Second Year Report

Thomas Lowbridge,
School of Mathematical Sciences,
University of Nottingham.

July 26, 2018

Contents

1 Patrolling games

This chapter analyses patrolling games played between an immobile attacker, who is only present in the network for a period of time, and a mobile patroller on a graph, $Q = (N, E)$. The attacker picks a node, i in Q, and a starting time, τ, forming a strategy, (i, τ) called an attack. Their attack lasts m time periods, called the attack time. Choosing a starting time to attack is equivalent to the attacker choosing an attacker interval, $I = \{\tau, ..., \tau + m - 1\}$. The patroller picks a walk, $W(t)$ in Q, with the choice to start at any node at time 0. The payoff of this win-lose game (to a maximizing patroller) is 1 if the attacker is caught and 0 if the attacker is not caught.

If the time horizon, $\mathcal{T} = \{0, 1, ..., T - 1\}$, is finite then both players have a finite collection of pure strategies. However if it is infinite then both players have an infinite collection of pure strategies. In either case, by the usual application of the minimax theorem [Ref], the value of the game exists and we call it $V(Q)$, the *patrol value* of Q (or of Q,T,m). In general, both players will require the use of mixed strategies and the patrol value of Q can be seen as the optimal probability of capture.

For an attack to succeed, the attacker needs to have completed their attack by the end of the game and hence, $tau \leq T - m$ (or equivalently $I \subset T$). As other attacker strategies always fail, we can safely them as they are dominated.

These games were introduced in [Ref], and many tools were found along with complete solutions to Hamiltonian and Complete Bipartite. While only a partial solution to the Line graph was found for 'long' attack times, later completed in [Ref].

In section [Ref] we deal with the ambiguity of the time-horizon in achieving the *diametric upper bound*, $V \leq \frac{m}{\bar{d}}$, as in Lemma 9 in [Ref]. It is proposed to place potential attacks at a pair of nodes which are the furtherest apart (called diametric points) for all starting $\tau \leq T - m$, with each of these potential attacks being played with equal probability. However this mixed attack strategy achieves the diametric bound only under restrictions to the time-horizon. We provide a concrete statement about these restrictions and provide an alternative mixed attack strategy, the *time-limited diametric attack strategy* to reduce them. We also extend the time-limited diametric attack strategy to a larger class of attack strategies which uses multiple nodes which form regular polygons and later irregular polygons.

As both the star graph, S_n (Equivalent to $K_{1,n}$), and line graph, L_k, have been completely solved, we aim to incorporate both ideas into the elongated star graph, S_n^k in which we elongated one of the edges in S_n k times. We partially solve this graph, in the range $m \leq 2(k + 1)$ in section [Ref] by developing an extension from the time-limited attack strategy to the *time-delayed attack strategy.* This attack strategy incorporates the idea of distance from the stars centre by spreading out potential attacks in time.

The time-delayed attack strategy is further extended, as we deal with more general star graphs, where all edges are possibly elongated. We finally introduce the idea of connecting the centres of generalised star graphs to form more complex graphs, with some partial solutions found by node simplification and decomposition.

1.1 Issue and correction of the diametric bound

We now state Lemma 9 in [**?**] as the diametric bound, the attackers bound by attacking randomly throughout time at the diametric nodes.

Lemma 1.1 (Diametric bound). *$V \leq \max\left\{\frac{1}{2}, \frac{m}{2\bar{d}}\right\}$, guaranteed by the diametric attack.*

We will now look at an issue with the diametric bound when $\bar{d} < m \leq 2\bar{d}$, so we are dealing with the bound $V \leq \frac{m}{2\bar{d}}$. The result of the lemma states with ambiguity that this holds for large m and T, however the results are used for smaller T such as in Theorem 16 in [**?**]. We will analyse this issue of 'large' T, and present a more concrete conclusion on conditions for which T must take in order to use this bound. We will then present an altered diametric attack, which can further help to reduce the conditions on T for the same bound.

