

Patrolling games on graphs collection

Thomas Lowbridge
School of Mathematical Sciences
University of Nottingham

December 6, 2017

This document houses all work on patrolling games on graphs.

Contents

1	Introduction to game	1
1.1	Pure game	1
1.2	Mixed game	2
1.3	Properties	2
2	Current Reduction Tools	2
2.1	Symmetry	2
2.2	Strategy Domination	3
3	Bounds	4
3.1	Basic Bounds(Attacker and Patroller)	4
3.2	Decomposition(Patroller)	4
3.3	Simplification(Patroller and Attacker)	5
3.4	Old Diametric attack(Attacker)	6
3.5	Covering(Patroller) and Independence(Attacker)	6
4	Simple Graph bounds and solutions	7
4.1	Hamiltonian	7
4.2	Biparite	8
4.3	Complete Bipartite	8
5	Line graph	9
6	New Bounds	9
6.1	Problem and Correction of diametric attack	9
6.2	Extension of diametric attack	11
6.3	Extension of Hamiltonian patrols	11
7	Elongated star graph	13
7.1	Introduction	13
7.2	Basic Analysis	13
7.3	Solution for $m \geq 2(k + 1)$	15
7.4	Solutions below Hamiltonian boundary	16

7.5	Improving the Hamiltonian bound	16
7.6	Solution for $m = 2k + 1$ and $m = 2k$	20
7.7	Solution for $m = 2k - 1$ and $m = 2k - 2$	22
7.8	Solution for certain m step-points	23
7.9	Various Stage-step points	24
7.10	Lower Line type attacks	26
7.11	Extending even solution	28
7.12	Combinatorial loop improvement	28
7.13	Odd solution	31
7.14	Solution for $m \leq 2(n - 1)$	34
7.15	Solution when $m > 2(n - 1)$	34
8	General star graph	35
8.1	Introduction	35
8.2	Basic Analysis	35
8.3	solution for $m \geq 2(k_{max} + 1)$	36
9	Dual star graphs	38
10	Linked Star graphs	38
	Appendices	i
A	Graph Definitions	i
B	Old Statements about Old diametric	iii
C	Improvemening Hamiltonian bound	iv
C.1	Reason For Probability of Interception formula	iv
C.2	Naive Improvement Complete analysis	v
C.3	Combinatorial Improvement Complete analysis	vi
C.4	Combinatorial loop improvement analysis	vii
D	Proof's	vii
D.1	Alternating Random Hamiltonian proof	vii
D.2	Condition on T for old diametric proof	viii
D.3	Time-limited diametric attack proof	ix
D.4	Polygonal attack proof	x
D.5	Time-delayed attack proof	xi
D.6	Type-delayed attack proof	xxiii
D.7	Stage-1-Step-point lemma proof	xxviii

1 Introduction to game

1.1 Pure game

A patrolling game $G = G(Q, T, m)$ is a win-lose, zero-sum game between a maximizing patroller (often referred to as she) and a minimizing attacker (often referred to as he). The parameters of the game are:

- The graph Q which has a set of nodes, N , and of edges, E .
- The length of time over which the game takes place, the time-horizon T .
- The length of time the attack takes to complete, the attack-time m .

Two forms of the game exist: the one-off game, which is played in a finite time interval $\mathcal{T} = \{0, 1, \dots, T-1\}$ denoted using G^o ; and the periodic game, which is played on the time circle $\mathcal{T}^* = \{0, 1, \dots, T-1\}$ (with the asterisk representing arithmetic on the time circle taking place modulo T) denoted using G^p . We will assume that $T \geq m$, otherwise all attacks will fail.

The pure strategies available to the patroller are called patrols, choosing a starting position and how to walk along the graph Q , $W : \mathcal{T} \rightarrow N$. With no restrictions in the one-off game, but the restriction that the edge $(W(T-1), W(0)) \in E$ in the periodic game (so that $W(T) = W(0)$). Let

$$\mathcal{W} = \{W \mid W : \mathcal{T} \rightarrow N \text{ s.t. } (W(t), W(t+1)) \in E \text{ for } t = 0, \dots, T-2\}$$

be the set of all pure patrols in the one-off game (and similarly \mathcal{W}^* in the periodic game). Let there be some ordering to the strategies $W_k \in \mathcal{W}$ (or $W_k \in \mathcal{W}^*$) for $k = 1, \dots, |\mathcal{W}|$ (or $k = 1, \dots, |\mathcal{W}^*|$ in the periodic game).

The pure strategies available to the attacker are pairs, $[i, I]$ for $i \in N$, called the attack node, and $I = \{\tau, \tau+1, \dots, \tau+m-1\} \subseteq \mathcal{T}$ (or $I \subseteq \mathcal{T}^*$ if periodic) called the attack interval (starting at time τ). Let $\mathcal{A} = \{[i, I] \mid i \in N, I \subseteq \mathcal{T}\}$ be the set of all possible pure attacks. Let there be some ordering to the strategies $A_k \in \mathcal{A}$ for $k = 1, \dots, |\mathcal{A}|$.

A patrol, W , intercepts the attack, $[i, I]$, if $i \in W(I) = \{W(\tau), W(\tau+1), \dots, W(\tau+m-1)\}$ and as our game is Win-Lose the pure payoff function is

$$P(W, [i, I]) = \begin{cases} 1 & \text{if } i \in W(I), \\ 0 & \text{if } i \notin W(I). \end{cases}$$

A pure payoff matrix $\mathcal{P} = (P(W_i, A_j))_{i \in \{1, \dots, |\mathcal{W}|\}, j \in \{1, \dots, |\mathcal{A}|\}}$ (with the change of \mathcal{W} to \mathcal{W}^* if in the periodic game) stores Win (1) or Lose (0) for each pair of pure strategies.

1.2 Mixed game

Let Π_W be the set of mixed strategies for the patroller in the one-off game and Π_W^* in the periodic game. Let Φ be the set of mixed strategies for the attacker.

In the mixed strategy game the patroller selects a strategy $\pi \in \Pi_W$ (or $\pi \in \Pi_W^*$ in the periodic game).

The attacker selects a strategy $\phi \in \Phi$. Then the mixed payoff function (Probability of Capture) is

$$P(\pi, \phi) = \sum_{i=1}^{|\mathcal{W}|} \sum_{j=1}^{|\mathcal{I}|} \mathcal{P}_{i,j} \pi_i \phi_j = \pi \mathcal{P} \phi$$

(with the change of \mathcal{W} to \mathcal{W}^* if in the periodic game).

We will also use the convention that a pure strategy is in the mixed strategy set, $W_i \in \Pi_W$ (or $W_i \in \Pi_W^*$) and $A_j \in \Phi$, to mean $\pi_k = \begin{cases} 1 & \text{if } k = i, \\ 0 & \text{if } k \neq i. \end{cases}$ and $\phi_k = \begin{cases} 1 & \text{if } k = j, \\ 0 & \text{if } k \neq j. \end{cases}$ respectively

The value of the game is denoted by $V = V(Q, T, m) \equiv \max_{\pi \in \Pi} \min_{\phi \in \Phi} P(\pi, \phi) = \min_{\phi \in \Phi} \max_{\pi \in \Pi} P(\pi, \phi)$ and when needed we distinguish between the one-off and period game by using the subscripts V^o and V^p respectively.

1.3 Properties

2 Current Reduction Tools

2.1 Symmetry

Symmetry of a graph's nodes allows us to reduce the number of attacks and patrols under consideration. This is because symmetric nodes must be treated identically (or some bias is formed for the other player). More formally

Definition 2.1 (Attacker Equivalence). We call two nodes, n_1 and n_2 , *equivalent* if there exists some automorphism, σ , of Q which takes one to the other, i.e $n_1 = \sigma(n_2)$ or $n_2 = \sigma(n_1)$. Two attack intervals, I_1 and I_2 , are called *equivalent* if there exists some automorphism, γ on the time interval, \mathcal{T} which is reflective, i.e $\gamma(t) = T - t$.

Lemma 2.2 (Attacker Symmetry). *We only need to consider the equivalence class of nodes for the attacker, with the same attack pattern formed on the equivalent nodes. Similarly we need only consider the equivalence class of attack*

times. Furthermore the periodic game has all attack times equivalent (under some rotation of the time circle) so only the node need be considered.

Definition 2.3 (Patroller Equivalence). We call two patrols, $W_1(t)$ and $W_2(t)$, equivalent if there exists some automorphism, σ , of Q takes one two the other, i.e $W_1(t) = \sigma(W_2(t))$ or $W_2(t) = \sigma(W_1(t))$.

Lemma 2.4 (Patroller Symmetry). We only need to consider the equivalence class of patrols for the patroller, with the same patrol pattern formed on the equivalent patrols.

So without affecting the value of the graph we can reduce the problem to one of searching through the equivalent class of strategies for both players.

2.2 Strategy Domination

Domination of some strategies for the patroller and attacker are given, which allows the removal of these strategies from the possible strategy space for the players.

Note. We use the term dominance to mean weak dominance.

Lemma 2.5 (Waiting Patrol Dominated). For Q connected and $T \geq 3$, $m \geq 2$, patrols staying at any node for three consecutive periods are dominated.

Corollary 2.6 (Penultimate Attack Dominated). For $m \geq 3$, attacks at penultimate (i.e non-leaf node adjacent to a leaf node) nodes are dominated.

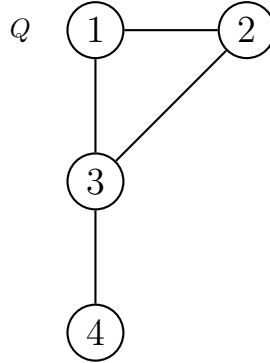


Figure 2.1: Graph Q used for examples

Example 2.7. For Q as seen Figure 2.1 We can see that nodes, 1 and 2, are equivalent nodes by using the automorphism $\sigma(1) = 2$, $\sigma(2) = 1$, $\sigma(i) = i \quad \forall i \neq 1, 2$. Similarly the patrols $W_1(t) = \begin{cases} 1 & \text{for } t \text{ even} \\ 3 & \text{for } t \text{ odd} \end{cases}$ and $W_2(t) = \begin{cases} 2 & \text{for } t \text{ even} \\ 3 & \text{for } t \text{ odd} \end{cases}$ are equivalent under the same automorphism.

The patrol $W_3(t) = \begin{cases} 1 & \text{for } t = 1, 2, 3 \\ 3 & \text{for } t \geq 4 \end{cases}$ is dominated (by $W_1(t)$)

Node 3 is penultimate and so will not be attacked.

3 Bounds

3.1 Basic Bounds(Attacker and Patroller)

We seek bounds on the value of the game V . The lower bounds are given in terms of the patroller's "good" strategy against all attacker options, similarly the upper bounds are given in terms of the attacker's "good" strategy against all the patroller options. When we reach tightness between the bound these "good" strategies become an optimal solution.

By the patroller waiting a random node they can achieve $V \geq \frac{1}{n}$ and by the attacker picking a random node with a fixed time I they can achieve $V \leq \frac{m}{n}$ (More generally $V \leq \frac{\omega}{n}$ for ω , the maximum number of nodes any patrol can cover).

Lemma 3.1 (General bounds).

$$\frac{1}{n} \leq V \leq \frac{\omega}{n} \leq \frac{m}{n}$$

Where ω is the maximum number of distinct nodes that can be visited in an attack interval.

3.2 Decomposition(Patroller)

We can consider decomposing the graph so that we just operate on parts with some appropriate probability.

Lemma 3.2 (Decomposition lower bound). *Consider decomposing Q into edge-preserving subgraphs Q_i for $i = 1, \dots, k$ with values $V_i = V(Q_i)$ such that $Q = \bigcup_{i=1}^k Q_i$ then*

$$V \geq \frac{1}{\sum_{i=1}^k \frac{1}{V_i}}$$

Furthermore in the case of a disjoint decomposition equality is reached

More explicitly an edge-preserving subgraph is a subgraph who has all possible connection between its nodes and disjoint means both edge and vertex disjoint. This means we are really only selecting nodes for the subgraph and the edges are mandated. The above provides a solution to build disjointly decomposable graphs, so it is only worth studying connected graphs.

Example 3.3. For Q as seen in Figure 2.1. Consider when $m = 3$, the decomposition of Q into the graphs Q_1 and Q_2 (as in Example Figure 3.1).

$V_1 = V(Q_1) = 1$ as alternating between 1 and 2 can catch every attack.
 $V_2 = V(L_3) = \frac{3}{4}$ (as seen in [?]).

Then we can get the bound $V \geq \frac{1}{(\frac{1}{1} + \frac{3}{4})} = \frac{4}{7}$.

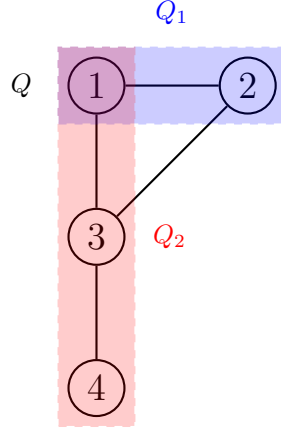


Figure 3.1: Decomposition of Q into Q_1 and Q_2 .

3.3 Simplification(Patroller and Attacker)

Definition 3.4 (Node Identification). The operation of Node identification on two nodes, u and v , of a graph, $G = (N, E)$ into a single node w , is a mapping $f : N \rightarrow N'$ resulting in a new graph $G' = (N', E')$ where $N' = (N \setminus \{u, v\}) \cup \{w\}$ with $E' = E \setminus \{(u, v)\}$ if $(u, v) \in E$ and under the condition that $\forall x \in N$, $f(x) \in N'$ is incident to $e' \in E'$ iff $e \in E$ is incident to $x \in N$. Furthermore if a graph, Q , undergoes repeated node identification to become Q' then we say it has been simplified.

Definition 3.5 (Embedded walk). An *Embedded walk*, W' , on Q' is the walk, W , done on Q under the simplification mapping of Q to Q' . i.e if $\pi : Q \rightarrow Q'$ is the simplification map, then $W' = \pi(W)$.

Lemma 3.6 (Simplification). *If Q' is a simplified version of Q then $V(Q') \geq V(Q)$*

This allows us to get bounds for both the patroller and attacker.

Example 3.7. For Q as seen in Figure 2.1. Consider when $m = 3$, the Simplification of the graph by identifying 1, 2 from Q to $Q' = L_3$ (as seen in Example Figure 3.2). Hence we can get the bound that $V(L_3) \geq V(Q)$ hence $V(Q) \leq \frac{3}{4}$

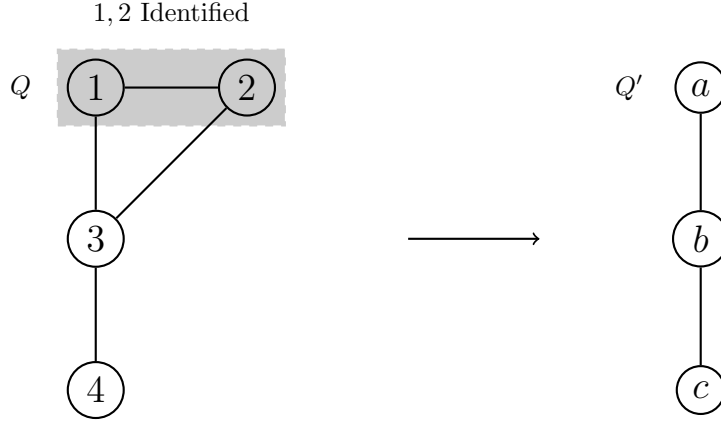


Figure 3.2: Simplification of Q to Q' by identification.

3.4 Old Diametric attack(Attacker)

Let $d(i, i')$ is the distance between nodes i and i' with the distance measured by the minimum number of edges.

Definition 3.8 (Graph Diameter). The diameter of a graph Q is defined by $\bar{d} = \max_{i, i' \in N} d(i, i')$. The node pairs satisfying this are called diametrical.

Lemma 3.9. *By the attacker playing equally likely at a pair of diametrical at a random time interval, called the diametric attack, gives $V \leq \max \left\{ \frac{m}{2\bar{d}}, \frac{1}{2} \right\}$*

However the bound presented in [?] seems to indicate for large T, m the second is chosen. However a simple counter example will show the bound does not allow hold

Example 3.10 (Problem with diametric attack). Consider the graph L_5 so $\bar{d} = 4$ for $T = m = 5$, then under the diametric attack, the patroller performing the patrol $\{1, 2, 3, 4, 5\}$ allows her to catch the two attacks. Hence the bound of $V \leq \frac{5}{8}$ given by lemma does not seem to hold.

3.5 Covering(Patroller) and Independence(Attacker)

Definition 3.11 (Covering). A patrol, W , is called *intercepting* if it intercepts every possible attack at every node contained in the patrol, i.e all nodes visited by W are in any subpath of length m (i.e visits are at most m apart).

A set of intercepting patrols forms a *Covering set* if every node in Q is contained in at least one of the patrols. Furthermore the *Covering number*, \mathcal{C} is the minimum cardinality of all the covering sets.

Definition 3.12 (Independence). Two nodes, i and i' , are called independent (under attack time, m) if any patrol intercepting an attack at i cannot also intercept an attack at i' .

For the one-off game this is equivalent to $d(i, i') \geq m$.

For the Periodic game this is equivalent to $d(i, i') \geq m$ and $2d(i, i') \leq T$ (due to returning to start).

A set of independent points forms a *Independent set* if every element of the set is independent of every other element. Furthermore the independence number \mathcal{I} is the maximum cardinality of all the independent sets.

Clearly $\mathcal{I} \leq \mathcal{C}$ as to cover a collection of independent nodes, at least that many covering patrols are needed (Possibly more if they also don't get every node in Q)

Lemma 3.13 (Covering and Independence).

$$\frac{1}{\mathcal{C}} \leq V \leq \frac{1}{\mathcal{I}}$$

4 Simple Graph bounds and solutions

4.1 Hamiltonian

A Hamiltonian graph is a graph with a Hamiltonian cycle. Two simple examples of such graphs are cyclic graphs, C_n and the complete graph, K_n . While Hamiltonian graphs can exhibit more than one Hamiltonian cycle we shall assume that we have selected one. We shall also assume that the attack, $m < n$, as otherwise by following the Hamiltonian cycle we guarantee capture (i.e for $m \geq n$, $V = 1$).

Definition 4.1 (Random Hamiltonian Patrol). A *Random Hamiltonian Patrol* is a mixed strategy starting with equal probability at all nodes and following the Hamiltonian cycle.

Theorem 4.2 (Hamiltonian). *If Q is Hamiltonian, by following the Random Hamiltonian Patrol (if feasible), the patroller can achieve $V \geq \frac{m}{n}$.*

This, along with a general upper bound from [?], provides the solution $V = \frac{m}{n}$.

Two common Hamiltonian graphs are the Cyclic graph (of n nodes C_n) and the Complete graph (of n nodes K_n).

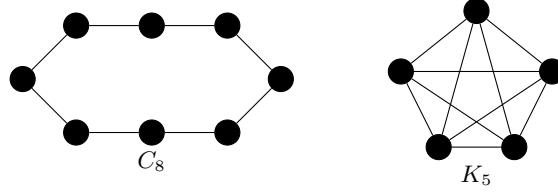


Figure 4.1: Examples of Cyclic and Complete graphs

4.2 Bipartite

A bipartite graph is a graph which can be partitioned into two sets, A and B (with $|A| = a, |B| = b$, assume WLOG that $b \geq a$) where edges only exist between these two sets. A special bipartite graph is the complete bipartite graph $K_{a,b}$.

Assume that $m < 2b$, as otherwise there exists a $2b$ period patrol which covers all nodes and guarantees capture (i.e if $m \geq 2b$, $V = 1$).

Definition 4.3 (Bipartite Attack). The *Bipartite Attack* selects nodes equiprobably from the larger set B for a fixed time interval, I (or for the two time intervals, I and $I + 1$ equiprobably).

Theorem 4.4 (Bipartite). *If Q is bipartite with $b \geq a$, by following the Bipartite Attack, the attacker can achieve $V \leq \frac{m}{2b}$*

The reasoning behind the bound is that any patrol must alternate between $|A|$ and $|B|$, so only visits a node from B every other time step.

4.3 Complete Bipartite

Corollary 4.5 (Complete Bipartite). *The value of the complete bipartite graph, $K_{a,b}$, with $b \geq a$, then $V = \frac{m}{2b}$.*

This is because a lower bound of $V \geq \frac{m}{2b}$ is given by the random Hamiltonian patrol in $K_{b,b}$, which simplifies to $K_{a,b}$.

The star graph, S_n is a graph with a centre and n nodes adjacent to the centre.

Corollary 4.6 (Star). *The value of $S_n \equiv K_{1,n}$ is $V = \frac{m}{2n}$.*

5 Line graph

6 New Bounds

6.1 Problem and Correction of diametric attack

Consider the patroller strategy against a diametric attack that simply oscillates between the two diametric points.

The total number of attacks the attacker is making under this diametric strategy is $2(T - m + 1)$, we will now measure how many the simple strategy for the patroller gets.

We will divide the set of captured attacks, depending on what is happening. This division shall be into start captures, middle captures and end captures.

The start captures are captures catching less than m attacks in the early times, i.e before the middle. The middle captures are captures catching exactly m attacks. The end captures are captures catching less than m attacks in the late times.

Example 6.1 (Problem with diametric attack). Consider the graph L_5 so $\bar{d} = 4$ for $T = 20, m = 6$, then under the diametric attack, the patroller oscillating between diametrics points gets.

Start Capture $1 + 5 = 6$ attacks initially.

Middle Capture $6 + 6 = 12$ attacks when arriving at node 5.

End Capture 4 attacks when finishing at node 1.

Giving 22 out of $2(20 - 6 + 1) = 30$ attacks, a better than $\frac{3}{4}$ value.

First off it is worth considering that the number of end attacks that are going to be caught will come from two values (if more than these would be in the middle). We could suggest that waiting at the start is more preferable to “stabilize” into the middle quicker. the cost for doing so is to remove one from each end value, if one of the end values is 1, then the penultimate middle is also reduced by 1 and thus becomes an end value.

While each time we decide to wait gains us 1 for each node in the start values, until it becomes a middle value. As $m \geq \bar{d}$, we are guaranteed that there are at least two start values (as we are looking at times 0 and $\bar{d} - 1 < m$). Therefore it is certain that waiting at the start (at least while there are two start values) is not worse than the just oscillating strategy.

Therefore we can wait until $t = m - (\bar{d} - 1)$ (which as $m > \bar{d} - 1$ means its is always possible), then this is only the start.

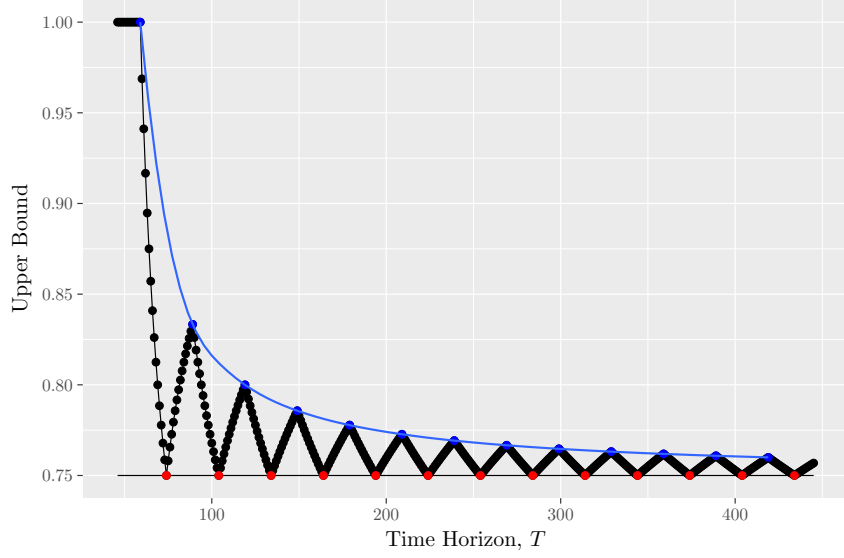


Figure 6.1: Best Upper Bound achievable under the diametric strategy

Let us count the pattern under this strategy,

Start: Capture $m - \bar{d}$ attacks initially by waiting.

