathrm{stc}}),$$

so we only need to consider pure walks for $2m$ units of time.

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet,\phi_{1-\mathrm{stc}}}(S_n^k, 2m, m) = \max_{\omega \in \Omega'''(S_n^k, 2m, m)} P(\omega, \phi_{1-\mathrm{stc}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_n^k, 2m, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ such that the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$, where $N_A = \{1\} \cup \{\frac{m+3}{2}, \frac{m+3}{2} + 1\} \cup N_1 \cup N_2$ where $N_1 = \{*_l \mid l \in \{1, ..., n-1\}\}$ and $N_2 = \{k+1, k-1, ... \frac{m+3}{2} + 2\}$,
- $$\nu_1 \in \begin{cases} \{0, 1, ..., m\} & \text{if } j_1 = 1, \\ \{\frac{m-1}{2}, \frac{m-1}{2} + 1\} & \text{if } j_1 = \frac{m+3}{2}, \\ \{\frac{m-1}{2}\} & \text{if } j_1 = \frac{m+3}{2} + 1, \\ \{\frac{m-1}{2}, \frac{m-1}{2} + 1\} & \text{if } j_1 \in N_1 \cup N_2, \end{cases}$$
  and $\nu_i = 0$ for all $i \in \{2, ..., q\}$ and
- $\nu_1 + \sum_{i=1}^{q-1} (d(j_i, j_{i+1}, N_A) + \nu_{i+1}) \equiv t_q + \nu_q = 2m - 1.$

We will define the beginning of the 'single attacks' as $f(i) = (\frac{m-1}{2})\mathbb{I}_{\{j_i \neq 1\}}$ and the beginning of 'addition attacks' as

$$f'(i) = \begin{cases} f(i) & \text{if } j_i \in N_1 \cup N_2, \\ f(i) + 1 & \text{if } j_i = 1, \\ T & \text{if } j_i \in \{\frac{m+3}{2}, \frac{m+3}{2} + 1\}, \end{cases}$$

where T is chosen to indicate that no additional attacks take place. Similarly we define the end of 'single attacks' as

$$g(i) = \begin{cases} \frac{m-1}{2} + 1 & \text{if } j_i \in N_1 \cup N_2, \\ \frac{m-1}{2} & \text{if } j_i = \frac{m+1}{2} + 1, \\ m & \text{if } j_i = 1, \end{cases}$$

and the end of 'additional attacks' as

$$g'(i) = \begin{cases} g(i) & \text{if } j_i \in N_1 \cup N_2, \\ g(i) - 1 & \text{if } j_i = 1, \\ 0 & \text{if } j_i \in \{\frac{m+3}{2}, \frac{m+3}{2} + 1\}, \end{cases}$$

where 0 is chosen to indicate that no additional attacks take place.

Then for any such walk $\omega \in \Omega'''(S_n^k, 2m, m)$ the payoff is given by

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{1-\text{stc}}) &= \sum_{i=1}^{q} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_i,t} \\ &= \sum_{i=1}^{q} \left(\sum_{t=b_i(\omega)}^{\min(t_i+\nu_i, g(i))} \frac{1}{\hat{d}} + \sum_{t=b'_i(\omega)}^{\min(t_i+\nu_i, g'(i))} \frac{1}{\hat{d}} \right) \\ &= \left(\sum_{t=f(1)}^{\min(\nu_1, g(1))} \frac{1}{\hat{d}} + \sum_{t=f'(1)}^{\min(\nu_1, g'(1))} \frac{1}{\hat{d}} \right) + \sum_{i=2}^{q} \left(\sum_{t=b_i(\omega)}^{g(i)} \frac{1}{\hat{d}} + \sum_{t=b'_i(\omega)}^{g'(i)} \frac{1}{\hat{d}} \right) \\ &\leq \left(\sum_{t=f(1)}^{\min(\nu_1, g(1))} \frac{1}{\hat{d}} + \sum_{t=f'(1)}^{\min(\nu_1, g'(1))} \frac{1}{\hat{d}} \right) + \sum_{i=2}^{q} \left(\sum_{t=c_i(\omega)}^{g(i)} \frac{1}{\hat{d}} + \sum_{t=c'_i(\omega)}^{g'(i)} \frac{1}{\hat{d}} \right) \end{aligned} \tag{B.8}$$

where $\hat{d} = 2(n+k)+m+2(n-1)$, $b_i(\omega) = \max(f(i), l_i(\omega)+1, t_i(\omega)-m+1)$ and $b'_i(\omega) = \max(f'(i), l_i(\omega)+1, t_i, t_i(\omega)-m+1)$ with $c_i(\omega) = \max(f(i), t_i(\omega)-m+1)$ and $c'_i(\omega) = \max(f'(i), t_i(\omega) - m + 1)$ giving the inequality by ignoring the last visit time.