The diametric attack has the attacker start at two diametrical nodes with equal probability and then choosing a starting attack time, $\tau = 0, 1..., T-m$ again with equal probability. Hence it can be thought of as the attacker making $2(T-m+1)$ attacks each with equal probability. To see how the patroller can do against this attacking strategy, we can consider a simple oscillatory strategy, going back and forth between the diametrical nodes, and see how many potential attacks are captured.

Example 1.2 (Issue with the diametric bound)**.** Consider the game $(Q = L_5, T = 20, m = 6)$ so $m = 6 > \bar{d} = 4$, then under the diametric attack the attacker has 30 attacks, starting at $0, ..., 14$ at nodes 1 and 5. If the patroller oscillates between diametric points they capture $1+5+6+6+4 = 22$ attacks out of $2(20-6+1) = 30$ attacks, for a capture chance of $V = \frac{11}{15}$, which is greater than diametric bound $\frac{3}{4}$.

We could now consider delaying the patroller initially to get a better number of attacks, e.g. delaying in the counter example to leave at time 2 instead of 0 gives $3+6+6+6+2 = 23$. In fact it can be shown that it is best for the patroller to delay themselves till time $m - \bar{d} - 1$ and then perform the oscillation between the diametric nodes (see appendix B.1.

This means the patroller will capture, by oscillating,

$$\underbrace{m-\bar{d}}_{\text{Waiting initially}} \quad + \quad \underbrace{(m \times (M+1))_+}_{\text{Visits which get exactly } m \text{ attacks}}$$
$$+ \underbrace{\left(T - \left(m - 1 + (M+1)\,\bar{d}\right)\right)_+ + \left(T - \left(m - 1 + (M+2)\,\bar{d}\right)\right)_+}_{\text{Penultimate and final node visits}}$$

Where, $M = \left\lfloor \frac{T-2m+1}{\bar{d}} \right\rfloor$, is the number of nodes after the two initial visits getting exactly m of the potential attacks, an exact derivation of this formula is left to appendix B.1.

The key point is that the upper bound given by the diametric attack is dependent on T. An example is given in Figure 1.1 which shows that for some T as in Lemma 1.3 and as the finite time horizon becomes infinite the corrected bound reaches the suggested bound in [**?**].

Lemma 1.3 (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. Furthermore as $T \to \infty$ then the diametric bound holds and moreover the errors are maximal when $T = 2m - 1 + k\bar{d}$ for some $k \in \mathbb{N}_0$ with an error, inversely proportional to the time horizon, explicitly $\frac{m\bar{d}-(m-\bar{d})^2}{2\bar{d}(m+(k+1)\bar{d})} = \mathcal{O}(\frac{1}{T})$*

For proof see Appendix B.2.

Fixing the diametric attack

A possible 'fix' to the issue is to limit the time the attack take place in, only using the time up to a point such that the condition is met, i.e. $T = m - 1 + (k+1)\bar{d}$ for some $k \in \mathbb{N}_0$.

Definition 1.4 (Time-limited diametric attack)**.** Attacking at a pair of diametric nodes equiprobably for the times I,$I+1$...,$I+\bar{d}-1$ (i.e starting attacks at $\tau, \tau+1, ..., \tau+\bar{d}-1$) is called the *timed diametric attack.*

To make the time-limited diametric attack feasible, we need that the condition holds for some $k = 0, 1, 2, ...$ and that means holding for it holding for $k = 0$ will suffice, i.e. $T \geq m - 1 + \bar{d}$.

Lemma 1.5 (Time-limited diametric attack bound)**.** *When $T \geq m + \bar{d} - 1$, the diametric bound $V \leq \max\{\frac{1}{2}, \frac{m}{2\bar{d}}\}$ is valid.*

For proof see appendix B.3.