Middle: Capture $m \times (\lfloor \frac{T-2m+1}{\bar{d}} \rfloor + 1)$ attacks in the middle cycles (if any middle times are possible, otherwise zero if negative).

End: Capture $T - 1 - (m - 1 + (\lfloor \frac{T-2m+1}{\bar{d}} \rfloor + 1)\bar{d})$ (at the penultimate node visit (if possible, this really is zero if its negative) and $T - 1 - (m - 1 + (\lfloor \frac{T-2m+1}{\bar{d}} \rfloor + 2)\bar{d})$ at the final node visit (again zero if negative).

This gives

$$m - \bar{d} + \left(m \times \left(\left\lfloor \frac{T - 2m + 1}{\bar{d}} \right\rfloor + 1 \right) \right)_+ + \\ \left(T - (m - 1 + (\left\lfloor \frac{T - 2m + 1}{\bar{d}} \right\rfloor + 1)\bar{d}) \right)_+ + \left(T - (m - 1 + (\left\lfloor \frac{T - 2m + 1}{\bar{d}} \right\rfloor + 2)\bar{d}) \right)_+$$

We will call $\alpha = \lfloor \frac{T-2m+1}{\bar{d}} \rfloor$

Lemma 6.2 (Condition on T for bound to hold). *When $T = m - 1 + (k + 1)\bar{d}$ for some $k \in \mathbb{N}_0$ then the diametric bound holds. Otherwise as $T \rightarrow \infty$ then the diametric bound holds.*

Proof: D.2

Fixing the diametric attack

A possible “fix” to the problem of the excess time is to limit the diametric attacks window in which attacks are placed.

Definition 6.3 (Time-limited diametric attack). Attacking at a pair of diametric nodes equiprobably for the times $I, I+1, \dots, I+\bar{d}-1$ (i.e starting attacks at $\tau, \tau+1, \dots, \tau+\bar{d}-1$) is called the *timed diametric attack*.

Note. The time-limited diametric attack is only feasible is $T \geq m + \bar{d} - 1$.

Lemma 6.4. When $T \geq m + \bar{d} - 1$, the diametric bound $V \leq \max\{\frac{1}{2}, \frac{m}{2\bar{d}}\}$ is valid.

Proof: D.3

6.2 Extension of diametric attack

The idea of the diametric attack, that of attacking points far away from each other can be extended. First of all it is not necessary to use the graph diameter, though on a pair of point this provides the strongest bound.

Consider attacking the points i and i' with $d = d(i, i')$. Then the timed “diametric” attack gives the bound $V \leq \max\{\frac{1}{2}, \frac{m}{2d}\}$.

While if only using a pair of nodes this is pointless as it worse than the standard use of a diametric pair. However consider a set of points $D = \{i \in N \mid d(i, i') = d\}$, this is a set of nodes mutually a distance d from each other. Then it is possible to use the same type of timed attack to get a bound

Definition 6.5 (Polygonal attack). A *d-polygonal attack* is an attack at a set of nodes $D = \{i \in N \mid d(i, i') = d\}$ at the time intervals $I, I+1, \dots, I+d-1$ (for a chosen initial I)

Note. Just like the time-limited diametric attack, the polygonal attack is only feasible if $T \geq m + d - 1$.

The idea behind this attack is very similar to the timed diametric attack.

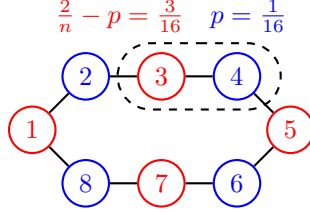
Lemma 6.6. When $T \geq m + d - 1$ and a set D as in the *d-polygonal attack*, the bound $V \leq \max\{\frac{1}{|D|}, \frac{m}{|D|d}\}$ is valid

Proof: D.4

6.3 Extension of Hamiltonian patrols

We seek to find a larger class of optimal patrols for hamiltonian graphs, by using groups of nodes.

Definition 6.7 (Alternating Random Hamiltonian Patrol(ARHP)). An *Alternating Random Hamiltonian Patrol (ARHP)* is a mixed strategy following the Hamiltonian cycle but with a probability p of starting at “even” nodes and a probability of $\frac{2}{n} - p$ of starting at “odd” nodes.



Example Figure 6.1: C_8 with the blue nodes being “even” nodes started at with probability $\frac{1}{16}$ and the red nodes being “odd” nodes started at with probability $\frac{3}{16}$.

Lemma 6.8. When n and m are both even, following the Alternating Random Hamiltonian Patrol, if feasible, gives the same lower bound as the random Hamiltonian patrol, i.e $V \geq \frac{m}{n}$.

Proof: D.1

If m is odd, say $m = 2m' + 1$ then in the above we get two possibilities for each node depending on the interval choice either $p + \frac{m-1}{n}$ or $\frac{m+1}{n} - p$. So choosing anything other than $p = \frac{1}{n}$ (which is the Random Hamiltonian Patrol strategy) gives a worse result for the patroller.

While not getting a better lower bound, the ARHP does give some control on how to perform optimally in a Hamiltonian graph. The idea of distributing the probability $\frac{2}{n}$ between two types of nodes can be extended to the idea of distributing the probability $\frac{k}{n}$ between k types of nodes.

As an extension to the Alternating Random Hamiltonian Patrol(ARHP) is the Block Random Hamiltonian Patrol(BRHP).

Definition 6.9 (Block Random Hamiltonian Patrol(BRHP)). A k type *Block Random Hamiltonian Patrol (BRHP)* is a mixed strategy following the Hamiltonian Patrol but with a probability p_i of starting at a “type i ” node for $i = 1, \dots, k$, where $\sum_{i=1}^k p_i = \frac{k}{n}$.

Lemma 6.10. When n and m are both multiples of k , then following the k type Block Random Hamiltonian Patrol, if feasible, gives the same lower bound as the random Hamiltonian patrol, i.e $V \geq \frac{m}{n}$

Proof. During any attack interval I , which is of length $m = km'$, then $W(I)$ contains m' “type i ” nodes for $i = 1, \dots, k$. Therefore by following the k type BRHP, π_{BRHP} , with probability p_i at “type i ” for $i = 1, \dots, k$. Then

$$P(\pi_{BRHP}, [i, I]) \geq \underbrace{\overbrace{p_1}^{\text{type 1}} + \dots + \overbrace{p_k}^{\text{type } k} + p_1 + \dots + p_n + \dots + p_1 + \dots + p_k}_{m=km' \text{ elements}} = 2m'$$

$$m'p_1 + \dots + m'p_k = m' \frac{k}{n} = \frac{m}{n} \quad \forall i \in N \quad \forall I \subseteq \mathcal{T}$$

Hence as it holds for all pure attacks

$$P(\pi_{BRHP}, \phi) \geq \frac{m}{n} \quad \forall \phi \in \Phi$$

Hence $V \geq \frac{m}{n}$. □

Note. In the one-off game, the BRHP strategy is always feasible. However in the periodic game, the BRHP strategy is only feasible if $T = k'n$ for some $k' \in \mathbb{N}$.

7 Elongated star graph

7.1 Introduction

The line and star graphs provided a good starting point for attempting to solving the problem with a general tree. If a more general version of the star graph can be solved, it may provide better bounds on a general tree and may in fact provide an exact solution (under certain conditions).

The idea is to extend the star graph to a more general graph which is a mix between the line and the star, by extending the length of the branches (at first just one branch). This may better model a tightly packed region to search and another that is far away, consider the example of a small town and larger city connected by a road.

Definition 7.1 (Elongated Star Graph). The *Elongated Star Graph*, S_n^k is made from S_n , by performing subdivision on one of the edges repeatedly k times, so that one of the external nodes is now $k + 1$ away from the centre.

The labelling will be done as in Example Figure 7.1 and we will from now assume that $n \geq 3$, as otherwise we are just dealing with the line.

7.2 Basic Analysis

To start our analysis of this graph, we can look at an expanded graph which can simplify down to our extended star graph. Consider the cyclic graph $C_{(2k+1)+(2n-1)} = C_{2(n+k)}$, we can simplify this graph by node identifying. The identifying map is one such that we identify nodes i to $2k + 2 - i$ for $i = 1, \dots, k$ and identify nodes $2k + 2, 2k + 4, \dots, 2k + 2n$.

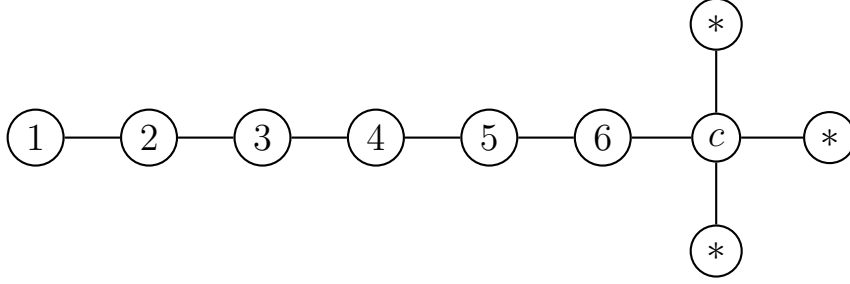


Figure 7.1: Labeling on the graph S_4^5 .

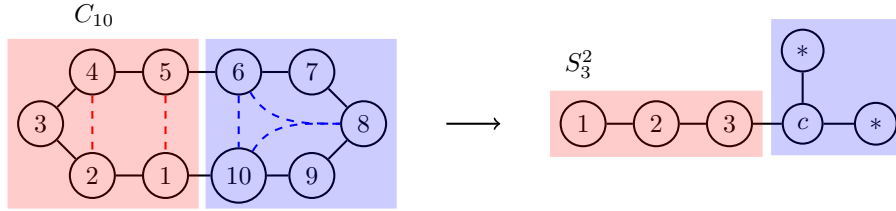


Figure 7.2: C_{10} can be simplified to S_3^2 by node identifying.

Definition 7.2 (Random Oscillation). The *Oscillation* on S_n^k is any embedded Hamiltonian Patrol on $C_{2(n+k)}$ under the simplification above. The *Random Oscillation* on S_n^k is the embedded Random Hamiltonian Patrol on $C_{2(n+k)}$ under the simplification above.

Lemma 7.3. For $m < 2(n+k)$ following the *Random Oscillation*,

$$V(S_n^k) \geq V(C_{2(n+k)}) = \frac{m}{2(n+k)}$$

and if $m \geq 2(n+k)$ then $V(S_n^k) = 1$, achieved by any *Oscillation*.

Hence we have the solution in $m \geq 2(n+k)$, so we can now restrict ourselves to $m < 2(n+k)$.

Next we look at getting an attacker bound. In S_n , the use of the equal weighting at each external nodes is similar to the diametric attack proposed in [?] but now picks the set of all points mutually at diameter. A natural analogue of this is to use some weighting depending on the length they are away from the centre.

Definition 7.4 (Weighted External Attack). The *Weighted External Attack* attacks at node 1 with probability $\frac{k+1}{n+k}$ and at each $*$ node with probability $\frac{1}{n+k}$ for a random attack interval I .

The idea of the weighted external attack is that we can form the upper bound $V \leq \frac{m}{2(n+k)}$. However, unlike first thought, it does not seem to give the upper

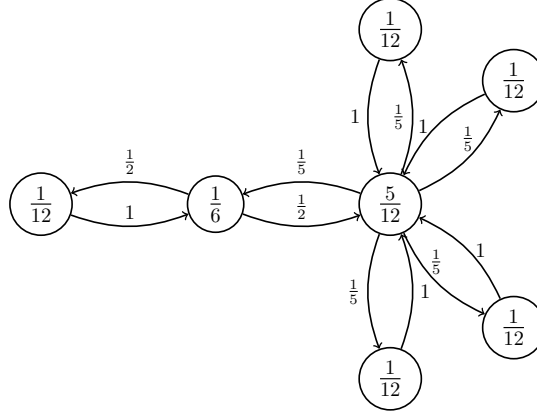


Figure 7.3: Starting points and initial movements of the Random Oscillation on S_5^1 .

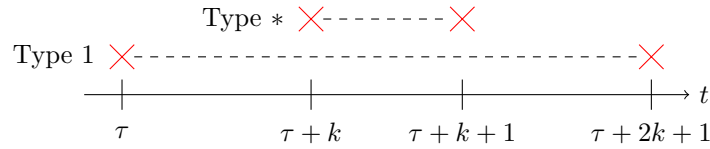
bound as we wished, the problem is clear that for a small time horizon there is an issue of being able to achieve a better patrolling strategy (as seen in example ??).

7.3 Solution for $m \geq 2(k+1)$

We will match the Oscillation bound of $\frac{m}{2(n+k)}$, by developing a further extended version of time-limited attack.

Definition 7.5 (Time-delayed attack). Let the *time-delayed attack*, be the attack that attacks at the extended node labeled 1 with probability $\frac{k+1}{n+k}$ and a particular normal node labeled $*$ with probability $\frac{1}{n+k}$.

If node 1 is chosen have the attack choose probability intervals with equal probability at the times $I, I+1, \dots, I+2k+1$ for some I (i.e starting attacks at $\tau, \tau+1, \dots, \tau+2k+1$). If a $*$ node is chosen start the attacks at the times $I+k, I+k+1$ with equal probability.



Note. The time-delayed attack uses the weighted suggested previously but delays the normal star ends starting times (and has them only occur for a shorter period of time)

Lemma 7.6. When $T \geq m+2k$, the analogous ‘diametric’ bound $V \leq \max\{\frac{k+1}{n+k}, \frac{m}{2(n+k)}\}$ is valid

Proof: ??

With this result we have the solution in our range

Lemma 7.7 (Solution in $m \geq 2(k+1)$). *By the attacker using the stage-delayed attack and the patroller using a random oscillation patrol, we achieve the value, when $2(k+1) \leq m \leq 2(n+k)$*

$$V = \frac{m}{2(n+k)}$$

Hence we have a solution for 2 regions $m > 2(n+k)$ and $2(k+1) \leq m \leq 2(n+k)$

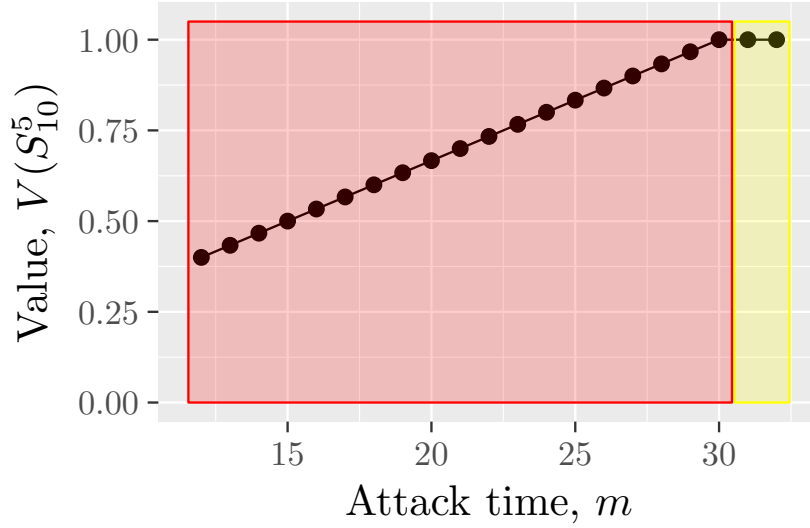


Figure 7.4: Value of the Star Graph, S_{10}^5

7.4 Solutions below Hamiltonian boundary

We now seek solutions below the Hamiltonian bound of value $\frac{m}{2(n+k)}$. In this region like in the line graph regions below oscillation boundary, we expect the patroller can do better in some fashion.

Guess: Some concave type region will exist below.

7.5 Improving the Hamiltonian bound

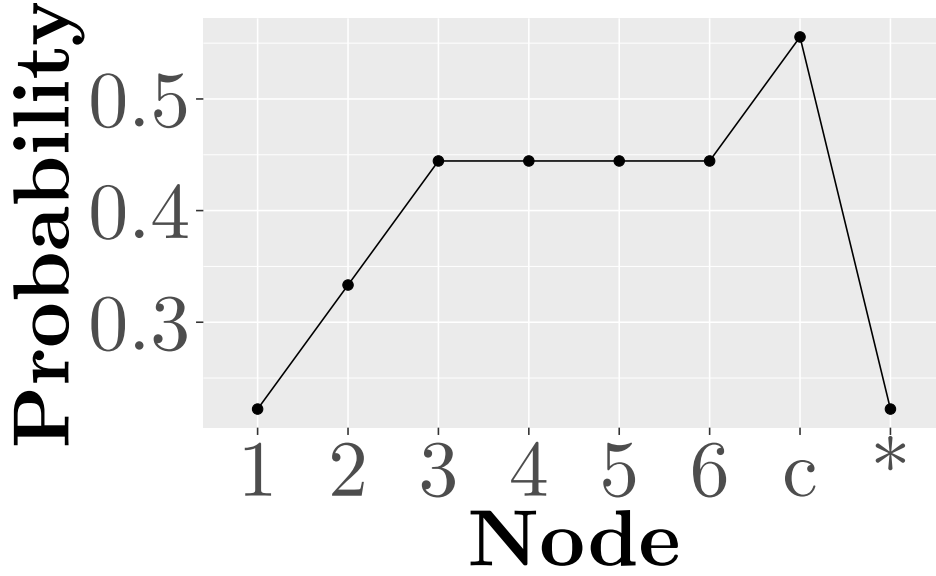
We could suggest a similar improvement to that done on the line, that is to improve the probability of intercepting an attack when following the random oscillation

Probability of interception

If the patroller is performing a Random Oscillation, then for a pure attack at node i the probability of capture is given by (derivation in appednix C.1),

$$w(i) = \begin{cases} \frac{\min(m+2(i-1), 2m)}{2(n+k)}, & \text{for } i \leq \frac{n+k}{2} + 1, \\ \frac{\min(m+2(n+k+1-i), 2m)}{2(n+k)}, & \text{for } i > \frac{n+k}{2} + 1, \\ \frac{\min(m+2(n-1), nm)}{2(n+k)}, & \text{for } i = c, \\ \frac{m}{2(n+k)}, & \text{for } i = *. \end{cases} \quad (1)$$

We will call $w(i)$ the probability of *interception* at node i .



Example Figure 7.1: Interception probabilities of S_4^5 when $m = 4$.

Definition 7.8 (End-ensuring cycle). Define a cycle of length m (if even) or $m - 1$ (if odd) to be *End-ensuring* if one of the points along the cycle is a leaf node. Define the *Half-length* of such cycles to be $\hat{m} = \lfloor \frac{m}{2} \rfloor$.

As End-ensuring cycles return to all the nodes in the cycle every m (if even) or $m - 1$ (if odd) it means that any pure attack against a node in the End-ensuring cycle will be guaranteed to be caught by such a patrol.

In [?] the authors improved nodes with poor interception probabilities by introducing two End-ensuring cycles. We shall do the same, though now more consideration needs to be taken on how to place these End-ensuring cycles.

The minimum probability of interception, $w_{min} \equiv \min_{i=1, \dots, k+1, c, *} w(i)$, gives the lower bound $V \geq w_{min}$. Hence improving this minimum will improve the lower bound. As it stands $w(1) = w(*) = w_{min}$, so improvement is required about 1 and $*$.

First consider partitioning the node set, N , into $L = \{i \mid i \leq \lfloor \frac{m}{2} \rfloor + 1, i \leq k+1\}$, $M = \{i \mid \lfloor \frac{m}{2} \rfloor + 2 \leq i \leq n+k - \lfloor \frac{m}{2} \rfloor, i \leq k+1\}$, $R = \{i \mid i \geq n+k+1 - \lfloor \frac{m}{2} \rfloor, i \leq k+1\}$, $S = \{c, *\}$. Then $w_{min} \equiv w_{min}^N = \min\{w_{min}^L, w_{min}^M, w_{min}^R, w_{min}^S\}$, where $w_{min}^X = \min_{i \in X} w(i)$.

Note. The set R is empty if $\lfloor \frac{m}{2} \rfloor \leq n-1$. The set M is empty if $\lfloor \frac{m}{2} \rfloor \geq k-1$ or $\lfloor \frac{m}{2} \rfloor \geq n+k-2$.

Now from equation 1

- $w_{min}^L = w(1)$, as for $i \in L$ we have that $w(i)$ is increasing in i .
- $w_{min}^M = w(\lfloor \frac{m}{2} \rfloor + 2) = \frac{2m}{2(n+k)}$, as for $i \in M$ we have that $w(i) = \frac{2m}{2(n+k)} \forall i \in M$.
- $w_{min}^R = w(k+1)$, as for $i \in R$ we have that $w(i)$ is decreasing in i .
- $w_{min}^S = w(*)$, as $w(c) > w(*)$.

So as long as the improvement made on L and M is non-decreasing then only the nodes 1 and $\lfloor \frac{m}{2} \rfloor + 2$ need be considered. Similarly if in R , the improvement is non-increasing then we only need to consider the node $k+1$. Finally if in S , the improvement improves node c as good as $*$ then we only need to consider $*$. Therefore we shall only consider such improvement strategies.

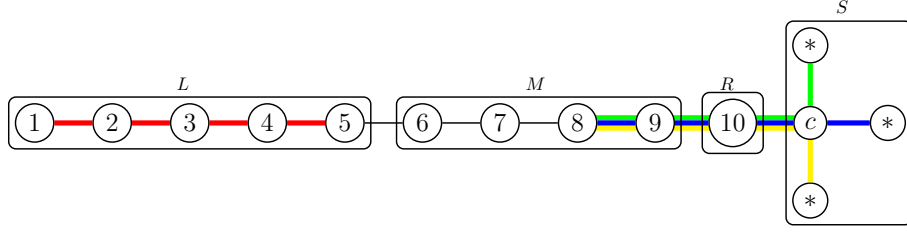
Similarly let $C_{min}^X(\pi) = \min_{i \in X} P(\pi, i)$, where $P(\pi, i) = \max_{I \subset \mathcal{I}} P(\pi, [i, I])$, be the probability of capture under π and $C_{min}(\pi) \equiv C_{min}^N = \min\{C_{min}^L, C_{min}^M, C_{min}^R, C_{min}^S\}$. Then we seek to select π to get $C_{min}(\pi) > W_{min} \equiv C_{min}(\pi_0)$, where π_0 is the Random Oscillation strategy.

We will use P to be the probability the Random Oscillation is played and $Q = 1 - P$. Let $Q = Q_L + Q_M + Q_R + Q_S$, where Q_X is the amount of probability used to improve the probability of intersection for the set X . We use q_X to be probability that the minimum of the set is improved by, so $C_{min}^X = PW_{min}^X + q_X$.

A Naive Improvement Let $\pi = \alpha(Q_L, Q_S)$ denote the Naive Improvement Policy. We will play the end-ensuring cycle, $\{1, \dots, \lfloor \frac{m}{2} \rfloor + 1, \dots, 1\}$, with probability Q_L (a non-decreasing improvement), giving $q_L = Q_L$. We will play, with probability Q_S , an end-ensuring cycle for each $*$ node, $\{*, c, \dots, k+3 - \lfloor \frac{m}{2} \rfloor, \dots, c, *\}$, (a non-decreasing improvement), giving $q_S = \frac{Q_S}{n-1}$.

Now we may also have improved some of the nodes in R , possibly up to node $k+3 - \lfloor \frac{m}{2} \rfloor \leq n+k+1 - \lfloor \frac{m}{2} \rfloor$ (as $n \geq 3$), meaning that all nodes in R are in the end-ensuring cycle $\{*, c, \dots, k+3 - \lfloor \frac{m}{2} \rfloor\}$, so $q_R = q_S$. If $M \neq \emptyset$ nodes in M may be improved, but the node $\lfloor \frac{m}{2} \rfloor + 2$ will not be improved as $\lfloor \frac{m}{2} \rfloor + 2 > \lfloor \frac{m}{2} \rfloor + 1$ and $\lfloor \frac{m}{2} \rfloor + 2 < k+3 - \lfloor \frac{m}{2} \rfloor$ (as $\lfloor \frac{m}{2} \rfloor < k-1$ when $M \neq \emptyset$), so $q_M = 0$.