For any considered move wait walk ω, for the inequality in equation (B.8), if $j_{i'} \neq 1$ for some $i' \in \{1, ..., q\}$ then ω' can be formed such that $j_i \neq 1$ for all $i \in \{i' + 1, ..., q\}$ with $P(\omega', \boldsymbol{\phi}_{1-\text{stc}}) \geq P(\omega, \boldsymbol{\phi}_{1-\text{stc}})$. Moreover these j_i for $i \in \{i' + 1, ..., q\}$ can be chosen such that they move towards the centre c before visiting nodes in N_1. In addition as visiting nodes in N_1 and N_2 are equivalent we now need only consider four possible walks by considering the starting positions.

For the first walk ω_1 we consider starting at node 1. So for the purpose of performance evaluation we have that

$$j_i \in \begin{cases} \{1\} & \text{if } i = 1, \\ \{\frac{m+3}{2}\} & \text{if } i = 2, \\ \{\frac{m+3}{2} + 1\} & \text{if } i = 3, \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

Note that we do not need to know the exact node sequence as all nodes bar nodes

$1, \frac{m+3}{2}, \frac{m+3}{2}+1$ have the same $f(i)$,$f'(i)$,$g(i)$ and $g'(i)$. By equation (B.8) we have

$$\begin{aligned}
P(\omega_1, \boldsymbol{\phi}_{1-\text{stc}}) &\leq \frac{\min(m+1, \nu_1+1)+\min(m-1, \nu_1)}{\hat{d}} \\
&\quad + \frac{\mathbb{I}_{\{\nu_1 \leq m-1\}} + \mathbb{I}_{\{\nu_1 \leq m-2\}} + 3\mathbb{I}_{\{\nu_1 \leq m-3\}} + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq m-(4+x)\}}}{\hat{d}} \\
&= \frac{\min(m+1, \nu_1+1)+\min(m-1, \nu_1)}{\hat{d}} \\
&\quad + \frac{\mathbb{I}_{\{\nu_1 \leq m-1\}} + \mathbb{I}_{\{\nu_1 \leq m-2\}} + 3\mathbb{I}_{\{\nu_1 \leq m-3\}} + 2\mathbb{I}_{\{\nu_1 \leq m-4\}}(m-3-\nu_1)}{\hat{d}} \\
&= \begin{cases} \frac{2m}{\hat{d}} & \text{if } \nu_1 \in \{m, m-1\} \\ \frac{2m-1}{\hat{d}} & \text{if } \nu_1 = m-2, \\ \frac{2m}{\hat{d}} & \text{if } \nu_1 \leq m-3 \end{cases} \leq \frac{2m}{\hat{d}}.
\end{aligned}$$

For the second walk ω_2 we consider starting at node $\frac{m+3}{2}$. Note that in this case $\nu_1 \in \{\frac{m-1}{2}, \frac{m-1}{2}+1\}$ and that

$$j_i \in \begin{cases} \{\frac{m+3}{2}\} & \text{if } i=1, \\ \{\frac{m+3}{2}+1\} & \text{if } i=2, \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

By equation (B.8) we have

$$\begin{aligned}
P(\omega_2, \boldsymbol{\phi}_{1-\text{stc}}) &\leq \frac{\min(2, \nu_1+1-\frac{m-1}{2})+1+\sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq \frac{m-1}{2}+m-(2+x)\}}}{\hat{d}} \\
&= \frac{\min(2, \nu_1+1-\frac{m-1}{2})+1+2\mathbb{I}_{\{\nu_1 \leq \frac{m-1}{2}+m-2\}}(\frac{m-1}{2}+m-1-\nu_1)}{\hat{d}} \\
&= \frac{2m}{\hat{d}}.
\end{aligned}$$

For the third walk ω_3 we consider starting at node $\frac{m+3}{2}+1$.Note that in this case $\nu_1 = \frac{m-1}{2}$ and that

$$j_i \in \begin{cases} \{\frac{m+3}{2}+1\} & \text{if } i=1, \\ N_1 \cup N_2 & \text{otherwise.} \end{cases}$$

By equation (B.8) we have

$$\begin{aligned}
P(\omega_3, \boldsymbol{\phi}_{1-\text{stc}}) &\leq \frac{1+4+\sum_{x=0}^{\infty} 2\mathbb{I}_{\{m \geq (3+x)\}}}{\hat{d}} \\
&= \frac{5+2\mathbb{I}_{\{m \geq 3\}}(m-2)}{\hat{d}} = \frac{2m+1}{\hat{d}}.
\end{aligned}$$