Using the Naive Improvement Policy we can achieve an improvement over the Random Oscillation if $2(n+k) - nm \geq 0$ and get a bound of $V \geq \frac{2m}{2(n+k)+nm}$



Example Figure 7.2: The Naive Improvement on S_4^9 for $m = 9$. The red lines indicating the end-ensuring cycle $\{1, 2, 3, 4, 5, 4, 3, 2, 1\}$ and the other coloured lines indicating the end-ensuring cycles, for each $*$, $\{*, c, 10, 9, 8, 9, 10, c, *\}$ (as $\lfloor \frac{m}{2} \rfloor = 4$).

(or $V \geq \frac{1}{n}$ if $M = \emptyset, R = \emptyset$), by choosing optimal Q_L and Q_S (See appendix C.2).

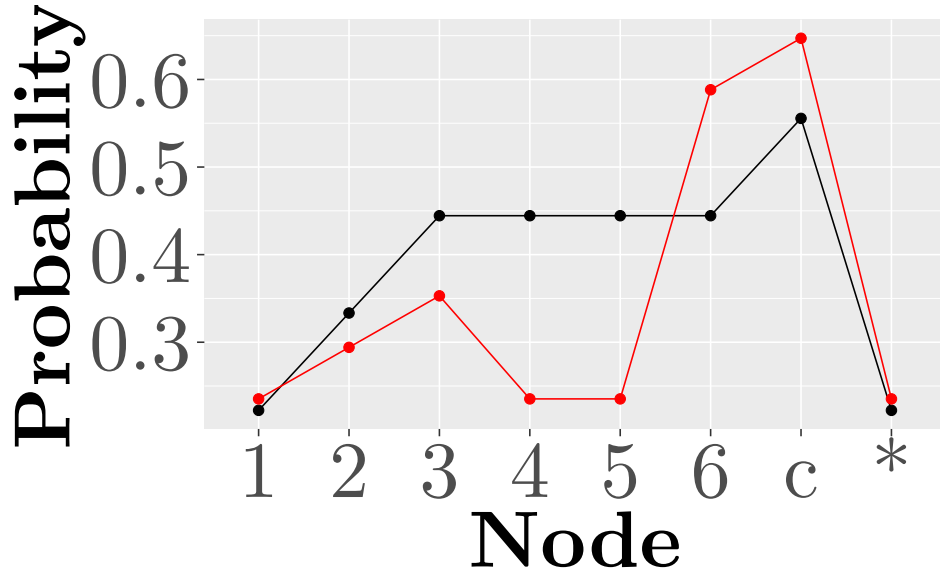


Figure 7.5: Interception probabilities of S_4^5 when $m = 4$, with the red Probabilities showing the Naive Improvement Policy $\alpha(\frac{2}{17}, \frac{6}{17})$.

A Combinatorial Improvement Let $\pi = \beta(Q_L, Q_S)$ denote the Combinatorial Improvement Policy. As before the end-ensuring cycle, $\{1, \dots, \lfloor \frac{m}{2} \rfloor + 1, \dots, 1\}$ is played with probability Q_L , so $q_L = Q_L$. Also;

Case i) If $R \neq \emptyset$ then $\lfloor \frac{m}{2} \rfloor = n + r$, for some excess r . Then form an end-ensuring cycle on the nodes $\{n + k + 1 - \lfloor \frac{m}{2} \rfloor, \dots, k + 1, c, *, c, *, \dots, c, *, c, k + 1, \dots, n + k + 1 - \lfloor \frac{m}{2} \rfloor\}$, this is of length $2(n - \lfloor \frac{m}{2} \rfloor + 1) + 2(n - 1) = 4n - 2\frac{m}{2} = 4n - 2(n + r) = 2n + 2r = 2\lfloor \frac{m}{2} \rfloor \leq m$ (so are end-ensuring). This cycle

will be played with probability Q_S and improves all the nodes in R (non-increasing) and S (c is better than $*$) so $q_R = Q_R$ and $q_S = Q_S$.

Case ii) If $R = \emptyset$ then an end-ensuring cycle is formed by choosing $\lfloor \frac{m}{2} \rfloor$ $*$ nodes each equally likely. This construction is performed with probability Q_S and nodes in S (c is better than $*$). The actually improvement made is

$$q_S = Q_S \times \mathbb{P}(\text{A particular } * \text{ node is picked}) = Q_S \times \frac{\lfloor \frac{m}{2} \rfloor}{n-1}$$

To distinguish between the cases we shall label case i) β_1 and case ii) β_2 .

Using the Choosing Improvement Policy we can achieve an improvement over the Random Oscillation in both cases (under some conditions) and get a lower bound. (See appendix C.3)

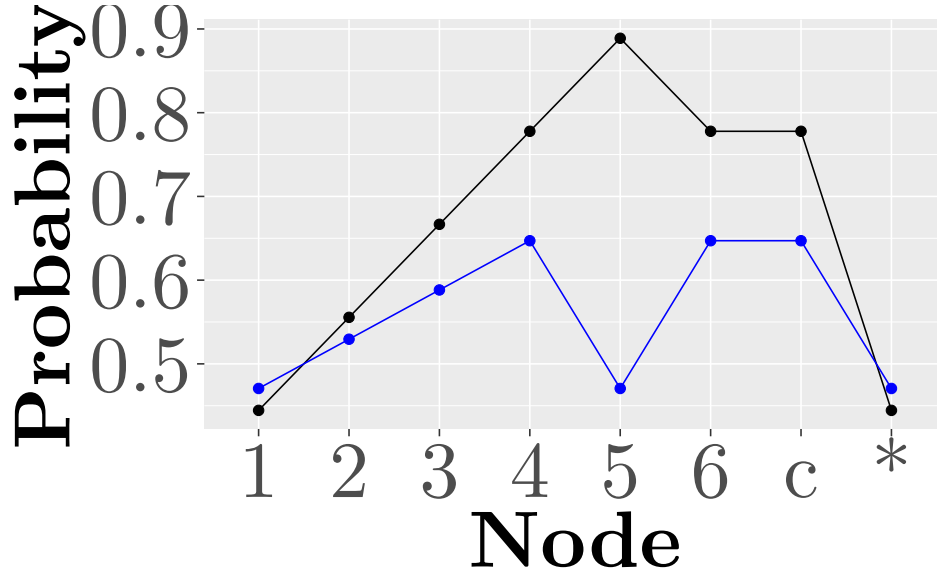


Figure 7.6: Interception probabilities of S_4^5 when $m = 8$, with the blue Probabilities showing the Choosing Improvement Policy $\beta_1 (\frac{2}{13}, \frac{2}{13})$.

7.6 Solution for $m = 2k + 1$ and $m = 2k$

Solution for $m = 2k + 1$

We will now present the solution for $m = 2k + 1$ which will depend on the comparison of m to $2(n - 1)$.

Let us first consider the case of $m \leq 2(n - 1)$. The patroller can look at the decomposition into S_{n-1} and L_{k+1} giving $V \geq \frac{1}{\frac{1}{V(S_{n-1})} + \frac{1}{V(L_{k+1})}} = \frac{1}{1 + \frac{2(n-1)}{2k+1}} = \frac{2k+1}{2k+1+2(n-1)} = \frac{m}{m+2(n-1)}$.

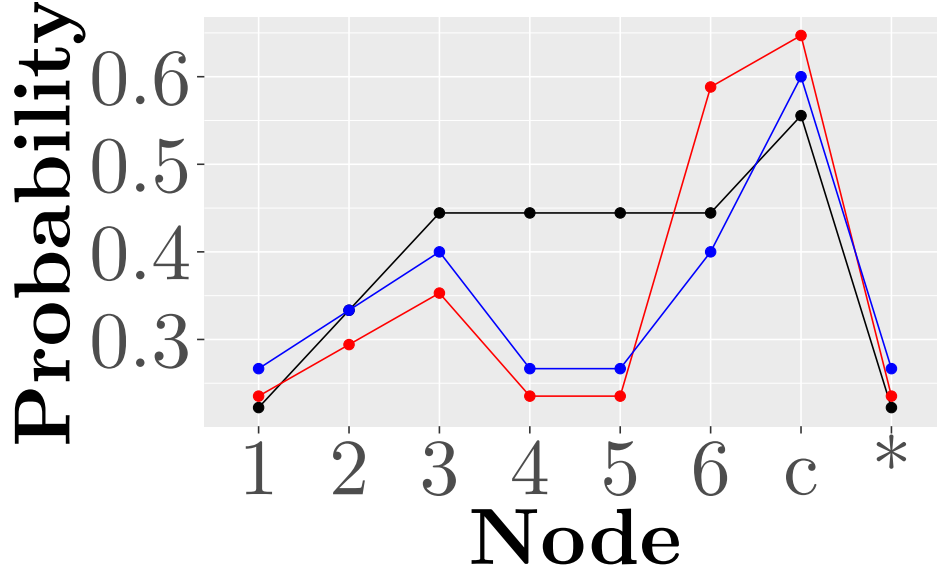


Figure 7.7: Interception probabilities of S_4^5 when $m = 4$, with the red Probabilities showing the Naive Improvement Policy $\alpha(\frac{2}{17}, \frac{6}{17})$ and the blue Probabilities showing the Choosing Improvement Policy $\beta_2(\frac{1}{7}, \frac{3}{14})$.

We suggest that the attacker can ‘augment’ their time-delayed strategy, this can be done by placing attacks at times $\tau, \dots, \tau + 2k$ at the elongated node and times $\tau + k, \tau + k + 1$ (or $k - 1, k$ is proposed to work just as well), giving the bound $V \leq \frac{2k+1}{2k+1+2(n-1)} = \frac{m}{m+2(n-1)}$.

Hence giving $V = \frac{m}{m+2(n-1)}$.

Proof of ‘augmented’ time-delayed strategy: NEEDED TO BE COMPLETE (IDEA: follows the same logic as the time-delayed strategy, the patroller cannot get more than m attacks)

In the case of $m > 2(n - 1)$. The patroller can use the combinatorial improvement with $R = \emptyset$, $M = \emptyset$, giving the bound $V \leq \frac{\lfloor \frac{m}{2} \rfloor}{\lfloor \frac{m}{2} \rfloor + n - 1} = \frac{2k}{2k+2(n-1)}$

Note. Different decompositions give a worse bound. eg. into S_n, L_k give $V \geq \frac{m}{m+2n} = \frac{2k+1}{2k+1+2n}$ (This has yet to be formally proved, but from this even if it is possible to get not 1 value for the star, then the bottom of the fraction is worse than this, and even this doesn’t beat the combinatorial improvement)

The attacker again ‘augments’ their time-delayed attack on to place attacks at times $0, \dots, 2k - 1$ at the elongated node and times $k - 1, k$ at normal external nodes, giving the bound $V \geq \frac{2k}{2k+2(n-1)}$.

Hence giving $V = \frac{2k}{2k+2(n-1)} = \frac{m-1}{m-1+2(n-1)}$

Proof of ‘augmented’ time-delayed strategy: NEEDED TO BE COMPLETE
(IDEA: follows the same logic as the time-delayed strategy, the patroller cannot get more than $2k$ attacks)

So we know that the solution in the region of $m \leq 2(n-1)$ is higher than the solution in the region of $m > 2(n-1)$

Solution for $m = 2k$

Here we will notice that we can follow the above, the patroller has both the decomposition into S_{n-1}, L_{k+1} (available only if $m \leq 2(n-1)$) and the combinatorial improvement (Always available) giving the same bound of $V \geq \frac{2k}{2k+2(n-1)} = \frac{m}{m+2(n-1)}$.

The patroller as above can use an ‘augmented’ version of the time-delayed strategy

7.7 Solution for $m = 2k - 1$ and $m = 2k - 2$

First we will note that $M \neq \emptyset$

Solution for $m = 2k - 1$ As before it is assumed that it will matter on our comparison of m to $2(n-1)$.

when $m \geq 2n - 1$ If $m \geq 2n - 1$ then $R \neq \emptyset$ we can use the combinatorial bound to get $V \geq \frac{m}{n+k+m} = \frac{2k-1}{n+3k-1}$. We note that this improvement is only possible if $m \leq n+k$ i.e $2k-1 \leq n+k$ so $k \leq n+1$. so $m = 2k-1 \leq 2n+1$ which means we are only allowed to use this improvement in the region where $2k-1 = 2n-1, 2n+1$ i.e $k = n, n+1$.

We will then consider both cases

for $k = n$ If $k = n$ then the bound is $V \geq \frac{2n-1}{4n-1}$, we claim in this case that this is the same as using the decomposition to $L_k = L_n$ and S_n getting us a bound of $V \geq \frac{1}{1+\frac{2n}{2n-1}} = \frac{2n-1}{4n-1}$.

Now the attacker requires an augmentation to there strategy

for $k = n + 1$ If $k = n + 1$ then the bound is $V \geq \frac{2n+1}{4n+2}$

The attacker should augment their attacks to simply place only $2(k-1)$ at the elongated node and $2(n-1)$ time centered around the centre node (2 for each external node). This will give a bound of $V \leq \frac{2(k-1)}{2(k-1)+2(n-1)} = \frac{2(k-1)}{2(n+k-2)}$.

When $m \leq 2(n-1)$ If we look at $m \leq 2(n-1)$, then we know that $R = \emptyset$. So the combinatorial bound gives us $V \geq \frac{2(2k-1)}{2(n+k)+(2k-1)(1+\frac{n-1}{k-2})} = \frac{4k-2}{2n+2k+2k-1+\frac{(n-1)(2k-1)}{k-2}}$

If $m < 2(n-1)$ then $R = \emptyset$ and we can use the combinatorial bound to get

$$V \geq \frac{2m}{2(n+k)+m(1+\lceil \frac{n-1}{2} \rceil)} = \frac{2(2k-1)}{2(n+k)+(2k-1)(1+\frac{n-1}{k-1})} = \frac{4k-2}{2(n+k)+(2k-1)+\frac{(2k-1)(n-1)}{k-1}}$$

Solution for $m = 2k - 2$

If $m \geq 2n - 1$ then $R \neq \emptyset$ and we can use the combinatorial bound to get

$$V \geq \frac{\lceil \frac{m}{2} \rceil}{\frac{m}{2} + n - 1} = \frac{2k-2}{2k-2+2(n-1)} = \frac{2k-2}{2(n+k-2)} = \frac{m}{2(n+k-2)}.$$

The attacker should augment their attacks to simply place only $2(k-1)$ at the elongated node and $2(n-1)$ time centered around the centre node (2 for each external node). This will give a bound of $V \leq \frac{m}{2(k-1)+2(n-1)} = \frac{m}{2(n+k-2)}.$

If $m < 2(n-1)$ then $R = \emptyset$ and we can use the combinatorial bound to get $V \geq \frac{2m}{2(n+k)+m(1+\lceil \frac{n-1}{2} \rceil)} = \frac{4k-4}{2(n+k)+(2k-2)(1+\frac{n-1}{k-1})} = \frac{4k-4}{2(n+k)+(2k-2)+2(n-1)} = \frac{2k-2}{n+k+k+n-2} = \frac{m}{2(n+k-1)}.$ An alternative here is decomposition into S_{n-1} and L_{k+1} which should give the same bound.

The attacker should augment their attacks to place only $2(k-1)$ at the elongated node and $2n$ time centred around the centre (n nodes, so one is along the elongated branch). This will give a bound of $V \leq \frac{m}{2(k-1)+2n} = \frac{m}{2(n+k-1)}.$

Note. The dependency on m in relation to how much of the elongated branch it can cover if its got to cover the normal nodes, then if m is high it can stretch along the elongated branch as well. This means when $R \neq \emptyset$ the attacker is at a disadvantage

7.8 Solution for certain m step-points

Stage-1-step-points For the first step-point the ‘simplification’ S_n^k to S_{n+1}^{k-2} then the attacker will get a bound of $V \leq \frac{m}{2(n+k-1)}$ for $m \geq 2(k-1)$.

At the first step-point $m = 2k-2$ Then suppose that the combinatorial improvement holds true with $R = \emptyset$ then we can get the bound $V \leq \frac{2m}{2(n+k)+(2k-2)(1+\frac{n-1}{k-1})} = \frac{m}{n+k+k-1+n-1} = \frac{m}{2(n+k-1)}$

Hence at this step point giving a value of $V = \frac{2k-2}{2(n+k-1)}.$

We will have more step points upon different ‘simplifications’ i.e the second step point occurs upon the simplification to $S_{n+1}^{(k-4,1)}$ and give us a bound of $V \leq \frac{m}{2(n+1+k-3)} = \frac{m}{2(n+k-2)}$ for $m \geq 2(k-3)$.

At the second step point $m = 2k-6$ Then supposing the combinatorial improvement holds then we can get assuming $R = \emptyset$ we can get $V \leq \frac{2(2k-6)}{2(n+k)+(2k-6)(1+\frac{n-1}{k-3})} = \frac{4k-12}{2n+2k+2k-6+2(n-1)} = \frac{2k-6}{2n+2k-4} = \frac{m}{2(n+k-4)}.$ Giving the value.

This will happen in general, so let us set up the formal method

Definition 7.9 (Stage-1-Step-Points). When $m = 2k + 2 - 4i \leq 2(n-1)$ for

some $i = 1, \dots, \lfloor \frac{k+1}{3} \rfloor$, we call this the stage1- i^{th} step-point.

Note. We require $m \leq 2(n-1)$ so that $R = \emptyset$ and the ‘correct’ combinatorial improvement works

Lemma 7.10 (Stage1-Step-point Values). *When m is the i^{th} step-point then*

$$V = \frac{m}{2(n+k-i)}$$

Proof: D.7

We could think that we can extend this idea further, however the limit on $i_{\max} = \lfloor \frac{k+1}{3} \rfloor$ is placed such that we are guaranteed to get improvement.

Various stage step points

We could consider a different amount of stages i.e S_7^5 could be simplified to S_9 when $m = 2$, however it seems like the combinatorial improvement does not help us out. It still feels like this is the best the attacker can do though.

Note. It is worth noting that in this case the simplification is also like looking for the largest independent set (which here is 9 nodes big).

We could also consider S_{10}^9 simplified to $S_{12}^{1,1}$ when $m = 4$ however this would give us a bound of $V \leq \frac{4}{28}$. Now in this case we can note that $4(1 + \frac{9}{2}) = 22 \leq 2(n+k) = 38$ so we could use the combinatorial improvement to get $V \geq \frac{4}{19+11} = \frac{4}{30}$. So we do not arrive at a tight bound.

Note. The graph $S_{10,9}$ is bipartite with $|B| = 9 + 5 = 14$, so we get $V \leq \frac{4}{28}$ this way as well.

The question is can the attacker or patroller do better? I think that the patroller can do better. The question is how?

Note. Our current step points limit us to at best $m = \frac{2}{3}(k+1)$

7.9 Various Stage-step points

As previously discussed we could consider more ‘centres’, we shall construct a variety of points below our current limit of $\frac{2}{3}(k+1)$, we shall work our way up

For $m = 2$, we have already used this idea to get the answer.

Now consider $m = 4$, what possible constructions are there, well the question is how should the branch be split up, we have $k+1$ to either pick to be centres, points along a branch or an attack point. Now we look from the left construction of placing 4 attacks, then we have $k-2$ nodes left between two centres, now we need to get the most out of this space using only 2 and 4 attacks. However in this case they use the same amount of space.

Infact this is general. Hence we will only every use 2 type attacks for the rest of the line. Hence we use $2(n-1) + 4 + 2 \lceil \frac{k-2}{2} \rceil$ attacks and guarantee catching at most $m = 4$ giving $V \leq \frac{4}{2n+k}$ (if k is even) or $V \leq \frac{4}{2n+k+1}$ (if k is odd)

More generally for m even we get $2(n-1) + m + 2 \lceil \frac{k-\frac{m}{2}}{2} \rceil$ and catch at most m giving us a bound of $V \leq \frac{m}{2n-2+m+k-\frac{m}{2}} = \frac{2m}{4n+2k+m-4}$ (if $k - \frac{m}{2}$ is even) or $V \leq \frac{m}{2n-2+m+k-\frac{m}{2}+1} = \frac{2m}{4n+2k+m-2}$ (if $k - \frac{m}{2}$ is odd).

However it may be possible that if $k - \frac{m}{2}$ is even then we can make an additional single attack in the 'remainder', we suggest $V \leq \frac{2m}{4n+2k+m-2}$ for any m even

Note. This argument agrees with the Stage1-step-points as we are using a particular step phase for m .

We now look at even $m \leq 2(n+k)$ and the combinatorial improvement (assuming $R \neq \emptyset$, i.e $m \leq 2(n-1)$) then $V \geq \frac{2m}{2(n+k)+m(1+\frac{n-1}{\lceil \frac{m}{2} \rceil})} = \frac{2m}{2(n+k)+m+2(n-1)} = \frac{2m}{4n+2k+m-2}$. This gives a match when m is even and $k - \frac{m}{2}$ is odd.

For m odd we get $2(n-1) + m + 2 \lceil \frac{k-\frac{m+1}{2}}{2} \rceil$ and catch at most m giving a bound of $V \leq \frac{m}{2n-2+m+k-\frac{m+1}{2}} = \frac{2m}{4n+2k+m-5}$ (if $k - \frac{m+1}{2}$ is even) and $V \leq \frac{m}{2n-2+m+k-\frac{m+1}{2}+1} = \frac{2m}{4n+2k+m-3}$ (if $k - \frac{m+1}{2}$ is odd).

Note. For a fixed k as we go through a cycle for m .

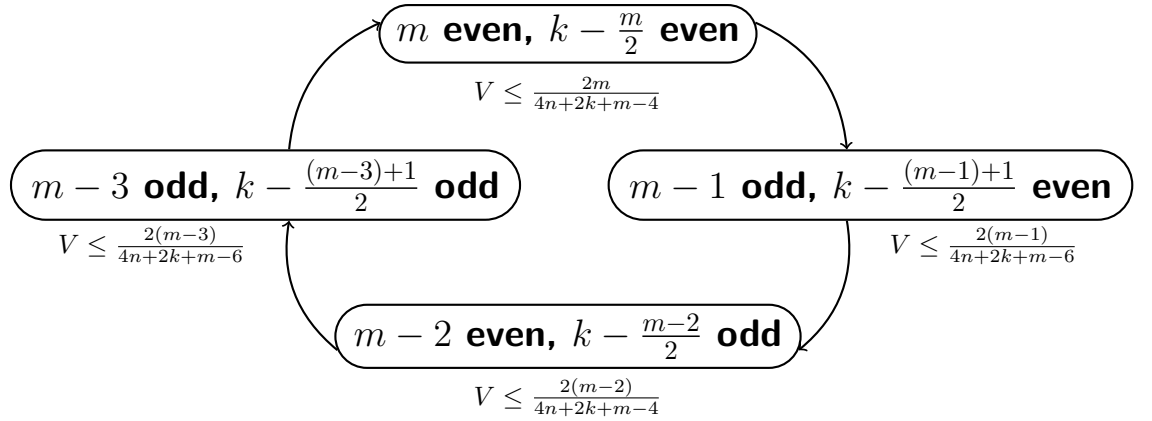


Figure 7.8: Cycle of m decreasing

The method seems to give us a pattern for the denominator which is 4 periodic but inside the period alternates between 2 values (and does 'better than expected' when m is even).

We can notice the other even m seems to be close to the Combinatorial improvement but is slightly too high, this could suggest that we can make a further

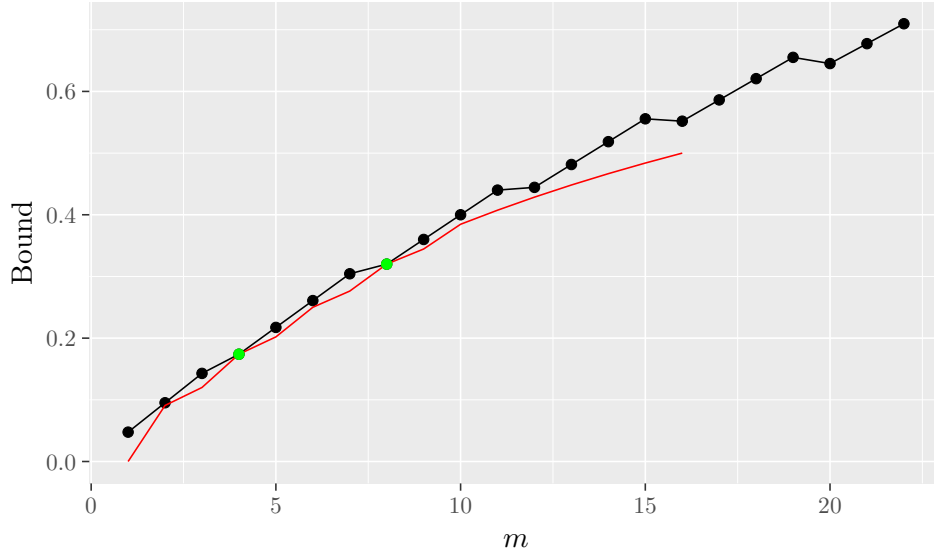


Figure 7.9: A plot of the UpperBound and the Combinatorial Improvement bound for S_6^{10}

improvement on the combinatorial (Though logistically this seems hard) or we can perform a better attack (This may require something like on the line). We will go to this type of analysis now

7.10 Lower Line type attacks

Looking at S_7^5 We note that by the above we know the solution for all even m bar 6, 10 in the graph. We present a solution for $m = 6$ in the form of a grid

Note. We note that a reflection is possible at the end $*$ and we still gain the number again (unlike usual)

This allows us to get $V(S_7^5, m = 6) = \frac{6}{21}$ (by combinatorial and this attack pattern).

We could try to guess the solution for $m = 10$ in S_7^5 using the same idea ? let's investigate, if it holds we wish to get the value of $V(S_7^5, m = 10) = \frac{10}{23}$ to match with the combinatorial improvement, however here $m = 2k$, so the solution is known to give $V(S_7^5, m = 10) = \frac{10}{22}$ by decomposition into S_6 and L_6 and 10 attacks at node 1 and 12 attacks at $*$ nodes (spaced accordingly).

So perhaps we need to consider a bigger example, let us look at S_7^9 and again we look at $m = 6$ first then we would want to get $V = \frac{6}{25}$, this would mean adapting our current attack to place 4 more attacks, well this is infact due to having an additonal 4 nodes along the line we guess we shouldn't have too much

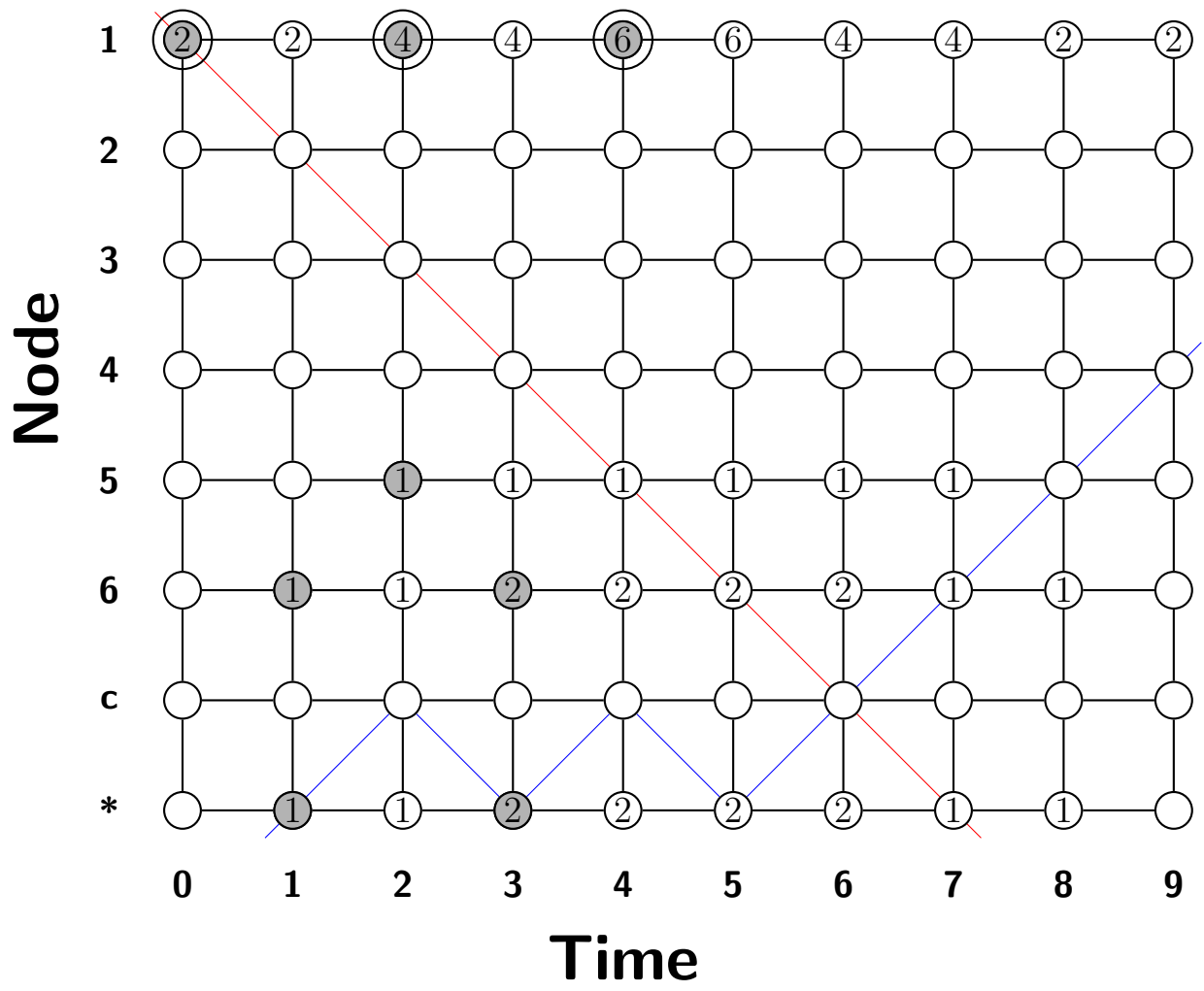


Figure 7.10: Grid solution of $m = 6$ on S_7^5 with two example paths

of an issue.

We have the solution $V(S_7^9, m = 6) = \frac{6}{25}$. Now we want to see if this logic applies to $m = 10$, by the combinatorial bound we expect that $V = \frac{10}{27}$ that is to place 2 more attacks.

As expected.

Hence we need to perform a generalisation to give us the answer when $m \leq 2(n-1)$ for all even m (except when $m = 2k$). We generalise this attack, when m is in the other version of the even cycle (i.e with $k - \frac{m}{2}$ even), by making $m+1+2a$ where $a = n-1 + \left\lceil \frac{k+1-(\frac{m}{2}+1)-1}{2} \right\rceil = n-1 + \frac{k-\frac{m}{2}}{2}$ as $k - \frac{m}{2} - 1$ is odd so $a = \frac{2n-2+k-\frac{m}{2}}{2}$ so we make in total $m+1+2n-2+k-\frac{m}{2} = 2n+k+\frac{m}{2}-1$. So we get $V \leq \frac{m}{2n+k+\frac{m}{2}+1} = \frac{2m}{2(n+k)+m+2(n-1)} = \frac{2m}{2(n+k)+m(1+\frac{n-1}{\frac{m}{2}})}$, hence matching with the combinatorial bound.

Hence a solution to all even $m \leq 2(n-1)$ can be found. Hence our cycle look like this

7.11 Extending even solution

The question is now is it possible to extend the even solution as described above when $m \leq 2(n-1)$ is odd. We note that in the cycle we can see the just look at making 1 less attacks to give an attackers bound. Let $m = m' - 1$ then we assume we can just simply take one attack away from node 1 attacks and we are away.

We can notice that the combinatorial improvement does much worse due to the $\lfloor \frac{m}{2} \rfloor$ part, we may decide a different improvement, following similiar principles is needed.

7.12 Combinatorial loop improvement

We alter the idea of using end-ensuring cycles on the set of S (please note we will deal with only the case of $R \neq \emptyset$ (as that is what we need this for odd m)). We will now improve the $n-1$ nodes by using a cycle of length $m+1$ (note we will be covering the case where m is odd), then when played this cycle has a probability of catching the attacker of $\frac{m}{m+1} \equiv \frac{m}{m'}$.

Now just like before we will need to play a collection of all combinations of such cycles, each cycle has a choice of $\frac{m'}{2}$ end points to visit. So we have a total of $\binom{n-1}{\frac{m'}{2}}$ and $\binom{n-2}{\frac{m'}{2}-1}$ contain any given end point. Hence we have

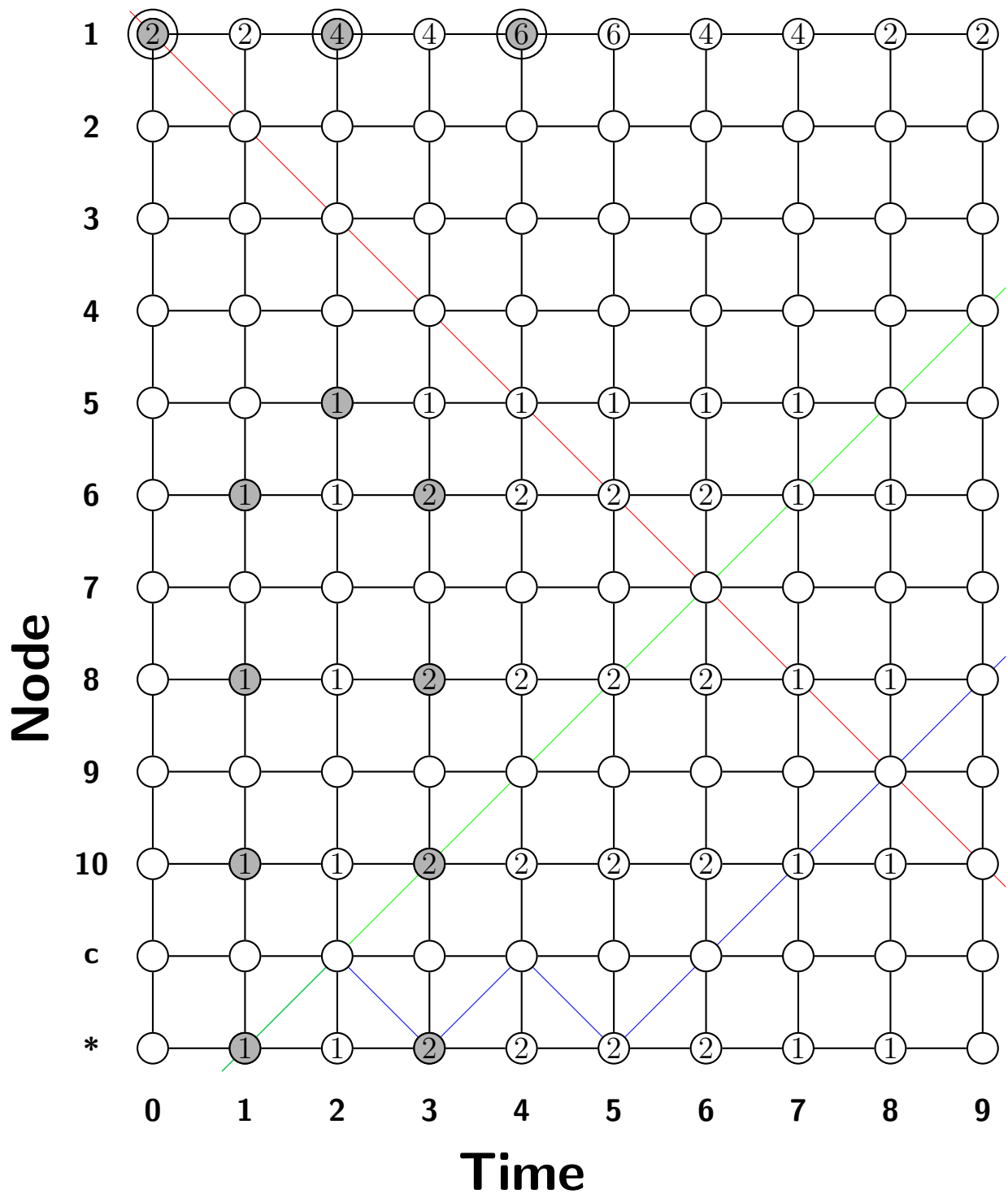


Figure 7.11: Grid solution of $m = 6$ on S_7^9 with three example paths

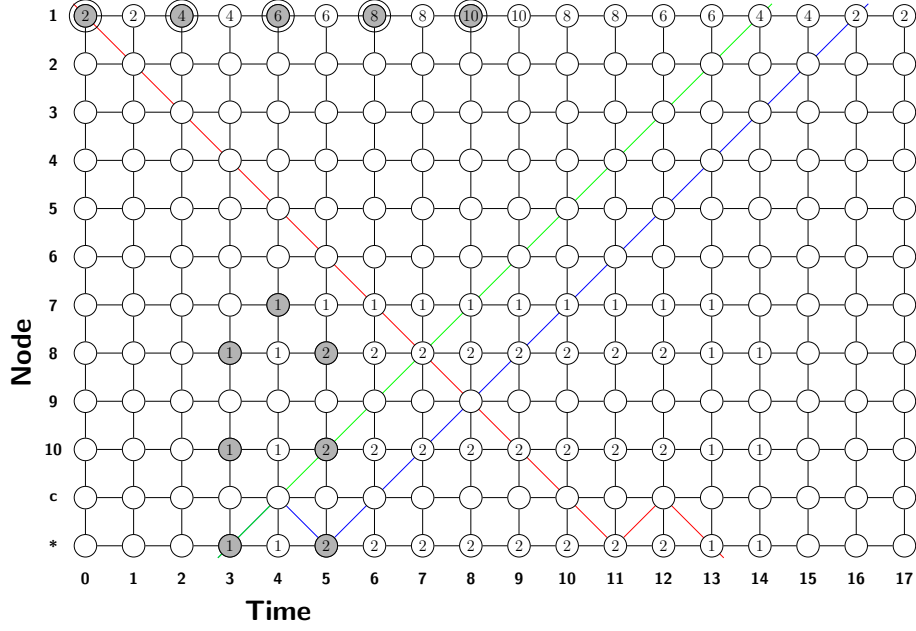


Figure 7.12: Grid solution of $m = 10$ on S_7^9 with three example paths

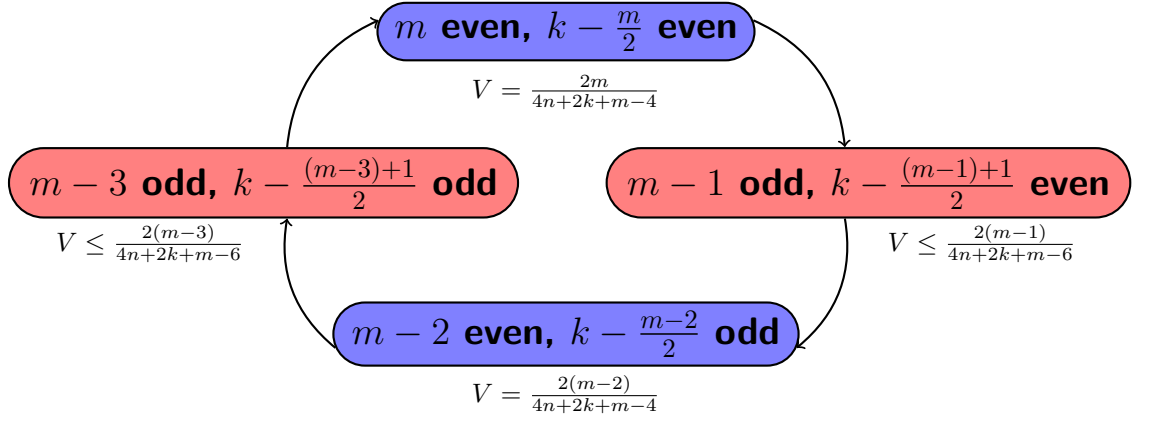


Figure 7.13: Cycle of m decreasing

$\frac{\binom{n-1}{\frac{m'}{2}}}{\binom{n-2}{\frac{m'}{2}-1}} = \frac{m'}{2(n-1)}$ Hence when we play this strategy with probability q , we improve each end with a probability of $q \times \frac{m'}{2(n-1)}$.

We will give the details and the bounds in the appendix C.4

But here we note that getting such a bound analogous to before is possible.

7.13 Odd solution

By using the combinatorial loop improvement when m is odd, then we can get a bound of $V \geq \frac{m}{2(n+k)+m+2(n-1)}$. We claim we can also get $V \leq \frac{2m}{2(n+k)+m+2(n-1)}$, now we know the solution for $m' = m + 1$ which is $V \leq \frac{2m'}{2(n+k)+m'+2(n-1)} = \frac{2m+2}{2(n+k)+m+1+2(n-1)}$, we now want to suggest moving 1 attack from the system

Let us say that $m = 5$ in S_7^9 then we have by placing attacks that $V \leq \frac{2m}{2(n+k)+m+2(n-1)-1}$ (if $k - \frac{m+1}{2}$ is odd) or $V \leq \frac{2m}{2(n+k)+m+2(n-1)-3}$ (if $k - \frac{m+1}{2}$ is even)

i.e as $9 - 3 = 6$ even we get $V \leq \frac{10}{32+5+10-3} = \frac{10}{48} = \frac{5}{24}$, now the combinatorial improvement gives me $V \geq \frac{10}{32+5+10} = \frac{5}{24.5}$. So from our $m = 6$ solution which would have even $k - \frac{6}{2}$, in which we got $V = \frac{6}{16+3+6} = \frac{6}{25}$, so from here it should be a case of removing an attack

A solution to get the combinatorial loop bound can be made by making a particular type of attack. Looking at a particular example on S_7^5 .

Note. In the picture we are using a double type attack as we need whole number to get our bound of $V \leq \frac{14}{43}$

Solution just after m even $k - \frac{m}{2}$ odd In this m is odd and $k - \frac{m+1}{2}$ is still odd In general this type of attack will make $2m$ attacks at node 1 and $4(n-1)$ (in total) attacks at the $*$ nodes, it will also make $4 * \left\lceil \frac{k+1-\frac{m+1}{2}-1}{2} \right\rceil + 1 = 4 * \left\lceil \frac{k-\frac{m+1}{2}}{2} \right\rceil + 1 = 4 * \left(\frac{k-\frac{m+1}{2}+1}{2} \right) + 1 = 2k - m - 1 + 2 + 1 = 2k - m + 2$ attacks in the middle. So in total $2m + 4(n-1) + 2k - m + 2 = 2(n+k) + m + 2(n-1)$. Hence the bound of $V \leq \frac{2m}{2(n+k)+m+2(n-1)}$ is achieved and this part of the cycle is solved

Solution just after m even $k - \frac{m}{2}$ even So here m is odd and $k - \frac{m+1}{2}$ is even. We again need to look at augmenting the strategy

Again we present a strategy on S_7^5 as an insight

Again we now reason that we make $2m$ attacks at node 1 and $4(n-1)$ (in total) attacks at $*$ nodes and in the middle we have $4 * \left\lceil \frac{k+1-\frac{m+1}{2}-2}{2} \right\rceil + 3 =$

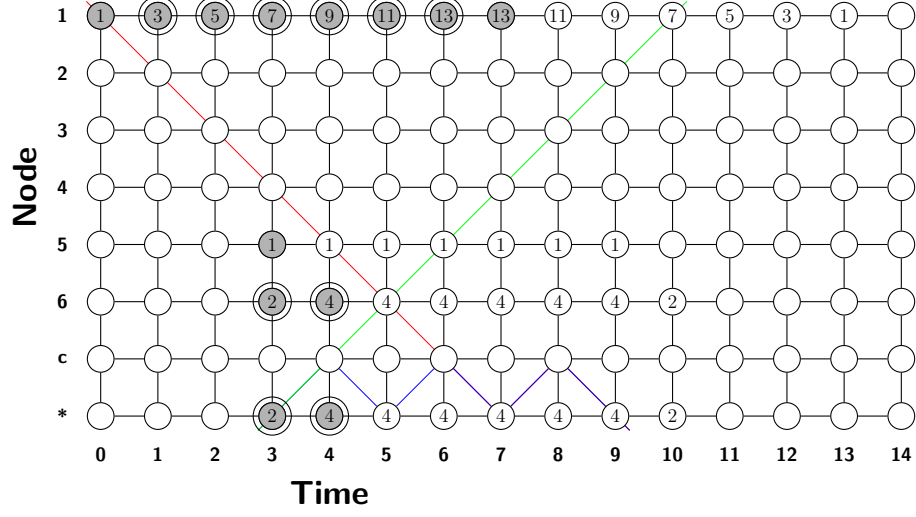


Figure 7.14: Grid solution of $m = 7$ on S_7^5 with two example paths

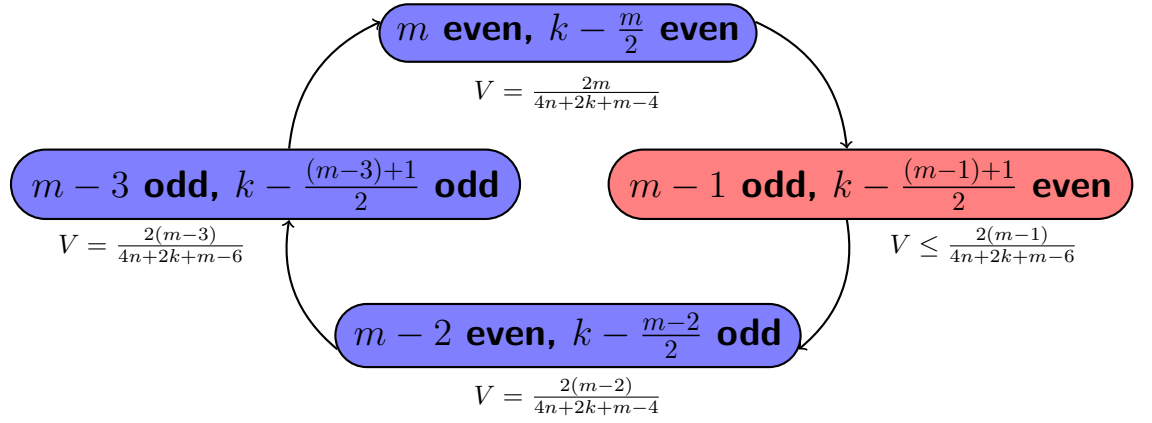


Figure 7.15: Cycle of m decreasing

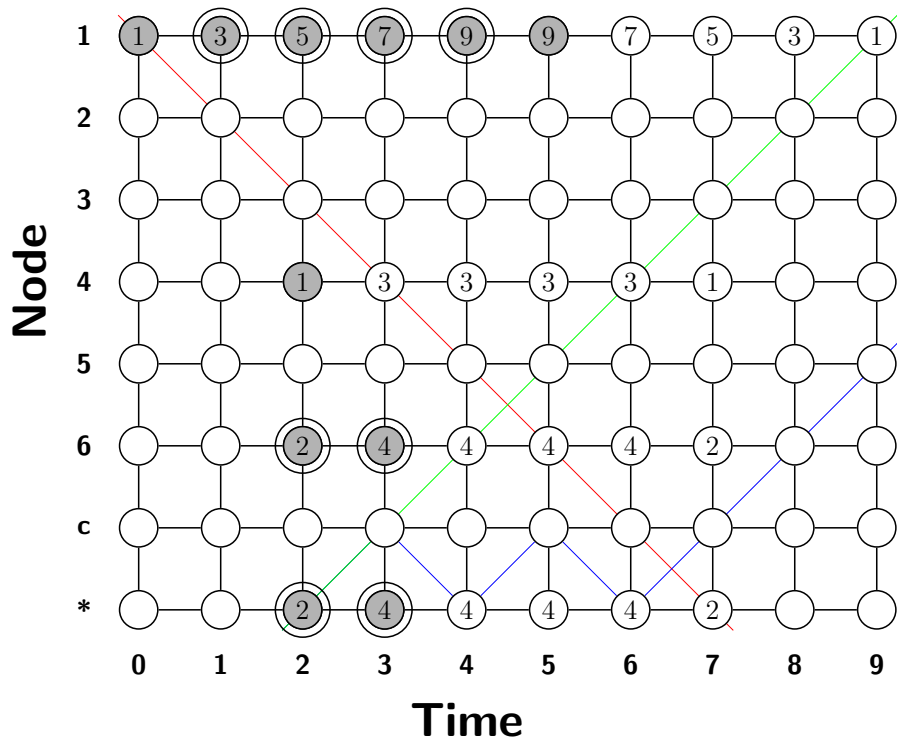


Figure 7.16: Grid solution of $m = 5$ on S_7^5 with two example paths

$4 * \left\lceil \frac{k - \frac{m+1}{2} - 1}{2} \right\rceil + 3 = 4 * \left(\frac{k - \frac{m+1}{2} - 1 + 1}{2} \right) + 3 = 2k - m - 1 + 3 = 2k - m + 2$, for a total of $2m + 4(n-1) + 2k - m + 2 = 2(n+k) + m + 2(n-1)$ attacks. Giving again the bound of $V \leq \frac{2m}{2(n+k)+m+2(n-1)}$ matching with the combinatorial loop bound.