For the fourth and final walk ω_4 we consider starting at any node in $N_1 \cup N_2$ and therefore $j_i \in N_1 \cup N_2$ for all $i \in \{1, ..., q\}$, with a choice for ν_4 of $\frac{m-1}{2}$ or $\frac{m-1}{2}+1$. By equation (B.8) we have

$$\begin{aligned} P(\omega_4, \boldsymbol{\phi}_{1-\mathrm{stc}}) &\leq \frac{2\min(2, \nu_1 + 1 - \frac{m-1}{2}) + \sum_{x=0}^{\infty} 2\mathbb{I}_{\{\nu_1 \leq \frac{m-1}{2} + m - (2+x)\}}}{\hat{d}} \\ &= \frac{2\min(2, \nu_1 + 1 - \frac{m-1}{2}) + 2\mathbb{I}_{\{\nu_1 \leq \frac{m-1}{2} + m - 2\}}(\frac{m-1}{2} + m - 1 - \nu_1)}{\hat{d}} \\ &= \frac{2m}{\hat{d}}. \end{aligned}$$

Therefore in any case we have that $P(\omega, \boldsymbol{\phi}_{1-\mathrm{stc}}) \leq \frac{2m+1}{\hat{d}}$ and so we have

$$V_{\bullet, \boldsymbol{\phi}_{1-\mathrm{stc}}}(S_n^k, 2m, m) \leq \frac{2m+1}{\hat{d}} = \frac{2m+1}{2(n+k) + m + 2(n-1)}. \tag{B.9}$$

The upper bound on the performance of $\boldsymbol{\phi}_{3-\mathrm{stc}}$, as in equation (B.9), gives

$$V(S_n^k, T, m) \leq \frac{2m+1}{2(n+k) + m + 2(n-1)}.$$

□

B.4 Proof of time-spread attacker strategy

For completeness we present the proof of lemma 4.4.3.

Proof. We aim to calculate $V_{\bullet, \boldsymbol{\phi}_{\mathrm{ts}}}(S_{n_1,n_2}, T, m)$, to use this performance as an upper bound, and by lemma 3.2.14 we can restrict the game length for such a calculation, as the lemma gives us that

$$V_{\bullet, \boldsymbol{\phi}_{\mathrm{ts}}}(S_{n_1,n_2}, T, m) = V_{\bullet, \boldsymbol{\phi}_{\mathrm{ts}}}(S_{n_1,n_2}, 2n_1 + m + 1, m) = \max_{W \in \mathcal{W}(S_{n_1,n_2}, 2n_1 + m + 1, m)} P(W, \boldsymbol{\phi}_{\mathrm{ts}}).$$

Furthermore by theorem 3.2.13 we have that

$$V_{\bullet, \boldsymbol{\phi}_{\mathrm{ts}}}(S_{n_1,n_2}, 2n_1 + m + 1, m) = \max_{\omega \in \Omega'''(S_{n_1,n_2}, 2n_1 + m + 1, m)} P(\omega, \boldsymbol{\phi}_{\mathrm{ts}}),$$

and so we need only consider move-wait walks $\omega \in \Omega'''(S_{n_1,n_2}, 2n_1 + m + 1, m)$ such that

$$\omega = ((j_1, \nu_1), ..., (j_q, \nu_q)),$$

for some $q \in \mathbb{N}$ where the following three conditions are met

- $j_1 \in N_A$, $j_i \in N_A(\omega, i-1)$ for all $i \in \{2, ..., q\}$ where the set of attacked nodes $N_A = N_1 \cup N_2$ in which $N_1 = \{*_{r,1} \mid r \in \{1, ..., n_1\}\}$ and $N_2 = \{*_{r,2} \mid r \in \{1, ..., n_2\}\}$,
- $\nu_i \in \{f(i), ..., t_i(\omega) - (2n_2 + 1 - f(i))\}$ for all $i \in \{1, ..., q\}$ where $f(i) = \mathbb{I}_{\{j_i \in N_1\}}$,
- $\nu_1 + \sum_{i=1}^{q-1}(d(j_i, j_{i+1}) + \nu_{i+1}, N_A) \equiv t_q + \nu_q = 2n_1 + m$.