7.14 Solution for $m \leq 2(n-1)$

Know we align ourselves with the solution for $m = 2k - 1$ and $m = 2k - 2$ by using these type ideas. However $m = 2k + 1$ and $m = 2k$ need to still be kept separate. But below this and before $m = 2$ or $m = 1$ we now know the answer, that it follows the combinatorial improvement (for $V = \frac{2m}{2(n+k)+m+2(n-1)}$)

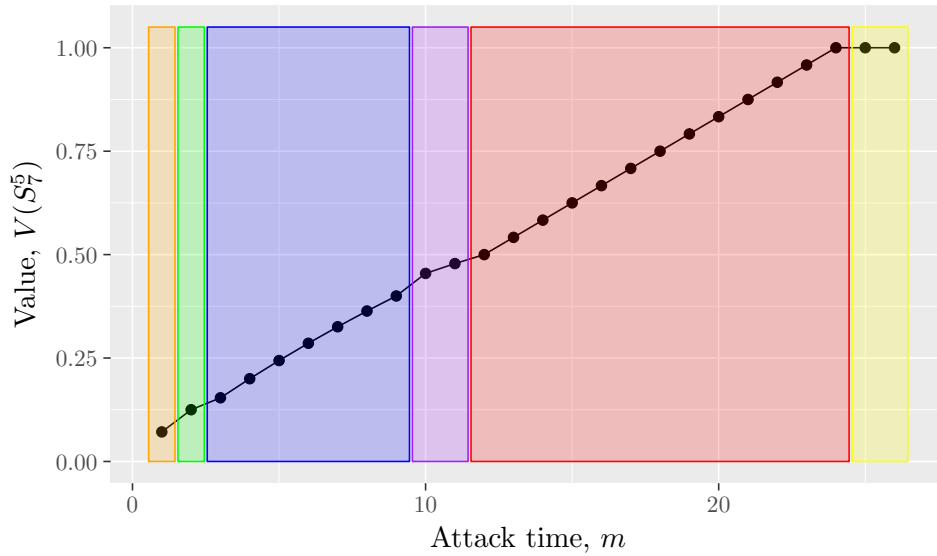


Figure 7.17: A graph of Values of S_7^5 for various m

Hence we have a complete solution if $2k \leq 2(n-1) \iff k \leq n-1$. Otherwise we will have a gap in our current solution till it $m \leq 2(n-1)$.

7.15 Solution when $m > 2(n-1)$

We suggest that we may seek answers along the line of the combinatorial improvement $V \geq \frac{2m}{2n+2k+2m}$. We will first note that $2n + 2k + 2m > 2n + m + 2(n-1) + 2k = 2(n+k) + m + 2(n-1)$ (so we note that we will be doing worse here than in the region $m \leq 2(n-1)$). This may hint that this strategy is not too good.

We believe that due to the nature of the combinatorial bound giving $V \geq \frac{m}{m+n+k} = \frac{m}{m+n'-1}$, where $n' = n + k + 1$ that they exist a line type solution

using $L_{n'}$ as attacks on this give the bound $V \leq \frac{m}{m+n'-1} = \frac{m}{m+n+k}$. However the issue with how to ‘split’ one of the ends to the star end is one that needs solving and it is not immediately, or even whether this will work, it just seems logical.

8 General star graph

8.1 Introduction

We can extension the ‘ k long’ elongated star graph, S_n^k , to the ‘ \mathbf{k} long’ elongated star graph, $S_n^{\mathbf{k}}$. Where $\mathbf{k} \in \mathbb{N}^h$ for some $h = 0, \dots, n$. This Vector \mathbf{k} describes the length of elongation of a subset of the n external nodes.

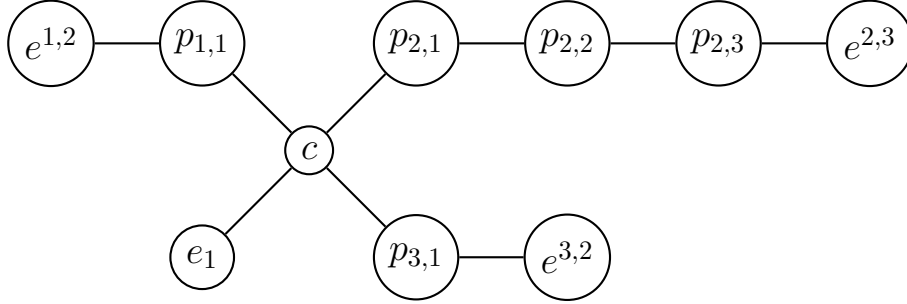


Figure 8.1: Labelling of $S_4^{1,3,1}$

The relabelling of nodes is such that ‘elongations’ can be identified by its position in the vector and the value of that position. $p_{i,j}$ is the ‘penultimate’ node of the i^{th} elongation and is currently j distance from the centre ($j = 0, \dots, n-1$) and e^{i,k_i} is similar the end of the i^{th} elongation, which is $k_i + 1$ distance from the centre. Any unaffected external nodes are labelled e_i for $i = 1, \dots, n-h$.

For ease of notation we will define $|k| \equiv \sum_{i=0}^{i=h} k_i$ and $k_{\max} \equiv \max_{i=1,2,\dots,h} k_i$

Definition 8.1 (General star graph). The *General star graph*, $S_n^{\mathbf{k}}$ ($\mathbf{k} \in \mathbb{N}^h$) is made from S_n , by performing subdivision on h of the initial edges each repeated k_i for $i = 1, \dots, h$.

Note. For ease of notation we will use the above format, but will understand that using $k_i = 0$ is a valid construction, so we can image that $\mathbf{k} \in \mathbb{N}^n$

8.2 Basic Analysis

To start our analysis of this graph, we will look at an expanded graph which can be simplified down to our general star graph. Consider the cyclic graph $C_{2(n+|k|)}$

, we can simplify this graph by node identification to $S_n^{\mathbf{k}}$. The identification mapping is harder to visual from a cycle graph.

For ease of understanding look at an example as in Figure 8.2

The mapping is done as such:

- The centre is identified from nodes $1, 1+2(k_1+1), 1+2(\sum_{i=1}^2(k_i+1)), \dots, 1+2(\sum_{i=1}^h(k_i+1), 1+2(|k|+h)+2), \dots, 1+2(|k|+h)+2(n-h)$
- The first branch is identified from the nodes between 2 and $2(k_1+1)$ (Inclusive). $n_{1,i}$ for $i = 1, \dots, k_1$ are identified by the two nodes $i+1$ and $2k_1+3-i$, the node n_{1,k_1+1} is identified by the one node k_1+2 .
- The j^{th} branch is identified from nodes between $2(\sum_{i=1}^{j-1}(k_i+1))$ and $2(\sum_{i=1}^j(k_i+1))$ (Inclusive). $n_{j,i}$ for $i = 1, \dots, k_j$ are identified by the two nodes $2(\sum_{i=1}^{j-1}(k_i+1)) + (i-1)$ and $2(\sum_{i=1}^j(k_i+1)) - (i-1)$, the node n_{j,k_j+1} is identified by the one node $2(\sum_{i=1}^{j-1}(k_i+1)) + k_j$.

This mapping gives rise to the general Oscillation

Definition 8.2 (General Random Oscillation). The *oscillation* on $S_n^{\mathbf{k}}$ is any embedded Hamiltonian patrol on $C_{2(n+|k|)}$ under the simplification above. The *random oscillation* on $S_n^{\mathbf{k}}$ is the embedded random Hamiltonian patrol on $C_{2(n+|k|)}$ under the simplification above.

Lemma 8.3. For $m < 2(n+|k|)$ following the random oscillation

$$V(S_n^{\mathbf{k}}) \geq V(C_{2(n+|k|)}) = \frac{m}{2(n+|k|)}$$

and if $m \geq 2(n+|k|)$ then $V(S_n^{\mathbf{k}}) = 1$, achieved by any oscillation.

8.3 solution for $m \geq 2(k_{max} + 1)$

We will match the oscillation bound of $\frac{m}{2(n+|k|)}$, by further extending time-limited attack and time-delayed attack into the type-delayed attack.

Definition 8.4 (Node types). A *type i* node, is an external node which has been extended i times. Let $k_{max} \equiv \max_{i=1, \dots, h} k_i$ be the maximum node type on the general extended star graph, $S_n^{\mathbf{k}}$.

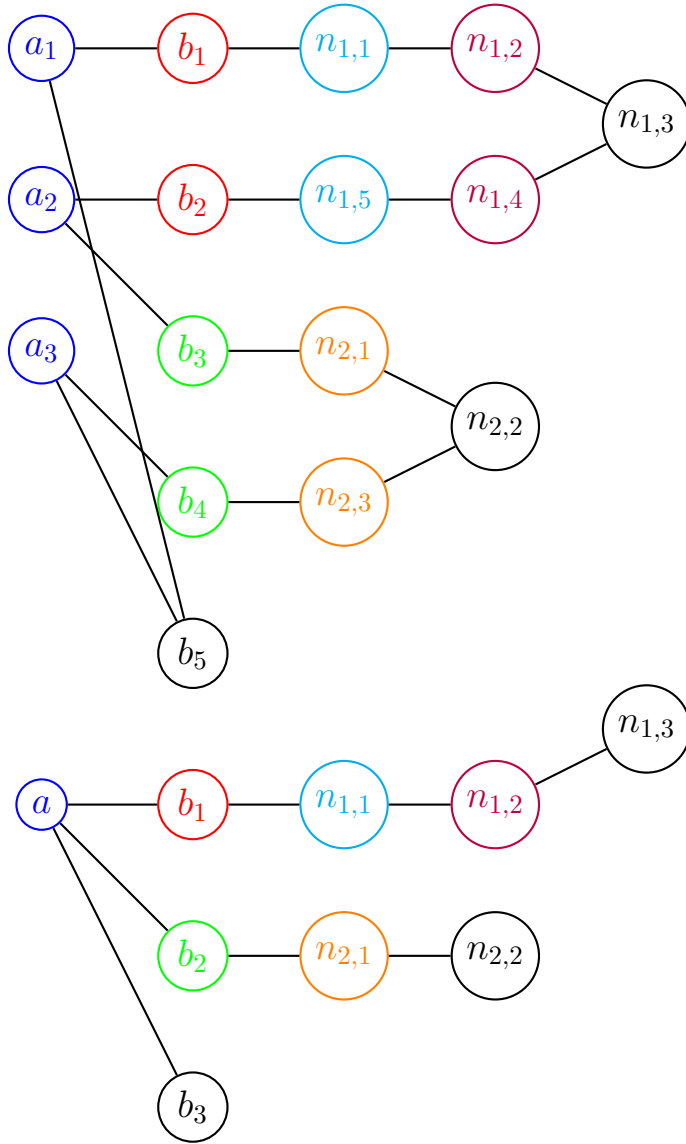
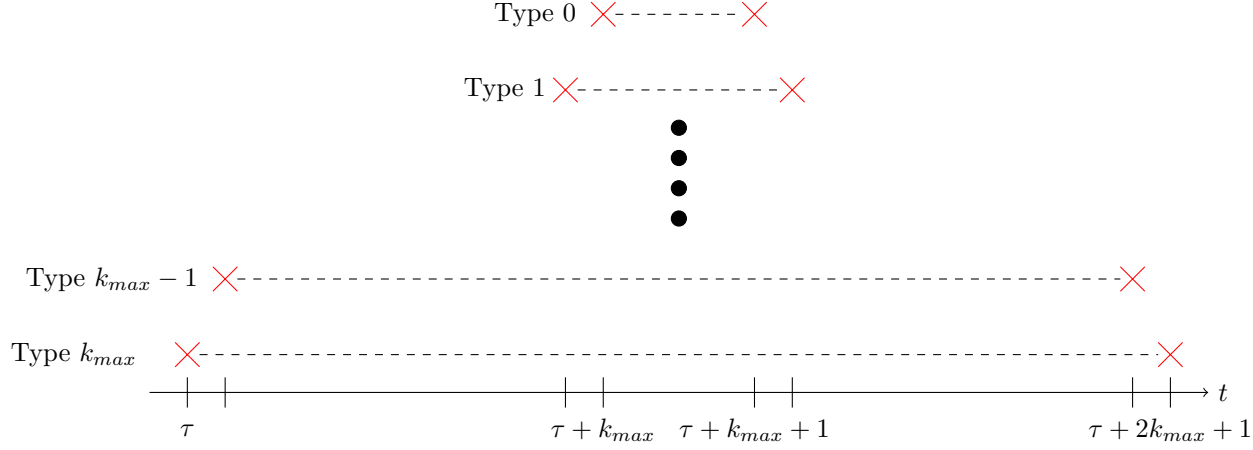


Figure 8.2: Simplification of C_{16} to $S_3^{3,2}$

Definition 8.5 (Type-delayed attack). Let the *Type-delayed attack*, be the attack that attacks at a type i node with probability $\frac{i+1}{n + \sum_{j=1}^h k_j} \forall i$. Choosing an attack interval I , the attack at a type i node chooses a interval from the following with equal probability: $I + (k_{max} - i), I + (k_{max} - i) + 1, \dots, I + k_{max} + i + 1 \forall i$ (i.e statring attacks at a type i node at times $\tau + (k_{max} - i) + 1, \dots, \tau + (k_{max} + i) + 1$).



Theorem 8.6. When $T \geq m + 2k_{max}$, the analogous ‘diametric’ bound is given by

$$V(S_n^k) \leq \max \left\{ \frac{k_{max} + 1}{n + \sum_{j=1}^h k_j}, \frac{m}{2 \left(n + \sum_{j=1}^h k_j \right)} \right\}$$

Proof: D.6

Corollary 8.7 (Solution in $m \geq 2(k_{max} + 1)$). By the attack using the type-delayed attack and the patroller using the random oscillation we achieve the value, when $2(k_{max} + 1) \leq m \leq 2(n + |k|)$,

$$V = \frac{m}{2(n + |k|)}$$

and when $m > 2(n + |k|)$ then $V = 1$.

9 Dual star graphs

10 Linked Star graphs

Appendices

A Graph Definitions

Definition A.1 (Graph). A *graph*, $G = G(N, E)$, is made up of: a set of *nodes* (also called *vertices* or *points*), N , which are places ; and a set of *edges* (also called *arcs* or *lines*), E , which are connections between places, so elements of E must be two-element subsets of N .

Definition A.2 (Subgraph). A graph $Q' = (N', E')$ is said to be a *subgraph* of $Q = (N, E)$ if $N' \subset N$ and $E' \subset E$.

A subgraph is said to be *induced* by N' (or *edge-preserving*) if E' contains all edges (from E) that have both end points in N' .

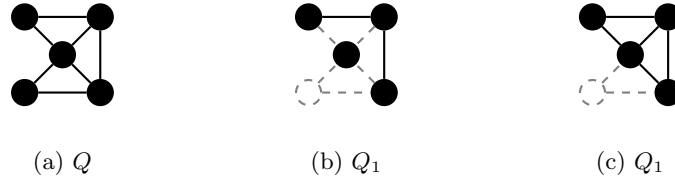


Figure A.1: Q_1 is a subgraph of Q . However it is not induced as it is missing possible edges connecting nodes that existed in Q . Q_2 shows the induced subgraph on the chosen set of nodes.

Definition A.3 (Walk, Path, Trail, Cycle). A sequence of nodes (n_0, n_1, \dots, n_l) is a *walk* of length l if $e_{n_i, n_{i+1}} \in E \forall i = 0, \dots, l-1$. Corresponding to a walk is the sequence of l edges $(e_{n_0, n_1}, e_{n_1, n_2}, \dots, e_{n_{l-1}, n_l})$.

A walk becomes a trail if each edge in the walk is distinct, i.e $e_{n_i, n_{i+1}} \neq e_{n_j, n_{j+1}} \forall i \neq j$. A trail becomes a path if each node in the walk is distinct (except possibly the start and final node), i.e $n_i \neq n_j \forall i \neq j \geq l-1$.

A walk, trail or path is said to be *closed* if the start and end nodes are the same. A *cycle* is a closed path with length, $l \geq 3$ (with the special case of $l = 3$ being called a *triangle*).

Definition A.4 (Hamiltonian cycle). A *Hamiltonian cycle* is a cycle which contains every node on the graph, i.e it is a cycle of length $l = |N|$. A graph that exhibits a Hamiltonian cycle is called *Hamiltonian*.

Example A.5. For the graph Q as in Figure A.1:

- An example of a walk is $(1, 2, 1, 5, 4, 2)$
- An example of a trail is $(1, 2, 5, 3, 4, 5, 1)$

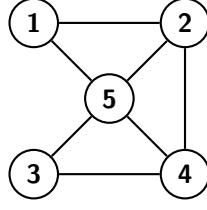


Figure A.1: Graph, Q

- An example of a path is $(1, 2, 4, 3)$
- An example of a Hamiltonian cycle is $(1, 2, 4, 3, 5, 1)$

Hence we would call the graph Q Hamiltonian.

Definition A.6 (Complete graphs). The *complete graph*, K_n , is a graph of n nodes, in which all edges are present, i.e $e_{i,i'} \in E \forall i, i' \in N$.

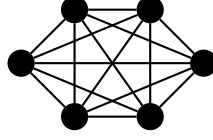


Figure A.2: The complete graph of 6 nodes, K_6 .

Definition A.7 (Bipartite). A graph is said to be *bipartite* if $N = A \cup B$, where $A \cap B = \emptyset$, and $e_{i,i'} \notin E \forall i, i' \in A$, $e_{i,i'} \notin E \forall i, i' \in B$.

Definition A.8 (Complete bipartite). The *complete bipartite graph*, $K_{a,b}$, is a bipartite graph of $a + b$ nodes (where $|A| = a, |B| = b$), in which all edges are present, i.e $e_{i,i'} \in E \forall i \in A \forall i' \in B$ and $e_{i,i'} \in E \forall i \in B \forall i' \in A$.

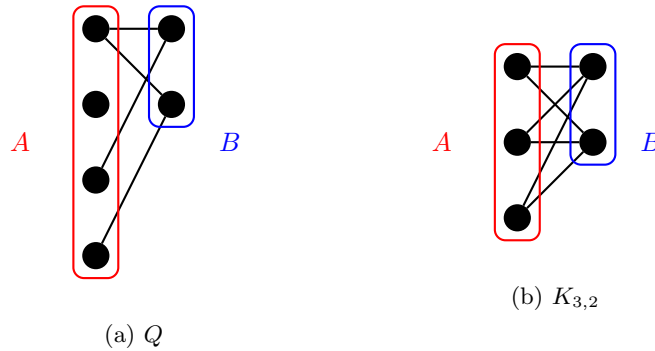


Figure A.3: A.3a is an example of a bipartite graph, Q . A.3b is the complete bipartite graph with set sizes of 3 and 2.

Definition A.9 (Subdivision,Smoothing). A *Subdivision* (or *expansion*) of a graph, G , is a new graph G' which is made by subdividing a chosen edge. The subdivision of an edge $\{u, v\}$ yields a graph with a new node w and the splitting of the edge $\{u, v\}$ into $\{u, w\}$ and $\{w, v\}$.

The reverse process is called *Smoothing* of a graph, G , is a new graph G' which is made by smoothing between two nodes. The smoothing out of a node pair (u, v) , with $d(u, v) = 2$ and with w between them, then w is removed along with the edges $\{u, w\}$ and $\{v, w\}$, then the edge $\{u, v\}$ is placed to connect u and v .

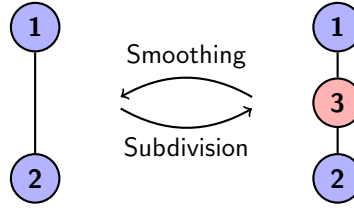


Figure A.4: Subdivision and Smoothing of the edge $\{1, 2\}$

B Old Statements about Old diametric

Ignoring the problem with the old diametric bound, the diametric idea from [?] can be extended when there are multiple diametric pairs.

Definition B.1 (Diametrical set). Let $D = \{i \in N \mid d(i, i') = \bar{d} \quad \forall i' \in D\}$

Lemma B.2. *By the attacker playing equally likely all nodes in D , at a random time interval, called the extended diametric attack gives $V \leq \max \left\{ \frac{m}{|D|\bar{d}}, \frac{1}{|D|} \right\}$*

Proof. The options for the patroller are to wait at a node and have probability $\frac{1}{|D|}$ or to move between a selection of these nodes and then wait (for some time) or move on.

The comparison of m to \bar{d} is important (if $m \leq \bar{d}$ waiting at any given node is good), moving is only beneficial if more attacks can be intercepted by moving rather than waiting that is that $m > \bar{d}$, as this means moving to another node claims m attacks each time compared to waiting which would just give the \bar{d} (as this is the time to move), so in this case moving is always better than waiting.

When moving, moving to the node visited the furthest in the best will give us the best chance of no-overlap, hence a cycle is formed.

So forming a cycle between the $|D|$ nodes each a distance \bar{d} apart means the probability of intercepting is $\frac{m}{|D|\bar{d}}$.

□

Example B.3. For S as seen in Figure B.1. By the attacker attacking with equal probability on the diametric set of the four 3 nodes, which have $\bar{d} = 4$. This gives that $V \leq \max\{\frac{m}{4 \times 4}, \frac{1}{4}\} = \max\{\frac{m}{16}, \frac{1}{4}\}$.

By its simplification from C_{16} , we can get that $V \geq \frac{m}{16}$.

Hence in the region of $m \geq 4$ we get that $V = \frac{m}{16}$

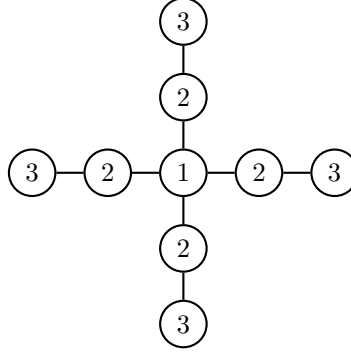


Figure B.1: Graph S used in example

C Improvemening Hamiltonian bound

C.1 Reason For Probability of Interception formula

To argue why this is consider looking at just the integer points on the graph first, then by looking whether it is best to return via the “left return” or a “right return”, that is the shortest time to the next return along the oscillation. The time/distance to return via the left motion, i.e through 1, is $2(i - 1)$ and the time/distance to return via the right motion, i.e through c and the set of $*$'s, is $2(n + k - (i - 1)) = 2(n + k + 1 - i)$. So the categories fall when $2(i - 1) < 2(n + k + 1 - i)$, $2(i - 1) = 2(n + k + 1 - i)$, and $2(i - 1) > 2(n + k + 1 - i)$. These decide the boundaries and they return with these distances.

Therefore the patroller catches the m initially and then either another t or m attacks on the return, where t is the time to next return (either $2(i - 1)$ or $2(n + k + 1 - i)$). This means that the patroller catches $\min(m + t, 2m)$ attacks and hence the probability of interception is $\frac{\min(m + t, 2m)}{2(n + k)}$ as they cycle is of length $2(n + k)$ and the number lies in $\min(m + t, 2m)$ of these.

For c consider that it catches m attacks initially and is returned to $n - 1$ times each at a distance/time of 2 apart, meaning it gains $t = 2$ or m attacks $n - 1$ attacks. Hence it intercepts $\min(m + 2(n - 1), nm)$ out the cycle length of $2(n + k)$.

Finally $*$ nodes are easy as they are they are not returned to in a single cycle, hence they intersect m out the possible $2(n+k)$.

C.2 Naive Improvement Complete analysis

1. If $M = \emptyset, R = \emptyset$ then $C_{min}(\alpha) = \min\{C_{min}^L, C_{min}^S\} = \min\{Pw(1) + q_L, Pw(*) + q_S\} = P \frac{m}{2(n+k)} + \min\{q_L, q_S\}$. As the Patroller wishes to select q_L, q_S to maximize this probability, the problem becomes

$$\begin{aligned} & \text{Maximize} && (1 - q_L - (n-1)q_S) \frac{m}{2(n+k)} + \min\{q_L, q_S\} \\ & \text{Subject to} && q_L + (n-1)q_S \leq 1 \text{ (Probability sum constraint)} \\ & && q_L, q_S \geq 0 \end{aligned}$$

Meaning that as due to the symmetry of q_L, q_S we must have that $q_L = q_S$ which means improvement is only possible if $2(n+k) - nm \geq 0$, then set $q_L = q_S = \frac{1}{n}$. Meaning $P = 0, Q_L = \frac{1}{n}, Q_R = \frac{n-1}{n}$ and giving a lower bound $V \geq \frac{1}{n}$.

2. If $M \neq \emptyset, R = \emptyset$ then $C_{min}(\alpha) = \min\{C_{min}^L, C_{min}^M, C_{min}^S\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} + 2 \rfloor), Pw(*) + q_S\} = P \frac{m}{2(n+k)} + \min\{q_L, P \frac{m}{2(n+k)}, q_S\}$. As the Patroller wishes to select q_L, q_S to maximize this probability, the problem becomes

$$\begin{aligned} & \text{Maximize} && (1 - q_L - q_S) \frac{m}{2(n+k)} + \min\{q_L, (1 - q_L - (n-1)q_S) \frac{m}{2(n+k)}, q_S\} \\ & \text{Subject to} && q_L + (n-1)q_S \leq 1 \text{ (Probability sum constraint)} \\ & && q_L, q_S \geq 0 \end{aligned}$$

Meaning that as due to the symmetry of q_L, q_S we must have that $q_L = q_S$, with improvement only possible if $2(n+k) - nm \geq 0$, which means we seek to maximize $(1 - nq_L) \frac{m}{2(n+k)} + \min\{q_L, (1 - nq_L) \frac{m}{2(n+k)}\}$ giving $q_L = \frac{m}{2(n+k)+nm}$ (as one is then decreasing in q_L and one is increasing in q_L). Meaning $P = \frac{2(n+k)}{2(n+k)+mn}, Q_L = \frac{m}{2(n+k)+mn}, Q_S = \frac{m(n-1)}{2(n+k)+mn}$ and giving a lower bound $V \geq \frac{2m}{2(n+k)+mn}$.

3. If $M \neq \emptyset, R \neq \emptyset$ then $C_{min}(\alpha) = \min\{C_{min}^L, C_{min}^M, C_{min}^R, C_{min}^S\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} + 2 \rfloor), Pw(k+1) + q_S, Pw(*) + q_S\} = P \frac{m}{2(n+k)} + \min\{q_L, P \frac{m}{2(n+k)}, q_S\}$. Now was $Pw(k+1) + q_S \geq Pw(*) + q_S$, we can ignore this element (meaning here it does not matter if R is empty or not). As the Patroller wishes to select q_L, q_S to maximize this probability, the problem becomes

$$\begin{aligned} & \text{Maximize} && (1 - q_L - (n-1)q_S) \frac{m}{2(n+k)} + \min\{q_L, (1 - q_L - (n-1)q_S) \frac{m}{2(n+k)}, q_S\} \\ & \text{Subject to} && q_L + (n-1)q_S \leq 1 \text{ (Probability sum constraint)} \\ & && q_L, q_S \geq 0 \end{aligned}$$

Meaning that as due to the symmetry of q_L, q_S we must have that $q_L = q_S$, with improvement only possible if $2(n+k) - nm \geq 0$, which means we seek to maximize $(1 - nq_L) \frac{m}{2(n+k)} + \min\{q_L, (1 - nq_L) \frac{m}{2(n+k)}\}$ giving $q_L = \frac{m}{2(n+k)+nm}$ (as one is then decreasing in q_L and one is increasing in q_L). Meaning $P = \frac{2(n+k)}{2(n+k)+mn}, Q_L = \frac{m}{2(n+k)+mn}, Q_S = \frac{m(n-1)}{2(n+k)+mn}$ and giving a lower bound $V \geq \frac{2m}{2(n+k)+mn}$.

C.3 Combinatorial Improvement Complete analysis

1. If $M = \emptyset, R = \emptyset$ then $C_{min}(\beta_2) = \min\{C_{min}^L, C_{min}^S\} = \min\{Pw(1) + q_L, Pw(*) + q_S\} = P_{\frac{m}{2(n+k)}} + \min\{q_L, q_S\}$. As the patroller wishes to select q_L, q_S to maximize this probability the problem becomes

$$\begin{aligned} & \text{Maximize} && (1 - q_L - \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_S) \frac{m}{2(n+k)} + \min\{q_L, q_S\} \\ & \text{Subject to} && q_L + \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_S \leq 1 \text{ (Probability sum constraint)} \\ & && q_L, q_S \geq 0 \end{aligned}$$

Meaning that due to the symmetry of q_L, q_S we must have that $q_L = q_S$, which means that improvement is only possible if $2(n+k) - m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor}) \geq 0$. Then setting $q_L = q_S = \frac{\lfloor \frac{m}{2} \rfloor}{\lfloor \frac{m}{2} \rfloor + n - 1}$. Meaning $P = 0, Q_L = \frac{\lfloor \frac{m}{2} \rfloor}{\lfloor \frac{m}{2} \rfloor + n - 1}, Q_S = \frac{n-1}{\lfloor \frac{m}{2} \rfloor + n - 1}$ and giving a lower bound of $V \geq \frac{\lfloor \frac{m}{2} \rfloor}{\lfloor \frac{m}{2} \rfloor + n - 1}$

2. If $M \neq \emptyset, R = \emptyset$ then $C_{min}(\beta_2) = \min\{C_{min}^L, C_{min}^M, C_{min}^S\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2), Pw(*) + q_S\} = P_{\frac{m}{2(n+k)}} + \min\{q_L, P_{\frac{m}{2(n+k), q_S}}\}$. As the patroller wishes to select q_L, q_S to maximize this probability the problem becomes

$$\begin{aligned} & \text{Maximize} && (1 - q_L - \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_S) \frac{m}{2(n+k)} + \min\{q_L, (1 - q_L - \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_S) \frac{m}{2(n+k)}, q_S\} \\ & \text{Subject to} && q_L + \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_S \leq 1 \text{ (Probability sum constraint)} \\ & && q_L, q_S \geq 0 \end{aligned}$$

Meaning that due to symmetry of q_L, q_S we must have that $q_L = q_S$, which means that the improvement is only possible if $2(n+k) - m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor}) \geq 0$.

Then setting $q_L = q_S$ means we must maximize $(1 - (1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_L) \frac{m}{2(n+k)} + \min\{q_L, 1 - (1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor} q_L) \frac{m}{2(n+k)}\}$ giving $q_L = \frac{m}{2(n+k) + m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor})}$ (as one is increasing in q_L and one is decreasing in q_L).

Meaning $P = \frac{2(n+k)}{2(n+k) + m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor})}, Q_L = \frac{m}{2(n+k) + m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor})}, Q_S = \frac{m(n-1)}{\lfloor \frac{m}{2} \rfloor (2(n+k) + m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor}))}$ and giving a lower bound of $V \geq \frac{2m}{2(n+k) + m(1 + \frac{n-1}{\lfloor \frac{m}{2} \rfloor})}$

3. If $M \neq \emptyset, R \neq \emptyset$ then $C_{min}(\beta_1) = \min\{C_{min}^L, C_{min}^M, C_{min}^R, C_{min}^S\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2), Pw(k+1) + q_S, Pw(*) + q_S\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2), Pw(*) + q_S\} = P_{\frac{m}{2(n+k)}} + \min\{q_L, P_{\frac{m}{2(n+k), q_S}}\}$. As $Pw(k+1) + q_S \geq Pw(1) + q_S$, we can ignore the element (meaning here it does not matter if R is empty or not). As the patroller wishes to select q_L, q_S to maximize this probability the problem becomes

$$\begin{aligned} & \text{Maximize} && (1 - q_L - q_S) \frac{m}{2(n+k)} + \min\{q_L, (1 - q_L - q_S) \frac{m}{2(n+k)}, q_S\} \\ & \text{Subject to} && q_L + q_S \leq 1 \text{ (Probability sum constraint)} \\ & && q_L, q_S \geq 0 \end{aligned}$$

Meaning that due to symmetry of q_L, q_S we must have that $q_L = q_S$, which means that the improvement is only possible if $n+k-m \geq 0$. Then

setting $q_L = q_S$ means we must maximize $(1 - 2q_L)\frac{m}{2(n+k)} + \min\{q_L, (1 - 2q_L)\frac{m}{2(n+k)}\}$ giving $q_L = \frac{m}{2(n+k+m)}$ (as one is increasing in q_L and one is decreasing in q_S). Meaning $P = \frac{2(n+k)}{2(n+k+m)}$, $Q_L = \frac{m}{2(n+k+m)}$, $Q_S = \frac{m}{2(n+k+m)}$ and giving a lower bound of $V \geq \frac{2m}{2(n+k+m)}$.

C.4 Combinatorial loop improvement analysis

We are only dealing with m odd so $m' = m + 1$ even and $M \neq \emptyset$ $R = \emptyset$. We will play the hamiltonian bound with probability P , the left end-ensuring improvement with probability q_L and the a particular right loop with probability q_S (so $Q_S = \frac{n-1}{\frac{m'}{2}}q_S$).

$C_{min}(C_{min}^L, C_{min}^M, C_{min}^S) = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2)Pw(*) + \frac{m}{m'}q_S\} = P\frac{m}{2(n+k)} + \min\{q_L, P\frac{m}{2(n+k)}, \frac{m}{m'}q_S\}$ As the patroller wishes to select q_L, q_S to maximise the probability the problem becomes

$$\begin{aligned} \text{Maximize} \quad & (1 - q_L - \frac{n-1}{\frac{m'}{2}}q_S)\frac{m}{2(n+k)} + \min\{q_L, (1 - q_L - \frac{n-1}{\frac{m'}{2}}q_S)\frac{m}{2(n+k)}, \frac{m}{m'}q_S\} \\ \text{Subject to} \quad & q_L + \frac{n-1}{\frac{m'}{2}}q_S \leq 1 \text{ (Probability sum constraint)} \\ & q_L, q_S \geq 0 \end{aligned}$$

Meaning that we must have $q_L = \frac{m}{m'}q_S$ which means the improvement is only possible if $m + 2(n-1) \leq 2(n+k)$ (for similarity to even normal combinatorial improvement $\iff m(1 + \frac{n-1}{\frac{m'}{2}}) \leq 2(n+k)$).

Then setting $q_L = \frac{m}{m'}q_S$ means we must maximize $(1 - \frac{m+2(n-1)}{m}q_L)\frac{m}{2(n+k)} + \min\{q_L, (1 - \frac{m+2(n-1)}{m}q_L)\frac{m}{2(n+k)}\}$ giving $q_L = \frac{m}{2(n+k)+m+2(n-1)}$. Meaning $P = \frac{2(n+k)}{2(n+k)+m+2(n-1)}$, $Q_L = q_L = \frac{m}{2(n+k)+m+2(n-1)}$ and $Q_S = \frac{n-1}{\frac{m'}{2}}q_S = \frac{2(n-1)}{m}q_L = \frac{2(n-1)}{2(n+k)+m+2(n-1)}$. Giving a lower bound of $V \geq \frac{2m}{2(n+k)+m+2(n-1)}$

Note. It is worth noting that this is analogous to the result found by the usual combinatorial improvement when m is even.

D Proof's

D.1 Alternating Random Hamiltonian proof

Proof. During any attack interval I which is of even length, then $W(I)$ contains m' "even" and m' "odd" nodes for a total of $m = 2m'$ nodes. Therefore by following the Alternating Random Hamiltonian Patrol, π_{ARHP} , with probability p at "even" nodes and probability $\frac{2}{n} - p$ at "odd" nodes. Then

$$\begin{aligned}
P(\pi_{ARHP}, [i, I]) &\geq \underbrace{\overbrace{p}^{\text{even node}} + \overbrace{\frac{2}{n} - p}^{\text{odd node}} + p + \frac{2}{n} - p + \dots + p + \frac{2}{n} - p}_{m=2m' \text{ elements}} \\
&= m'p + m'(\frac{2}{n} - p) = \frac{2m'}{n} = \frac{m}{n} \quad \forall i \in N \quad \forall I \subseteq \mathcal{T}
\end{aligned}$$

Hence as it holds for all pure attacks

$$P(\pi_{ARHP}, \phi) \geq \frac{m}{n} \quad \forall \phi \in \Phi$$

Hence $V \geq \frac{m}{n}$. □

If m is odd, say $m = 2m' + 1$ then in the above we get two possibilities for each node depending on the interval choice either $p + \frac{m-1}{n}$ or $\frac{m+1}{n} - p$. So choosing anything other than $p = \frac{1}{n}$ (which is the Random Hamiltonian Patrol strategy) gives a worse result for the patroller.

D.2 Condition on T for old diametric proof

Proof. Using $T = m - 1 + (k + 1)\bar{d}$ in the formula gives, $\alpha = \left\lfloor \frac{(k+1)\bar{d}-m}{\bar{d}} \right\rfloor = (k+1) + \left\lfloor \frac{-m}{\bar{d}} \right\rfloor = (k+1) - 2 = (k-1)$ (the final part is because $2 > \frac{-m}{\bar{d}} \geq -1$ and we will assume that $m > \bar{d}$ here otherwise waiting at one node is just as good as the bound we are trying to achieve)

$$m - \bar{d} + (m + m(k-1))_+ + (m - 1 - (m - 1 + (k-1+1)\bar{d}))_+ + (m - 1 - (m - 1 + (k+1)\bar{d}))_+$$

which is $m - \bar{d} + (mk)_+ + ((k+1)\bar{d} - k\bar{d})_+ + ((k+1)\bar{d} - (k+1)\bar{d})_+$ giving $m - \bar{d} + mk + (\bar{d})_+ + (0)_+ = m(k+1)$. Giving the fraction of $\frac{m(k+1)}{2(k+1)\bar{d}} = \frac{m}{2\bar{d}}$.

For the second part, first we seek to prove that within the choice of T from $m - 1 + (k+1)\bar{d} + r$ where $0 \leq r < \bar{d}$ is the maximum when $r = m$ (i.e $T = 2m - 1 + (k+1)\bar{d}$).

As the choice of r only affects the final 3 parts (middle and ends values), we can just look at considering these values and seeing what the maximal choice is.

$$\text{Upon substitution we get that: } \alpha = \left\lfloor \frac{(k+1)\bar{d}+r-m}{\bar{d}} \right\rfloor = (k+1) + \left\lfloor \frac{r-m}{\bar{d}} \right\rfloor$$

so formula becomes $(m(\alpha+1))_+ + ((k+1)\bar{d} + r - (\alpha+1)\bar{d})_+ + ((k+1)\bar{d} + r - (\alpha+2)\bar{d})_+$. To decide r we need to know if middle values or end values are non-zero.

Note. The second end value will never be non-zero as $(k+1)\bar{d} + r - (\alpha+2)\bar{d} = (k+1)\bar{d} + r - ((k+1) + \left\lfloor \frac{r-m}{\bar{d}} \right\rfloor + 2)\bar{d} = r - (\left\lfloor \frac{r-m}{\bar{d}} \right\rfloor + 2)\bar{d} < r - (-1+2)\bar{d} = r - \bar{d} < 0$.

- No middle values and no end values is impossible assuming $k \in \mathbb{N}_0$.

- Middle values but no end values. As we really want to maximize the end value, increasing r up to the point where α increases (giving a raise of m attacks captured) but increases the number of total attacks by 2 each time it is raised. Hence minimal r is chosen to increase α . This is when $r - m = -\bar{d}$ i.e $r = m - \bar{d}$ changes α by 1 (as $r < m - \bar{d}$ gives an -1 to α , but critical point is when equal to).
- Middle values and end value. As we are looking at $(m - \bar{d})(\alpha + 1) + (k + 1)\bar{d} + r$, we still want to increase α without increasing the number of attacks too much, i.e as above.

Then we show that the maximal subsequence tends to the bound as $T \rightarrow \infty$, i.e as $k \rightarrow \infty$. When substituted, we get that $\alpha = (k + 1)$ and so the formula becomes $m - \bar{d} + (m + m(k + 2))_+ + (2m - 1 - (m - 1 + (k + 2)\bar{d}))_+ + (2m - 1 - (m - 1 + (k + 3)\bar{d}))_+$ giving $m - \bar{d} + (k + 3)m + (m - (k + 2)\bar{d})_+ + (m - (k + 3)\bar{d})_+$. As $m < 2\bar{d}$ then we get $m(k + 4) - \bar{d}$ caught out of $2(m + k\bar{d})$ giving a fraction of $\frac{m(k+4)-\bar{d}}{2(m+k\bar{d})} \rightarrow \frac{m}{2\bar{d}}$.

Hence as the maximal subsequence tends down to the bound, it implies the result. \square

D.3 Time-limited diametric attack proof

Proof. First consider all the pure patrolling strategies, $W_i \in \mathcal{W}$, Then as the attacker is only attacking two ends, henceforth called n_1 and $n_{\bar{d}}$, any patrol not starting at n_1 or $n_{\bar{d}}$ is dominated by one that does. This is because the patrol will not capture any attacks until they visit either n_1 or $n_{\bar{d}}$, and then capture a set of attacks that started there previously. The patrol might as well wait there up until this point and do at least as good as arriving there for the first time.

Formally, assume that n_1 is the end node first reached by a patrol, $W(t)$ at time, $t_1 = \min \{t \mid W(t) = n_1\}$, then we can form the patrol, $U(t) = \begin{cases} n_1 & \text{for } t \leq t_1, \\ W(t) & \text{for } t > t_1. \end{cases}$ and $P(U, \phi) \geq P(W, \phi)$ where ϕ is the timed diametric attack (or infact the normal diametric attack).

Now we are restricted to patrols starting at end points, it is similar to see when leaving an end point, there is no other decision as you must travel to the other end point, assumed to be $n_{\bar{d}}$. Hence the question becomes when to leave n_1 and travel to $n_{\bar{d}}$. Obviously it should only be undertaken if the journey can be made and more attacks can be caught by doing so.

WLOG assume that $\tau = 0$ (other just wait longer initially, as attacks haven't started), then our choice is what leaving time (last time before moving): $t_l \in \{0, 1, \dots, m - 2\}$, to pick to maximize the number of attacks caught; or $t_L = \infty$, never leaving to get \bar{d} attacks.

Leaving: Choosing $t_L \in \{0, 1, \dots, m - 2\}$ gives the patroller $\frac{m}{2\bar{d}}$ as,

Leaving at t_L gives us $\min(t_L + 1, \bar{d})$ attacks caught at n_L , and $\min(m + \bar{d} - 2 - (t_L + \bar{d}) + 1, \bar{d}) = \min(m - 1 - t_L, \bar{d})$.

Now choosing $t_L > \bar{d} - 1$, doesn't improve the first value and possibly lowers the second value. Hence we restrict ourselves to leave if we catch all attacks, i.e $t_L \leq \bar{d} - 1$. Now in this region lowering t_L lowers it by 1 and raises it only raises the second on by 1 if $m - 1 - t_L \leq \bar{d}$ (i.e $t_L \geq m - 1 - \bar{d}$ or any t_L if $m - 1 - \bar{d} \leq 0$). Hence any choice of $(m - 1 - \bar{d})_+ \leq t_L \leq \bar{d} - 1$ is equally as good. This gives a number of attacks caught as $t_L + 1 + m - 1 - t_L = m$ out of $2\bar{d}$ placed attacks. Hence giving $V \leq \frac{m}{2\bar{d}}$.

Staying: Choosing $t_L = \infty$ gives the patroller $\frac{1}{2}$

Hence as the patroller can pick from these two options, it gives $V \leq \max\{\frac{1}{2}, \frac{m}{2\bar{d}}\}$. More explicitly it gives $V \leq \frac{1}{2}$ if $m < \bar{d}$, and $V \leq \frac{m}{2\bar{d}}$ is $m \geq \bar{d}$. \square

D.4 Polygonal attack proof

Proof. First consider all the pure patrolling strategies, $W_i \in \mathcal{W}$, Then we immediately restrict ourselves to pure patrols starting at an attack point, assumed WLOG n_1 is the first node hit by a patrol. This restriction is possibly as if our current strategy, W , doesn't hit n_1 until time $t_{hit} = \min\{t \mid W(t) = n_1\}$, then we can form the patrol $U(t) = \begin{cases} n_1 & \text{for } t \leq t_{hit}, \\ W(t) & \text{for } t > t_1. \end{cases}$ with $P(U, \phi) \geq P(W, \phi)$, where ϕ is the polygonal attack.

Now we are restricted to start at n_1 , and when we leave n_1 we will not return immediately for a similar reason to the above (as we could form a patrol which waits longer before leaving to do at least as good). So when we leave we can only go to one of the other end points, say WLOG n_2 (as all end points attacks are equivalent). Similarly from this position we will be waiting until we leave and then head to another end point, which is not yet visited.

Note. We cannot every have a unvisited end point as $m \leq |D|d$ (otherwise the value is 1) and we will therefore never visit the same node twice.

This means the patroller will be performing a strategy that only chooses when to leave a currently at node. This means a choice among the vector $\mathbf{t} = (t_1, \dots, t_{|D|})$ where $t_i = \max\{t \mid W(t) = n_i\}$, with $t_i = \infty$ meaning never leave and the vector increasing by at least d for increase in index, i.e $t_{i+1} \geq t_i + d$ (and so by induction $t_i \geq (i - 1)d$).

Now we will suppose that $kd \leq m < (k + 1)d$ for some $k \in \{0, 1, \dots, |D| - 1\}$, as this will determine the behaviour of the vector \mathbf{t} and assume WLOG that $\tau = 0$ (so $I = \{0, 1, \dots, m - 1\}$)

We can first argue that when leaving, if the current time, $t \geq d$ then there is no point waiting for any time at a node. This is because past this time, no more

attack at that node are being carried out and hence the maximum number has been caught when arriving at a node. Hence we really only have a decision for the first node t_1 (as $t_i \geq d$ for $i \geq 2$), and this decision will be based on the total number of attacks caught.

Now as $m < (k+1)d$ along our strategy the maximum number of end points we will visit it $(k+1)$.