The payoff for ω against $\boldsymbol{\phi}_{\text{ts}}$ is given by

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{\text{ts}}) &= \sum_{i=1}^{q} \sum_{t=n_i(\omega)}^{t_i(\omega)+\nu_i} \varphi_{j_i,t} \\ &= \sum_{i=1}^{q} \sum_{t=n_i'(\omega)}^{\min(2n_2+1-f(i), t_i(\omega)+\nu_i)} \frac{1}{\hat{d}}, \end{aligned} \tag{B.10}$$

where $n_i(\omega) = \max(0, l_i(\omega)+1, t_i(\omega)-m+1)$, $n_i' = \max(f(i), l_i(\omega)+1, t_i(\omega)-m+1)$ and $\hat{d} = 2n_2(n_1 + n_2 + 1)$.

Consider ω_1 such that $j_1 \in N_1$ then we can replace this initial node with $j_1 \in N_2$ to create ω_2 and we have $P(\omega_2, \boldsymbol{\phi}_{\text{ts}}) \geq P(\omega_1, \boldsymbol{\phi}_{\text{ts}})$.

Therefore we can consider only ω such that $j_1 \in N_2$, now let us assume without loss of generality that nodes in either set, N_1 or N_2, are visited in ascending index order, then $j_1 = *_{1,2}$. Furthermore for any ω such that $j_i \in N_1$ for any $2 \leq i \leq n_2$ we can form ω' by replacing these j_i with nodes in N_2 which does not lower the payoff as $P(\omega', \boldsymbol{\phi}_{\text{ts}}) \geq P(\omega, \boldsymbol{\phi}_{\text{ts}})$. Thus we need only ω such that $j_i = *_{i,2}$ for $1 \leq i \leq n_2$.

Now given ω such that $j_i \in N_2$ for $n_2+1 \leq i \leq n_1+n_2$ we can replace these with $j_i \in N_1$ creating ω' which does not lower the payoff as $P(\omega', \boldsymbol{\phi}_{\text{ts}}) \geq P(\omega, \boldsymbol{\phi}_{\text{ts}})$. Then given that we need only consider ω such that $j_i = *_{i,2}$ for $1 \leq i \leq n_2$ and $j_i = *_{i-n_2,1}$ for $n_2+1 \leq i \leq n_1+n_2$ we know the only nodes left in $N_A(\omega, n_1+n_2) = N_2$ and so these are returned to in ascending order.

Therefore we only need to consider ω such that

$$j_i = \begin{cases} *_{i,2} & \text{for } 1 \leq n_2, \\ *_{i-n_2,1} & \text{for } n_2 + 1 \leq n_1 + n_2, \\ *_{i-(n_1+n_2),2} & \text{for } n_1 + n_2 + 1 \leq n_1 + 2n_2, \end{cases}$$

and $\nu_i \in \{0, ..., 2n_2 + 1 - \sum_{i'=1}^{i-1}(2 + \nu_{i'})\}$ for $1 \leq i \leq n_2$ and $\nu_i = 0$ for $i \geq n_2 + 1$.

With such restrictions we get that equation (B.10) becomes

$$P(\omega, \boldsymbol{\phi}_{\text{ts}}) = \frac{\min(\nu_1 + 1, 2n_1 + 1)}{\hat{d}} + \frac{\sum_{i=2}^{n_2} \min(B_1(i) - C_1(i) + 1, 0)}{\hat{d}} + \frac{\sum_{i=n_2+1}^{n_1+n_2} \min(2n_1 + 1 - C_2(i), 0)}{\hat{d}} + \frac{\sum_{i=n_1+n_2+1}^{n_1+2n_2} \min(2n_2 + 2 - C_3(i))}{\hat{d}}, \tag{B.11}$$

where $B_1(i) = \min(2n_2 + 1, 2(i-1) + \sum_{i'=1}^{i} \nu_{i'})$, $C_1(i) = \max(0, 2(i-1) - m + 1 + \sum_{i'=1}^{i-1} \nu_{i'})$, $C_2(i) = C_1(i) + 1$ and $C_3(i) = \max(2(i - n_1 - n_2 - 1) + \nu_{i-n_1-n_2} + 1 + \sum_{i'=1}^{i-n_1-n_2-1} \nu_{i'}, 2i - m + 1 + \sum_{i'=1}^{i-1} \nu_{i'})$.