The formula under this patrol for the number of attacks caught will be

$$\min(t_1 + 1, d) + \min(m + d - 2 - (t_1 + d) + 1, d) + \min(m + d - 2 - (t_1 + 2d) + 1, d) + \dots + \min(m + d - 2 - (t_1 + (k-1)d) + 1, d)$$

Note. No negative values are to be summed

As t_1 is limited to pick $d-1$ or below (as otherwise picking up no additional attacks), we can note that decreases t_1 the first term decreases, and the second term increases only if $0 \leq m-1-t_1 \leq d$, i.e $m-1-d \leq t_1 \leq m-1$.

Now as $m-1-d \geq kd-1-d = (k-1)d-1$ this means that we require that $(k-1)d-1 \leq t_1 \leq d-1$ meaning unless $k=1$, this will not increase, the value will always be d as $\min(m-1-d, d) \geq \min(kd-1-d, d) = \min((k-1)d-1, d) = d$ (for $k \geq 2$).

Similary the j^{th} value, past the first value, has the contribution of $\min(m-1-t_1-(j-1)d, d)$, so for a increase we only get it if $0 \leq m-1-t_1-(j-1)d \leq d \implies m-1-(j-1)d \geq t_1 \geq m-1-jd \geq (k-j)d-1$ and as $t_1 \leq d-1$ it means it will only be increasing if $j=k$ (note. $j=k-1$ gives an acceptable range of t_1 but does not allow it to decrease in the first place).

So decreasing t_1 from $d-1$ by 1 increases the last value by 1 only if $t_1 \geq m-1-kd$. However this still provides overall number of attacks caught. Now note by our

$$\min(t_1 + 1, d) + \underbrace{d + d + \dots + d}_{k-1 \text{ lots of } d} + \min(m-1-t_1+(k-1)d, d) = \min(d-1+1, d) + (k-1)d + \min(m-1-(d-1)+(k-1)d, d)$$

As noting $m-kd \geq 0$ by assumption earlier. This gives $V \leq \frac{m}{|D|d}$

It has already been noted that any choice of t_i for $i \geq 2$ should be $t_i = t_i + (i-1)d$. So there will never be option to use these as $t_i = \infty$. However t_1 could be chosen to be ∞ , this would give a bound of $V \leq \frac{1}{|D|}$.

Hence the patroller can pick from these two options, giving $V \leq \max\{\frac{1}{|D|}, \frac{m}{|D|d}\}$. More explicitly it gives $V \leq \frac{1}{|D|}$ if $m < d$ and $V \leq \frac{m}{|D|d}$ if $m \geq d$. \square

D.5 Time-delayed attack proof

Proof. First Consider a patrolling strategy that is at a $*$ node at time $t \geq k$ (i.e $*$ node attacks have begun) and seek to show that staying amongst the $*$ nodes

until the attacks end is at least as good as moving to node 1 and waiting, and possibly returning to $*$ (if time allows).

We will consider two cases of $m \leq 2(k+1)$ and $m > 2(k+1)$.

1. In this case we will first show that returning to $*$ type nodes is never an option once it is left, consider leaving at t , then the first possible return would be $t + 2(k+2) \geq k + 2(k+2) > k + m$ and hence all the attacks would be caught and there would be no point returning. Hence the only option in this scenario is whether it is worth it leave at this point t and go to node 1 and wait until the end of the game.

If she was to stay around the end nodes then from this point onwards (not including the node we are at, at time t) we would get a payoff of $\frac{k+m-t-1}{2(n+k)}$.

This is because we either get:

$$\begin{aligned}
 k+m \text{ odd } & \underbrace{\frac{1}{n+k} + \dots + \frac{1}{n+k}}_{\frac{k+m-t}{2} \text{ times}} \\
 k+m \text{ even } & \underbrace{\frac{1}{n+k} + \dots + \frac{1}{n+k}}_{\frac{k+m-1-t}{2} - 1 \text{ times}} + \frac{1}{2(n+k)}
 \end{aligned}$$

In either case the payoff for moving about at the $*$ type nodes is as above.

Now consider moving away to 1, which will be reached at time $t+k+2$ and as we must wait here the payoff depends on a few things. It is

$$\frac{\min(m, t+k+2, m-(t+k+2-(2k+1)))}{2(k+1)} \times \frac{k+1}{n+k} = \frac{\min(m, t+k+2, k+m-t-1)}{2(n+k)}$$

now note as $t \geq k \implies m > k+m-t-1$ and as $m \leq 2(k+1) \implies t+k+2 \geq m$. Moreover this implies we will be in the final stretch of attacks and no new attacks will be taking place, so no more attacks will be claimed by waiting here (though moving back is just as fruitless)

Hence in this case the payoff is $\frac{k+m-t-1}{2(n+k)}$. The exact same benefit as to staying around the $*$ nodes, hence both moving and staying are equally as good, so once left a $*$ node she will have to wait at 1 (and infact the game will be over) and we have no incentive not to do it (INFACIT IF THE CONDITION ABOUT LIMITED number of $*$ nodes is brought up it is infact the best option).

Hence for any $t \geq k$ when we are at a $*$ node we might as well move to 1 and end the game.

Now consider being at $*$ for some time $s \leq k-1$, then we can decided to wait to time k and then make the decision as above and move to 1 or we can move immediately to 1.

Waiting to time k gives us $\frac{1}{2(n+k)} + \frac{k+m-k-1}{2(n+k)} = \frac{m}{2(n+k)}$.

Now leaving at time s means we arrive at 1 at time $s + k + 2$, however, if this is the plan then it is clear that starting at 1 is optimal (which we will deal with next).

Now starting at 1 at time 0, consider the decision to move to 1 at time q (under which the decision will complete the game, as moving get us there at time $q + k + 2 \geq k$ so the decision to move back immediately is chosen), This means we get a payoff of

$$\frac{q+1}{2(n+k)} + \frac{1}{n+k} + \frac{m - (q + 2(k+2) - (2k+1))}{2(n+k)} = \frac{q+1}{2(n+k)} + \frac{1}{n+k} + \frac{m-q-3}{2(n+k)} = \frac{m}{2(n+k)}$$

With the knowledge that the choice of q 's choice to achieve this, if $q \geq m-2$ is chosen it is worse than the sum as, not as good on arriving and nothing gained on coming back, hence it only achieves $\frac{q+1}{2(n+k)}$, so $q = 2k+1$ might as well be chosen for $\frac{k+1}{n+k}$ (note. here that $q = 2k+1$ is always in this zone as $m \leq 2(k+1)$)

Hence it is best to wait for all time at node 1 and achieve $\frac{k+1}{n+k} \geq \frac{m}{2(n+k)}$ (for $m \leq 2(k+1)$)

2. When $m > 2(k+1)$, we shall again first consider starting at a $*$ at time $t \geq k$, however now it is not possible to state that once it is left that it can never be returned to. We care about what to do between t and $k+m$, now we will first seek to show that moving and waiting at 1 is just as good as moving around $*$ types.

Moving purely around $*$ nodes will get us as before $\frac{k+m-t-1}{2(n+k)}$. Moving and waiting at 1 gets us $\frac{2(k+1)-(t-m+k+1)+}{2(k+1)} \times \frac{k+1}{n+k} = \frac{k+m-t-1}{2(n+k)}$. (or $\frac{2(k+1)}{2(n+k)} = \frac{k+1}{n+k}$, if they go early enough to catch all attacks)

The only idea that could possibly be better is to move to 1 and then wait for period of time, say q times waiting, and then move back. This would yeild

$$\frac{q + \min(2(k+1), t+k+2, t+k+2 - (t-m+k+1))}{2(k+1)} \times \frac{k+1}{n+k} = \frac{q+2(k+1)}{2(n+k)}$$

($+\frac{1}{n+k}$ or $+\frac{1}{2(n+k)}$ if arriving before all attacks at $*$ have completed) from being there for a period and then remake the decision of what to do from $t+q+2(k+2)$. During this time period between t and $t+q+2(k+2)$ (assuming the game at $*$ is not over yet) then we will get

$$\underbrace{\frac{1}{n+k} + \dots + \frac{1}{n+k}}_{\frac{q+2k+2}{2} \text{ times}} = \frac{q+2k+2}{2(n+k)}$$

($+\frac{1}{n+k}$ or $+\frac{1}{2(n+k)}$ at end)

Hence it is better to move an wait at 1 then return if we want to, however returning serves no purpose as we will leave immediately. Hence moving

to 1 and waiting is the best option if $t \geq k$. Hence consider starting at a $*$, it would be either wait till k and move back or move earlier. Moving earlier would mean that she might as well have started at node 1.

Hence the only option for starting at node $*$ is to get $\frac{1}{2(n+k)} + \frac{k+m-t-1}{2(n+k)}$ (or $\frac{1}{2(n+k)} + \frac{k+1}{n+k}$ if all attacks are caught at node 1 at time $2k+2$ (i.e $m > 2k+3$)).

So starting at $*$ means getting $\frac{k+1}{n+k} + \frac{1}{2(n+k)}$

Let $m = 2(k+1) + r$ for $r \in \mathbb{N}$. Then considering starting at $*$ nodes, then we must decide to move to 1 or wait till k then make a decision. However deciding to move to 1, means we might as well have started at 1.

If we at a $*$ node at some time $t \geq k$ (let $t = k + t_e$), then we can decide to (wait only if $t = k$ until $t = k+1$) move to another $*$ node arriving at $t+2$ or move to 1 arriving at $t+k+2$. Now we aim to show that the option of moving to another $*$ node is not strictly better. Assume it is then it is a repeated action until time $k+m-2 = 3k+r$ or $k+m-1 = 3k+r+1$ (depending on which parity we are in).

We will get a payoff of $\frac{3k+r-t}{2(n+k)} = \frac{k+m-2-t}{2(n+k)}$ or $\frac{3k+r+1-t}{2(n+k)} = \frac{k+m-1-t}{2(n+k)}$ (depending on parity). Then it will have to move to node 1 arriving at $2k+2+m$ (and the game will be over)

Now consider moving to 1 and waiting, arrive at time $t+k+2 = 2k+2+t_e$. Then the payoff is $\frac{\min(2(k+1), k+m-t-1)}{2(n+k)}$. Which of these is chosen as the minimum will be decided by whether they still arrive in time to collect the first few attacks (i.e it depends on the second part $k+m-t-1 = 2k+r-t_e+1$ which means if its greater than $2k+2$ i.e depending on the distant between $r-1-t_e$)

More explicitly the term $k+m-t-1 \geq 2k+2$ if $r-1-t_e \geq 0$, in this case however it is possible to make up the difference of $r-1-t-e$ by only waiting till $2k+1$ (we will be there at $t+k+2 \geq 2(k+1)$, so will leave immediately back) then returning to the $*$ at $3k+4$ this is in comparison to $k+m = 3k+2+r$ meaning that we get an additional payoff depending on r 's parity. If r is odd then we gain $\frac{1}{2(n+k)} + \frac{r-2}{2(n+k)} = \frac{r-1}{2(n+k)}$. If r is even then we gain $\frac{r-1}{2(n+k)}$.

So we will be on $\frac{2(k+1)+r-1}{2(n+k)} = \frac{m-1}{2(n+k)}$ which is better than $\frac{k+m-1-t}{2(n+k)}$ and hence moving to 1 and moving back at $2k+2$ (arriving back at $3k+4$) is better.

Similarly in the case of $2k+2 > k+m-t-1$ if $r-1-t_e < 0$, then we can still do better by moving off, as we hit here at time $2k+2+t_e$ (so all attacks that are catchable have been caught), so leave and get back to star at $*$ nodes at $3k+4+t_e$ in comparison to $k+m = 3k+2+r$ means an additional payoff depending on r 's parity with t_e . In either case we get $\frac{r-t_e-1}{2(n+k)}$. But as this is negative it really means that all the attacks have already ended here, as $*$ attacks end at $k+m = 3k+2+r$ and as we arrive at $3k+4+t_e$ (and $r-1-t_e < 0$). Hence no additional values can be given and an overall value of $\frac{k+m-t-1}{2(n+k)}$ is given for this case.

But in both given scenario's it is still better than waiting and playing round all the *'s to end the game.

Hence it is at least as good to follow this strategy rather than repeatedly move between * nodes. This means it is better than a single choice and therefore is the best thing to do in such a position.

Meaning the only option for starting at a * node is to wait until $t = k$, the first real decision and decided to move to 1 getting a payoff of either $\frac{1}{2(n+k)} + \frac{m-1}{2(n+k)} = \frac{m}{2(n+k)}$

If we start at 1, then we can choose when to leave and visit *, say leave at time q and arrive at * at $q + k + 2 \geq k$ as we know that from this position she will move back immediately (arriving back at $q + 2(k + 2)$ and hence will only get one of the nodes value of $\frac{1}{n+k}$, once back at 1 we will travel back to *, as all attacks here are over (arriving at $q + 3(k + 2)$). This type of strategy will get her a payoff of

$$\frac{q+1}{2(k+1)} \times \frac{k+1}{n+k} + \frac{1}{k+1} + \frac{2(k+1) - (q+1)}{2(k+1)} \times \frac{k+1}{n+k} + \frac{(r-q-3)_+}{2(n+k)} = \frac{k+2 + (r-q-3)_+}{n+k}$$

.

The other choice is not to visit in the middle and get all the attacks at 1, then move across, suggest $q = 2k + 1$, now there is an opportunity to move to capture * still occuring at $k + m = 3k + 2 + r$ when we arrive at $3k + 3$. (The complete sum depends on the parity of r) We will get $\frac{r}{2(n+k)}$. Meaning the overall payoff for playing this strategy is $\frac{k+1}{n+k} + \frac{r}{2(n+k)} = \frac{m}{2(n+k)}$.

In the case of $r - q - 3 \geq 0$ then we are comparing $\frac{k+r-q-1}{2(n+k)} < \frac{m}{2(n+k)}$

In the case of $r - q - 3 < 0$ then we are comparing $\frac{k+2}{2(n+k)} < \frac{m}{2(n+k)} = \frac{2k+2+r}{2(n+k)}$.

Hence the available strategy to her if starting at 1 is to wait until $2k + 1$ and move to the * nodes claiming $\frac{m}{2(n+k)}$. If starting at * has to wait until k then move to 1 and then back to * claiming a payoff of $\frac{m}{2(n+k)}$.

Hence the best she can do in this situation is to follow one of the strategies. \square

It would be 'recommended' to follow the wait at 1 until time $2k + 1$ and then move as it is also optimal for all cases of m , whereas starting at * is only valid for $m > 2(k + 1)$.

Altered proof below

Proof. We shall first consider the case of $m \leq 2(k + 1)$. Consider starting at a * node then the options for any time $t < k$ is wait (one time period and reconsider moving then) or move to another * node or move to node 1.

Now immediately we can remove moving to another * node as this is dominated

by just waiting for two time periods. Now If waiting is the dominant strategy then we must continue to wait until time k upon which attacks at $*$ nodes begin.

Now consider being at a $*$ node for a time $k \leq t \leq k + m$ (when attacks are commencing), then the options are to move to another $*$ node claiming some benefit (if $t < k + m - 1$, otherwise no point in moving and infact the game is over at this point) or moving to 1 (arriving at $t + k + 2$) and catching some attacks there hopefully (if $t < k + m - 1$, otherwise no point in moving and infact the game is over).

Note. The special case of $t = k$ in which she can wait for one time period will be covered later

Now consider that if moving to another $*$ node dominates moving to 1 then it will be done for all time (as the generated payoff is the same, for all time, apart from possibly, from time $t = k + m - 2$ in which the generated payoff either way will be $\frac{1}{2(n+k)}$).

Now moving around the star nodes from time t until time $k + m - 1$ or $k + m$ (depending on the parity of these values i.e $t = 6$ $k + m - 1 = 10$ (even parity) or $t = 7$ $k + m = 11$ (odd parity)) gives us a payoff of exactly $\frac{1}{2(n+k)}$ per unit time, or more concretely

$$\frac{k + m - 1 - t}{2} \times \frac{1}{n + k} = \frac{k + m - 1 - t}{2(n + k)}$$

or

$$\frac{k + m - 2 - t}{2} \times \frac{1}{n + k} + \frac{1}{2(n + k)} = \frac{k + m - 1 - t}{2(n + k)}$$

In either case the same value is given (note. the initial payoff for being at a $*$ node at time t is not counted here)

The other decision to move to 1 and then make a decision, means arriving at 1 at time $t + k + 2 \geq 2k + 2 = 2(k + 1)$ meaning all attacks have already begun here (and infact it is impossible to decide to move back as return to $*$ at $t + 2(k + 2) \geq 3k + 4 > k + m$). This means that a payoff of

$$\frac{2k + m - (t + k + 2) + 1}{2(k + 1)} \times \frac{k + 1}{n + k} = \frac{k + m - 1 - t}{2(n + k)}$$

. Hence it is easy to see that it is not strictly dominating the alternative strategy.

For $t = k$ we can perform the addition strategy of wait one time period to gain $\frac{1}{2(n+k)}$ then we can decide to move to 1 getting $\frac{k+m-1-(k+1)}{2(n+k)}$, getting in total $\frac{m-1}{2(n+k)} = \frac{k+m-t-1}{2(n+k)}$, the same as above.

Now for starting at $*$ if it is before k then considering moving to 1 is dominated by just starting at 1, so the only option is to wait until k and then move to 1 (or move around 1 nodes, but they may not be enough) giving a total payoff of $\frac{m}{2(n+k)}$.

Now consider starting at node 1, we could consider waiting forever and getting $\frac{k+1}{2(n+k)}$, or we could consider moving to $*$ at some point in time say q .

Now doing the later will mean we arrive at $q + k + 2 \geq k$ (now assuming $q + k + 2 \leq k + m - 1$ i.e $q \leq m - 3$) then we will catch something as following the strategy for being at $*$ after time k . Meaning we get

$$\frac{q+1}{2(k+1)} \times \frac{k+1}{n+k} + \frac{1}{n+k} + \frac{k+m-1-(q+k+2)}{2(n+k)} = \frac{m}{2(n+k)}$$

(or worse if we don't arrive in time to catch things).

As $m \leq 2(k+1)$ it is clear that the option to wait at 1 and catch all attacks is better giving us in this case $\frac{k+1}{n+k}$.

Now for the case of $m > 2(k+1)$, let $m = 2(k+1) + r$ (where $r \geq 1$) \square

Lemma D.1. *The payoff for being at a $*$ node at a time $k \leq t \leq k + m - 1$ is*

$$\frac{(k+m-t-1)_+}{2(n+k)}$$

As long as

Note. This payoff does not the intial payoff for being at $*$ at t , only future decisions

Proof. We will first cover $t \geq k+1$ (and later cover $t = k$ as a special case). Now the options are to go to another star arriving at $t+2$ (then remake a decision) or to move to 1.

Let us first look at moving to 1, then we arrive at 1 at time $t+k+2$ (and as $t \geq k$, it means all attacks occurring at 1 have begun) so we claim a payoff of

$$\frac{\min\{2(k+1) - x, (2k+m-(t+k+2)+1-x)\}_+}{2(k+1)} \times \frac{k+1}{n+k} = \frac{\min\{2(k+1), (k+m-t-1-x)\}_+}{2(n+k)}$$

Where $x \geq 0$ is the overlap with attacks already caught.

Now choosing to move to $s *$ nodes before moving to 1 gives a payoff of

$$s \times \frac{1}{n+k} + \frac{\min\{2(k+1) - (x)_+, (k+m-t-2s-1-(x-2s)_+)\}_+}{2(n+k)}$$

.

So it is best to move to all the $*$ nodes, which haven't been visited before, before moving to 1.

In the case of $m \leq 2(k+1)$, it is impossible to get any overlap as the time to leave and return is at least $2(k+2) > m$, so in this case $x = 0$. Also $(k+m-t-1)_+ \leq m-1 < 2(k+1)$, so the payoff from moving to 1 immediately becomes

$$\frac{k+m-t-1}{2(n+k)}$$

Similarly the other case becomes

$$s \times \frac{1}{n+k} + \frac{k+m-t-1-2s}{2(n+k)} = \frac{k+m-1-t}{2(n+k)}$$

In the case of $m > 2(k+1)$ (let $m = 2(k+1) + r$), overlap is definitely possible if 1 has been visited before. Now the game ends if she is at a $*$ at any time past $k+m = 3k+2+r$ or at 1 at any time past $2k+m = 4k+2+r$.

If 1 hasn't been visited before then $x = 0 \left((k+m-t-2s-1 - (x-2s)_+)_+ = (k+m-t-2s-1)_+ = (3k+1+r-t-2s)_+ \leq (2k+1+r-2s)_+ \right)$, the pay-off becomes either

$$s \times \frac{1}{n+k} + \frac{k+1}{n+k} = \frac{k+s+1}{n+k}$$

if $3k+1+r-t-2s > 2k+2$ (i.e $k-1+r-t-2s > 0$, so $2s < k+r-1-t$) or

$$s \times \frac{1}{n+k} + \frac{3k+1+r-t-2s}{2(n+k)} = \frac{k+m-t-1}{2(n+k)}$$

otherwise

Examining the first part gives us

$$\frac{k+s+1}{n+k} = \frac{2(k+1)+2s}{2(n+k)} < \frac{2(k+1)+k+r-1-t}{2(n+k)} < \frac{k+m-1-t}{2(n+k)}$$

. We will end at 1 at time $t+2s+k+2$, meaning we might as well choose $s = 0$ and arrive as early as possible getting $\frac{k+m-t-1}{2(n+k)}$

If there is some overlap then we will get the above if $x-2s \leq 0$ (except the first part will be even worse with $-x$ in the numerator).

If $x > 2s$ though then we will get either

$$s \times \frac{1}{n+k} + \frac{2(k+1)-x}{2(n+k)} = \frac{2(k+1)+2s-x}{n+k}$$

if $3k+1+r-t-2s-(x-2s) > 2k+2$ (i.e $k+1+r-t-x > 0$,

$$s \times \frac{1}{n+k} + \frac{3k+1-t-2s-(x-2s)}{2(n+k)} = \frac{3k+1-t-x+2s}{2(n+k)} = \frac{k+m-1-t-x+2s}{2(n+k)}$$

Hence in this case the best we can do is pick the highest s to try not to overlap as much.

In any case the best she can do from this position is $\frac{k+m-t-1}{2(n+k)}$ □

Proof. We will first deal with the case that $t \geq k + 1$. We will also first assume that there is no initial overlap of attacks at 1 caught, that is if we arrive at node 1 at $t + k + 2$ we will not be there at a time when attacks we previously caught would still be happening. Let the overlap be denoted by x , so first look at $x = 0$. Now our only choice from node $*$ is to move to s other $*$ nodes that we haven't yet visited and then move to 1 (in the hope of catching some attacks).

The payoff for doing this gives us

$$\begin{aligned} & s \times \frac{1}{n+k} + \frac{\min\{2(k+1), (2k+m-(t+2s+k+2)+1-x)_+\}}{2(k+1)} \times \frac{k+1}{n+k} \\ &= \frac{2s}{(n+k)} + \frac{\min\{2(k+1), (k+m-t-1-2s-x)_+\}}{2(n+k)} \end{aligned}$$

and as $x = 0$

$$\frac{2s}{2(n+k)} + \frac{\min\{2(k+1), (k+m-t-1-2s)_+\}}{2(n+k)}$$

From this it should be clear that the payoff is non-decreasing in s and so choosing s as the maximum would seem to be a logical choice.

Now for a moment we will consider the future when we are at 1, as we will arrive at time $t+k+2+2s \geq 2k+2+2s \geq 2k+2$ all attacks occurring at 1 have been and we should no longer consider waiting or returning to this node, also moving away brings us back to $*$ nodes (arriving at time $t+2(k+2)+2s \geq 2k+4+2s$, when the attacks end at $k+m-1$ or $k+m$), now as before all attacks have begun but they may not have ended. Hence we could consider moving around these $*$ nodes until the game ends. This means we actually get a payoff of

$$\begin{aligned} & \frac{2s}{2(n+k)} + \frac{\min\{2(k+1), (k+m-t-1-2s)_+\}}{2(n+k)} + \frac{\left(\frac{k+m-1-(t+2(k+2)+2s)}{2}\right)_+}{n+k} \\ &= \frac{2s}{2(n+k)} + \frac{\min\{2(k+1), (k+m-t-1-2s)_+\}}{2(n+k)} + \frac{(m-k-5-t-2s)_+}{2(n+k)} \end{aligned}$$

or

$$\begin{aligned} & \frac{2s}{2(n+k)} + \frac{\min\{2(k+1), (k+m-t-1-2s)_+\}}{2(n+k)} + \frac{\left(\frac{k+m-2-(t+2(k+2)+2s)}{2}\right)_+}{n+k} + \frac{1}{2(n+k)} \\ &= \frac{2s}{2(n+k)} + \frac{\min\{2(k+1), (k+m-t-1-2s)_+\}}{2(n+k)} + \frac{(m-k-5-t-2s)_+}{2(n+k)} \end{aligned}$$

So we only need to worry about this is $m-k-5-t-2s > 0$. If $m = 2(k+1)+r$ then $m-k-5-t-2s = 2(k+1)+r-k-5-t-2s = k+r-3-t-2s$ So if $k+r-3-t-2s > 0$ then we will worry about this possibility of doing the movement at the ends. However if $k+r-3-t-2s > 0 \implies 3k+1+r-t-2s > 2(k+1) \implies k+m-t-1-2s > 2(k+1)$, meaning that the payoff actually becomes

$$\frac{2s}{2(n+k)} + \frac{2(k+1)}{2(n+k)} + \frac{m-k-5-t-2s}{2(n+k)} = \frac{k+m-t-3}{2(n+k)}$$

So in this case the choice of s is irrelevant, Hence we might as pick s to be the highest possible, call it $s_{max} = \min\{n - 1 - y, \frac{k+m-1-t}{2}\}$ (if odd parity) or $s_{max} = \min\{n - 1 - y, \frac{k+m-t}{2}\}$ (if even parity)(Note. In even parity we will get the starting payoff slightly differently). Where y is the number of currently visited * nodes at time t .

For each type of parity let us cover the two cases

Odd Parity: 1. Let $n - 1 - y \geq \frac{k+m-1-t}{2}$ then the payoff we get becomes

$$\begin{aligned} & \frac{\frac{k+m-1-t}{2}}{n+k} + \frac{\min\{2(k+1), (k+m-t-2 \times \frac{k+m-1-t}{2} - 1)_+\}}{2(n+k)} \\ &= \frac{k+m-1-t}{2(n+k)} + \frac{\min\{2(k+1), 0\}}{2(n+k)} = \frac{k+m-1-t}{2(n+k)} \end{aligned}$$

2. Let $n - 1 - y < \frac{k+m-1-t}{2}$ then the payoff we get becomes

$$\frac{n-1-y}{n+k} + \frac{\min\{2(k+1), (k+m-t-2(n-1-y)-1)_+\}}{2(n+k)}$$

Further split into subcases

a) Let $k+m-t-2(n-1-y)-1 < 0$ then the payoff becomes

$$\frac{n-1-y}{n+k} < \frac{k+m-t-1}{2(n+k)}$$

b) Let $0 \leq k+m-t-2(n-1-y)-1 \leq 2(k+1)$ then the payoff becomes

$$\frac{n-1-y}{n+k} + \frac{k+m-t-2(n-1-y)-1}{2(n+k)} = \frac{k+m-t-1}{2(n+k)}$$

c) Let $k+m-t-2(n-1-y)-1 > 2(k+1)$ then the payoff becomes

$$\frac{n-1-y}{n+k} + \frac{2(k+1)}{2(n+k)} = \frac{n+k-y}{n+k} < \frac{k+m-t-1}{2(n+k)}$$

$$\text{As } k+m-t-2(n-1-y)-1 > 2(k+1) \implies k+m-t-1 > 2(n-1-y+k+1) = 2(n+k-y)$$

Even Parity: 1. Let $n - 1 - y \geq \frac{k+m-t}{2}$ then the payoff we get becomes

$$\begin{aligned} & \frac{\frac{k+m-t}{2} - 1}{n+k} + \frac{1}{2(n+k)} + \frac{\min\{2(k+1), (k+m-t-2 \times \frac{k+m-t}{2} - 1)_+\}}{2(n+k)} \\ &= \frac{k+m-1-t}{2(n+k)} + \frac{\min\{2(k+1), (-1)_+\}}{2(n+k)} = \frac{k+m-1-t}{2(n+k)} \end{aligned}$$

2. Let $n - 1 - y < \frac{k+m-t}{2}$ then the payoff we get becomes

$$\frac{n-1-y}{n+k} + \frac{\min\{2(k+1), (k+m-t-2(n-1-y)-1)_+\}}{2(n+k)}$$

Note. In this case, the ‘problem’ with having a final time only pick up $\frac{1}{2(n+k)}$ is not possible as we will have left before this time. Further split into subcases

a) Let $k + m - t - 2(n - 1 - y) - 1 < 0$ then the payoff becomes

$$\frac{n - 1 - y}{n + k} < \frac{k + m - t - 1}{2(n + k)}$$

b) Let $0 \leq k + m - t - 2(n - 1 - y) - 1 \leq 2(k + 1)$ then the payoff becomes

$$\frac{n - 1 - y}{n + k} + \frac{k + m - t - 2(n - 1 - y) - 1}{2(n + k)} = \frac{k + m - t - 1}{2(n + k)}$$

c) Let $k + m - t - 2(n - 1 - y) - 1 > 2(k + 1)$ then the payoff becomes

$$\frac{n - 1 - y}{n + k} + \frac{2(k + 1)}{2(n + k)} = \frac{n + k - y}{n + k} < \frac{k + m - t - 1}{2(n + k)}$$

$$\text{As } k + m - t - 2(n - 1 - y) - 1 > 2(k + 1) \implies k + m - t - 1 > 2(n - 1 - y + k + 1) = 2(n + k - y)$$

Now let us consider that there is some initial overlap, being more concretely $x = x(s) = (x(0) - 2s)_+ = (x_0 - 2s)_+$. Then if $x - 2s \leq 0$ we get the same as above. Otherwise if $x - 2s > 0$ then the payoff is

$$\begin{aligned} & \frac{s}{n + k} + \frac{\min\{2(k + 1), (k + m - t - 1 - 2s - (x_0 - 2s))_+\}}{2(n + k)} \\ &= \frac{s}{n + k} + \frac{\min\{2(k + 1), (k + m - t - 1 - x_0)_+\}}{2(n + k)} \end{aligned}$$

And again the payoff is non-decreasing in s , so again s_{max} is the best choice. We can follow a similar idea to the above when there was no overlap, though has the chance of being lower. Hence it will only harm the values

The exact algebra follows the same process expect now the second term in the visiting of node 1 has a subtraction and is therefore less rewarding.

Hence from this position the best the patroller can do is get a payoff of $\frac{k+m-t-1}{2(n+k)}$ and the best patroller strategy is to travel amongst as many $*$ nodes that have not already been visited and then head off to 1.

Now consider the special case of being at a $*$ at $t = k$, then the above can be followed to achieve no better than $\frac{m-1}{2(n+k)}$ or we can wait and pick up $\frac{1}{2(n+k)}$ by catching the second attack and then proceed to follow the above at $t = k + 1$ getting $\frac{m-2}{2(n+k)}$. Either strategy yields the same payoff, so we shall suggest just strictly following the above. \square

Theorem D.2 (Lower Bound).

$$V \leq \max \left\{ \frac{k + 1}{n + k}, \frac{m}{2(n + k)} \right\}$$

Proof. Suppose the Patroller starts at a $*$ node at time $t \leq k - 1$, then the options are to move to node 1 which is dominated by just starting at node 1

and waiting till $t + k + 2$ (as no attacks are picked up at $*$ nodes. Or she can wait till time t and follow the above lemma getting a payoff of

$$\frac{1}{2(n+k)} + \frac{m-1}{2(n+k)} = \frac{m}{2(n+k)}$$

. So starting at $*$ provides a payoff of at most $\frac{m}{2(n+k)}$. Now consider starting at node 1 at time 0 then if we leave and want to return we can only do so once (as upon the first return the time will be at least $2(k+2)$ so all attacks will have begun).

Now split the solution into two regions for $m = 2(k+1) + r$, for $-2(k+1) + 1 \leq r \leq 0$ and $r \geq 1$. Consider waiting until time q and then leaving, as you will hit a $*$ node at time $q + k + 2$ meaning we can use the lemma to generate the max payoff here. For a total payoff of

1. If $q + k + 2 > k + m = 3k + 2 + r$

$$\begin{aligned} & \frac{q+1}{2(k+1)} \times \frac{k+1}{n+k} + \frac{(k+m-(q+k+2)-1)_+}{2(n+k)} \\ &= \frac{q+1}{2(n+k)} \end{aligned}$$

2. If $q + k + 2 = k + m = 3k + 2 + r$

$$\begin{aligned} & \frac{q+1}{2(k+1)} \times \frac{k+1}{n+k} + \frac{1}{2(n+k)} + \frac{(k+m-(q+k+2)-1)_+}{2(n+k)} \\ &= \frac{q+2}{2(n+k)} \end{aligned}$$

3. If $q + k + 2 \leq k + m = 3k + 2 + r$

$$\begin{aligned} & \frac{q+1}{2(k+1)} \times \frac{k+1}{n+k} + \frac{1}{n+k} + \frac{(k+m-(q+k+2)-1)_+}{2(n+k)} \\ &= \frac{m}{2(n+k)} \end{aligned}$$

So as these are non-decreasing in q we pick $q = 2k + 1$. Then we get

1. If $3k + 3 > 3k + 2 + r$ i.e $r \leq 0$

$$\frac{2k+1+1}{2(n+k)} = \frac{k+1}{n+k}$$

2. If $3k + 3 = 3k + 2 + r$ i.e $r = 1$

$$\frac{2k+1+2}{2(n+k)} = \frac{2(k+1)+1}{2(n+k)} = \frac{m}{2(n+k)}$$

3. If $q + k + 2 \leq k + m = 3k + 2 + r$ i.e $r \geq 2$

$$\frac{m}{2(n+k)}$$

Hence if $m \leq 2(k+1)$ (i.e $r \leq 0$) then we get $\frac{k+1}{n+k}$ and if $m > 2(k+1)$ (i.e $r \geq 1$) we get $\frac{m}{2(n+k)}$ and hence the result. \square

D.6 Type-delayed attack proof

We begin with a lemma used in the proof

Lemma D.3. *When $k_{max} - i \leq t \leq k_{max} + i$ we get that*

$$\frac{(k_{max} + 1 + i - t)_+}{2 \binom{n + \sum_{j=1}^h k_j}{}} + \frac{(m - 2(i + 1))_+}{2 \binom{n + \sum_{j=1}^h k_j}{}} \geq \frac{(k_{max} - i + m - t - 1)_+}{2 \binom{n + \sum_{j=1}^h k_j}{}}$$

Proof. If $m \geq 2(i + 1)$, then we get

$$\frac{k_{max} + 1 + i - t + m - 2(i - 1)}{2 \binom{n + \sum_{j=1}^h k_j}{}} = \frac{k_{max} - i + m - t - 1}{2 \binom{n + \sum_{j=1}^h k_j}{}} \geq \frac{(k_{max} - i + m - t - 1)_+}{2 \binom{n + \sum_{j=1}^h k_j}{}}$$

If $m \leq 2(i + 1)$, then we get

$$\frac{k_{max} + 1 + i - t}{2 \binom{n + \sum_{j=1}^h k_j}{}} \geq \frac{(k_{max} + 1 + i - t + m - 2(i + 1))_+}{2 \binom{n + \sum_{j=1}^h k_j}{}} = \frac{(k_{max} - i + m - t - 1)_+}{2 \binom{n + \sum_{j=1}^h k_j}{}}$$

\square

The main proof of the attack lemma's future payoff

Note. The difference in future payoff depends on whether the patroller has the option to wait (till the last attack at the type i node) or whether they have no option but to move.

Remark. It is worth noting that the game essentially ends at time $t = 2k_{max} + m$, so this is really the last time we could possibly care about. Further as a type i node attacks finish at $t = k_{max} + i + m$, and so a movement to a type j node means arriving at $t = k_{max} + 2i + 2 + j + m \geq k_{max} + j + m \forall j$, meaning that no future attacks can be caught if at a type i node at $t = k_{max} + i + m$, and this means this is the artificial end for the game at a type i node.

Lemma D.4. *When the Type-delayed attack is possible, then the future payoff*

for being at a type i node at time t is given by:

$$\frac{(k_{max} + 1 + i - t)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} + \frac{(m - 2(i + 1))_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \text{ if } k_{max} - i \leq t \leq k_{max} + i$$

$$\frac{(k_{max} - i + m - t - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \text{ if } t \geq k_{max} + i + 1$$

Proof. We will first prove the statement for when $t \geq k_{max} + i + 1$, by the use of strong backwards induction.

Base Case:

For $t = k_{max} + i + m$, as above it is known that no future attacks can be caught (and the game is over) as arriving at a type j means arriving at $k_{max} + 2i + j + 2 + m \geq k_{max} + j + m$ meaning all attacks are over ($\forall j$). Hence the future payoff is 0. Now the formula gives

$$\frac{(k_{max} - i + m - (k_{max} + i + m) - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{(-2i - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} = 0$$

Hence it is true for $t = k_{max} + i + m$.

Induction hypothesis:

For some $t_1 \geq k_{max} + i + 1$ assume that the formula for $t \geq k_{max} + i + 1$ is true for all $t \geq t_1 + 1$.

Induction step:

At a type i node at time $t_1 \geq k_{max} + i + 1$, all attacks have begun at this node so waiting is not really an option (i.e it is clearly dominated by moving immediately). So moving to a type j means arriving at $t_1 + i + 2 + j$ giving an immediate payoff and meaning (under the Induction hypothesis) a future payoff is received. Giving at best a total payoff of

$$\underbrace{\frac{\min(2(j + 1), (k_{max} + j + m - (t_1 + i + 2 + j) + 1)_+)}{2(j + 1)} \times \frac{j + 1}{n + \sum_{j=1}^h k_j}}_{\text{Immediate reward at type } j}$$

$$+ \underbrace{\frac{(k_{max} - j + m - (t_1 + i + 2 + j) - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)}}_{\text{Future reward from type } j}$$

$$= \frac{\min(2(j + 1), (k_{max} + m - t_1 - i - 1)_+)}{2 \left(n + \sum_{j=1}^h k_j \right)} + \frac{(k_{max} - i + m - t_1 - 2j - 3)_+}{2 \left(n + \sum_{j=1}^h k_j \right)}$$

Now we will split it into the subcase of $2(j+1) \geq k_{max} + m - t_1 - i - 1$. This means $k_{max} + m - t_1 - 3 - 2j \leq 0$, then the payoff becomes

$$\frac{(k_{max} + m - t_1 - i - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)}$$

Otherwise if $2(j+1) < k_{max} + m - t_1 - i - 1$. This means $k_{max} + m - t_1 - i - 3 - 2j > 0$, then the payoff becomes

$$\begin{aligned} & \frac{2(j+1) + k_{max} - i + m - t_1 - 2j - 3}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{k_{max} - i + m - t_1 - 1}{2 \left(n + \sum_{j=1}^h k_j \right)} \\ & = \frac{(k_{max} - i + m - t_1 - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \end{aligned}$$

Hence by mathematical induction it is true for $t \geq k_{max} + i + 1$ and the form is correct.

Next we seek to prove the formula for the values of $t \leq k_{max} + i$, again using strong backwards induction.

Base case:

For $t = k_{max} + i$, we have the option to wait for one unit of time (to catch the final attack at this node) and then claim the reward for this and the future rewards, or can move immediately to some type j node (arriving at $k_{max} + 2i + 2 + j$ and claim the reward and future rewards.

Waiting gives

$$\begin{aligned} & \frac{1}{2(i+1)} \times \frac{i+1}{n + \sum_{j=1}^h k_j} + \frac{(k_{max} - i + m - (k_{max} + i + 1) - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \\ & = \frac{1}{2 \left(n + \sum_{j=1}^h k_j \right)} + \frac{(m - 2(i+1))_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \end{aligned}$$

Moving gives

$$\begin{aligned}
& \frac{\min(2(j+1), (k_{max} + j + m - (k_{max} + 2i + 2 + j) + 1)_+)}{2 \left(n + \sum_{j=1}^h k_j \right)} \\
& + \frac{(k_{max} - j + m - (k_{max} + 2i + 2 + j) - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \\
& = \frac{\min(2(j+1), (m - 2i - 1)_+) + (m - 2i - 2j - 3)_+}{2 \left(n + \sum_{j=1}^h k_j \right)}
\end{aligned}$$

When moving it will depend on if $2(j+1) \geq m - 2i - 1$, which means $m - 2i - 2j - 3 \leq 0$ giving $\frac{(m-2i-1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)}$

Or if $2(j+1) < m - 2i - 1$, meaning $m - 2i - 2j - 3 > 0$ gives $\frac{2(j+1)+m-2i-2j-3}{2 \left(n + \sum_{j=1}^h k_j \right)} =$

$$\frac{m-2i-1}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{1}{2 \left(n + \sum_{j=1}^h k_j \right)} + \frac{(m-2(i+1))_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \quad (\text{Note. Here } m - 2(i+1) > 1)$$

Hence with either option the best that the patroller can do is $\frac{1}{2 \left(n + \sum_{j=1}^h k_j \right)} +$

$\frac{(m-2(i+1))_+}{2 \left(n + \sum_{j=1}^h k_j \right)}$, which is the correct form of the formula.

Induction hypothesis:

Assume that the formula for $k_{max} - i \leq t \leq k_{max} + i$ is true for all $t \geq t_1 + 1$.

Induction step:

At a type i node at time $k_{max} - i \leq t_1 \leq k_{max} + i - 1$, the options is to wait for some period of time, say q periods, and then move to a type j node and claim future rewards (which will depends on which case the arrival time falls into), this decision means we will arrive at the type j node at $t_1 + q + i + 2 + j \geq k_{max} + q + 2 + j$ meaning we will always fall into the second category for future rewards.

This means the future payoff for such a decision will be

$$\begin{aligned}
& \frac{q}{2(i+1)} \times \frac{i+1}{n + \sum_{j=1}^h k_j} + \frac{\min(m, 2(j+1), (k_{max} + j + m - (t_1 + q + i + 2 + j) + 1)_+)}{2(j+1)} \times \frac{j+1}{n + \sum_{j=1}^h k_j} \\
& + \frac{(k_{max} - i + m - (t_1 + i + 2 + j) - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \\
& = \frac{q + \min(2(j+1), (k_{max} - i + m - t_1 - 1 - q)_+) + (k_{max} - j + m - (t_1 + q + i + 2 + j) - 1)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \\
& = \frac{q + \min(m, 2(j+1), (k_{max} - i + m - t_1 - 1 - q)_+) + (k_{max} - i - 2j + m - t_1 - 3 - q)_+}{2 \left(n + \sum_{j=1}^h k_j \right)} \\
& = \frac{q + \min(m, 2(j+1), (k_{max} - i + m - t_1 - 1 - q)_+) + (k_{max} - i + m - t_1 - 1 - q - 2(j+1))_+}{2 \left(n + \sum_{j=1}^h k_j \right)}
\end{aligned}$$

Now if $m \geq 2(j+1) \geq k_{max} - i + m - t_1 - 1 - q \geq 0$ then we get

$$\frac{q + k_{max} - i + m - t_1 - 1 - q}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{k_{max} - i + m - t_1 - 1}{2 \left(n + \sum_{j=1}^h k_j \right)}$$

If $m \geq 2(j+1) \geq k_{max} - i + m - t_1 - 1 - q$ and $k_{max} - i + m - t_1 - 1 - q \leq 0$ then we get $\frac{q}{2 \left(n + \sum_{j=1}^h k_j \right)}$

If $m \geq k_{max} - i + m - t_1 - 1 - q \geq 2(j+1)$ then we get $\frac{q + 2(j+1) + k_{max} - i + m - t_1 - 1 - q - 2(j+1)}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{k_{max} - i + m - t_1 - 1}{2 \left(n + \sum_{j=1}^h k_j \right)}$

If $k_{max} - i + m - t_1 - 1 - q \geq m \geq 2(j+1)$ then we get $\frac{q + 2(j+1) + k_{max} - i + m - t_1 - 1 - q - 2(j+1)}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{k_{max} - i + m - t_1 - 1}{2 \left(n + \sum_{j=1}^h k_j \right)}$

If $k_{max} - i + m - t_1 - 1 - q \geq 2(j+1) \geq m$ then we get $\frac{q + m + k_{max} - i + m - t_1 - 1 - q - 2(j+1)}{2 \left(n + \sum_{j=1}^h k_j \right)} \leq \frac{q + 2(j+1) + k_{max} - i + m - t_1 - 1 - q - 2(j+1)}{2 \left(n + \sum_{j=1}^h k_j \right)} = \frac{k_{max} - i + m - t_1 - 1}{2 \left(n + \sum_{j=1}^h k_j \right)}$

In every case we know that the value is less than $\frac{(k_{max}+1+i-t)_+}{2\left(n+\sum_{j=1}^h k_j\right)} + \frac{(m-2(i+1))_+}{2\left(n+\sum_{j=1}^h k_j\right)}$.

Hence by mathematical induction it is true for $k_{max} - i \leq t \leq k_{max} + i$ and the form is correct.

□

The theorem's proof is now easy to follow

Proof. Starting at a type i node means waiting till time $t = k_{max} - i$ and then receiving a payoff from the previous lemma. This should be clear as for any time, $s < k_{max} - i$, the decision is to wait or move to another type node and as no attacks can be claimed by waiting if moving was better then the patroller might as well start there. Hence starting at type i provides a payoff of

$$\frac{1}{2(i+1)} \times \frac{i+1}{n+\sum_{j=1}^h k_j} + \frac{2i+1}{2\left(n+\sum_{j=1}^h k_j\right)} + \frac{(m-2(i+1))_+}{2\left(n+\sum_{j=1}^h k_j\right)} = \frac{i+1}{n+\sum_{j=1}^h k_j} + \frac{(m-2(i+1))_+}{2\left(n+\sum_{j=1}^h k_j\right)}$$

So if $m \geq 2(i+1)$ then we get $\frac{m}{2\left(n+\sum_{j=1}^h k_j\right)}$ and if $m < 2(i+1)$ then we get $\frac{i+1}{n+\sum_{j=1}^h k_j}$. Hence if $m \geq 2(k_{max}+1)$ then we get $\frac{m}{2\left(n+\sum_{j=1}^h k_j\right)}$ and if $m < 2(k_{max})$ we get $\frac{k+1}{n+\sum_{j=1}^h k_j}$. □

D.7 Stage-1-Step-point lemma proof

We will get both bounds, first the patrollers bound is simple by applying the combinatorial strategy

Proof. We will first note that $m = 2k + 2 - 4i \leq 2(n-1)$ satisfies the condition for the improvement to be possible. We are in the case with $M \neq \emptyset$ and $R = \emptyset$ and $m = 2k + 2 - 4i \implies m(1 + \frac{n-1}{2}) \leq 2(n+k) \iff 2k - 4i \leq 2k \iff i \geq 0$.

So we get the bound $V \geq \frac{2(2k+2-4i)}{2(n+k)+(2k+2-4i)(1+\frac{n-1}{k+1-3i})} = \frac{2(2k+2-4i)}{2n+2k+2k+2-4i+2(n-1)} = \frac{m}{2(n+k-i)}$ as required □

Now we will propose the 'simplification' to $S_{n+1}^{k-2i, i-1}$

Proof. The simplification mapping is:

Now we note that at $m = 2k + 2 - 4i \geq 2(k_{max} + 1)$ as $2(k_{max} + 1)$ is either $2(k - 2i + 1) = 2k + 2 - 4i$ or $2(i - 1 + 1) = 2i \leq 2k + 2 - 4i$ (as $i \leq \lfloor \frac{k+1}{3} \rfloor \implies 6i \leq 2k + 2$).

This gives a bound $V \leq V(S_{n+1}) = \frac{m}{2(n+1+k-2i+i-1)} = \frac{m}{2(n+k-i)}$ \square