From equation (B.11), with the limitations on the waiting times, we can see that we need only consider $\nu_1 \neq 0$ and $\nu_i = 0$ for all $i \neq 1$. Further to this we see that $\nu_1 = 0$ then maximizes the equation. Thus for $m = n_2 + x$ for some $x \in \mathbb{N}$ when x is even we have

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{\text{ts}}) &= \frac{\sum_{i=1}^{n_2}(2i-1)}{\hat{d}} + \frac{\frac{x}{2} \times 2n_2 + \sum_{i=1}^{n_2} 2(n_2 - i)}{\hat{d}} \\ &= \frac{n_2(n_2+1)}{\hat{d}} + \frac{xn_2 + n_2(n_2-1)}{\hat{d}} = \frac{n_2(n_2+x)}{\hat{d}}, \end{aligned}$$

and when x is odd we have

$$\begin{aligned} P(\omega, \boldsymbol{\phi}_{\text{ts}}) &= \frac{\sum_{i=1}^{n_2}(2i-1)}{\hat{d}} + \frac{\frac{x-1}{2} \times 2n_2 + \sum_{i=1}^{n_2} (2(n_2 - i) + 1)}{\hat{d}} \\ &= \frac{n_2(n_2+1)}{\hat{d}} + \frac{(x-1)n_2 + n_2^2}{\hat{d}} = \frac{n_2(n_2+x)}{\hat{d}}. \end{aligned}$$

So we have

$$P(\omega, \boldsymbol{\phi}_{\text{ts}}) = \frac{mn_2}{\hat{d}} = \frac{m}{2(n_1+n_2+1)}.$$

Therefore

$$V_{\bullet, \boldsymbol{\phi}_{\text{ts}}}(S_{n_1,n_2}, 2n_1 + m + 1, m) = \frac{m}{2(n_1+n_2+1)}. \tag{B.12}$$

The performance of $\boldsymbol{\phi}_{\text{ts}}$, as in equation (B.12), gives the upper bound on the value of the game

$$V(S_{n_1,n_2}, T, m) \leq \frac{m}{2(n_1+n_2+1)}.$$

$\square$

Appendix C

Extensions

C.1 Proof of changing distances on superfluous edges

For completeness we present the proof of lemma 6.1.4 which uses the same idea as that of the proof for lemma 6.1.3.

Proof. As $\boldsymbol{\pi}^*$ does not use any edge $e \in F$ and as $D(e) = D'(e)$ for all $e \in E \setminus F$ we have that $\boldsymbol{\pi}^* \in \Pi(Q, D', T, m)$ and hence by using $\boldsymbol{\pi}^*$ in the game $G = (Q, D', T, m)$ we have

$$\begin{aligned} V(Q, D', T, m) \geq V_{\boldsymbol{\pi}^*, \bullet}(Q, D', T, m) &= \max_{a \in \mathcal{A}(Q, D', T, m)} P(\boldsymbol{\pi}^*, a) \\ &= \max_{a \in \mathcal{A}(Q, D, T, m)} P(\boldsymbol{\pi}^*, a) = V(Q, D, T, m). \end{aligned} \tag{C.1}$$

By using lemma 6.1.1 to compare the game $G(Q, D', T, m)$ to the game $G(Q, D, T, m)$ we get

$$V(Q, D', T, m) \leq V(Q, D, T, m). \tag{C.2}$$

The lower bound in equation (C.1) and the upper bound in equation (C.2) are equal and thus $V(Q, D', T, m) = V(Q, D, T, m)$. $\square$

C.2 Proof of random spread Hamiltonian cycle

We present an alternative proof for theorem 6.3.6. We do so by proving the lower bound for the patroller using the random spread Hamiltonian strategy in a similar fashion to the original way the random Hamiltonian was proved in [16]. For ease of notation we can relabel the nodes of $Q = (N, E) \in \mathcal{H}$ to be $N = \{1, ..., n\}$ such that $H(t) = t + 1$ for all $t \in \{0, ..., n - 1\}$. We therefore prove,

$$V(k, 1, Q, T, m) \geq \min\left(1, \frac{km}{n}\right)$$

Proof. For any pure attack (j, τ) which has the attack interval $I = \{\tau, ..., \tau + m - 1\}$, each pure scheduler strategy $\boldsymbol{W}$, played with non-zero probability in $\boldsymbol{s}_{\text{rsH}}$, is such that for each patroller $r \in \{1, ..., k\}$ their implemented walk W_r is such that

$$\bigcup_{r=1}^{k} W_r(I) = \{j \mod n \mid j = i, ..., i + km\},$$

for some $i \in \{0, ..., n-1\}$ as each subsequent patroller is placed m ahead of the previous. Hence if $km \geq n$ there are no missing nodes for any i and hence we get a value of 1.

Thus for each starting position of patroller 1 ($i = 1, ..., n$), the number nodes in $\bigcup_{r=1}^{k} W_r(I)$ form an km-arc. Hence as the starting position of patroller 1 is chosen with equal probability from all nodes, the chance of any attack being caught is,

$$\frac{km}{n}.$$

Hence we obtain the lower bound required for tightness with lemma 6.3.3. □