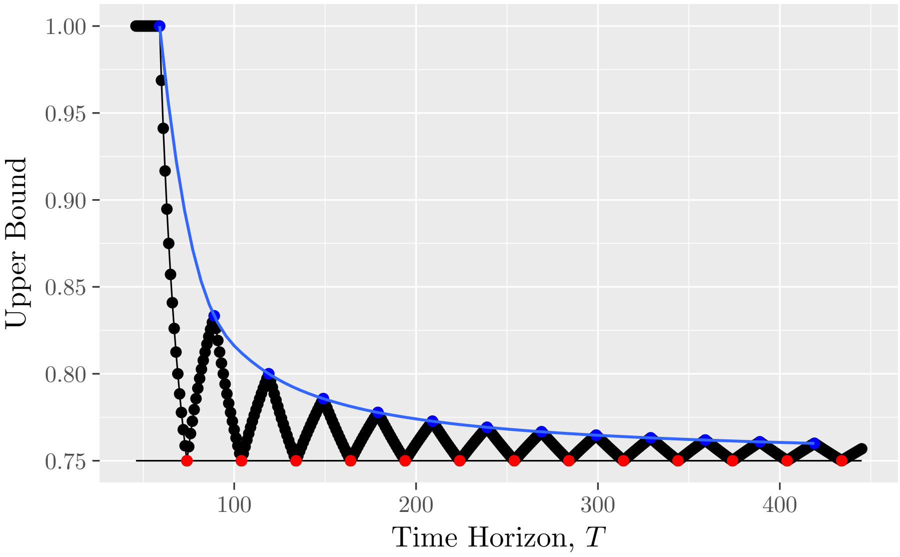


Figure 1.1: Best Upper Bound achievable under the diametric strategy

1.2 New Tools

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

Definition 1.6 (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.

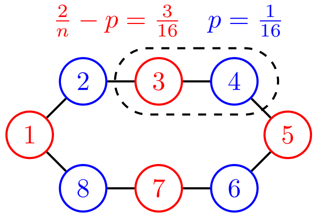


Figure 1.2: C_8 with the blue nodes being "even" nodes started at with probability $\frac{1}{16}$ and the red nodes being "odd" nodes started at with probability $\frac{3}{16}$.

Lemma 1.7. *When n and m are both even, following the ARHP 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 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, $\boldsymbol{\pi}_{ARHP}$, with probability p at "even" nodes and probability $\frac{2}{n} - p$ at "odd" nodes. Then

$$P(\boldsymbol{\pi}_{ARHP}, [i, I]) \geq \underbrace{\overbrace{p}^{\text{even node}} + \overbrace{\frac{2}{n} - p}^{\text{odd node}} + p + \frac{2}{n} - p + ... + p + \frac{2}{n} - p}_{m=2m' \text{ elements}}$$

$$= m'p + m'(\frac{2}{n} - p) = \frac{2m'}{n} = \frac{m}{n} \quad \forall i \in N \quad \forall I \subseteq \mathcal{T}$$

Hence as it holds for all pure attacks

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

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.

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 seen in Appendix B.6).

We now look at extending the idea of the diametric attack. First we notice that we are not forced to use the graph's diameter, $\bar{d}$, and in fact we can use any distance between two selected nodes, d. We replace $\bar{d}$ with d to get a two node distance attack, giving us a bound of $V \leq \max\{\frac{1}{2}, \frac{m}{2d}\}$. However this cannot be a better bound than using the graph's diameter, so this is not useful.

A useful extension, would be not using two nodes but using multiple points each the same distance apart from each other mutually.

Definition 1.8 (Polygonal attack)**.** A *d-polygonal attack* is an attack at a set of nodes D such that $d(i, i') = d \, \forall i, i' \in D$ at the time intervals $I, I+1, ..., I+d-1$, for some fixed time interval I.

We can also consider having the points at least the same distance apart, instead of the same distance apart. This construction is considered as it is less restricting on the graphical structure and still provides the same bound.

Definition 1.9 (Uneven polygonal attack)**.** A *d-uneven polygonal attack* is an attack at a set of nodes D such that $d(i, i') \geq d \, \forall i, i' \in D$ at the time intervals $I, I+1, ..., I+d-1$, for some fixed time interval I.

We need to consider the feasibility of these attacks, due to requiring d potential starting times, we need that $T \geq m - d + 1$. The idea behind this attack is very similar to the timed-limited diametric attack, and the proof follows the same logic (expect with more 'diametric' nodes each some distance at least d away from all others).

Lemma 1.10. *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.*

We conjecture that the uneven-polygonal attack also has the same bound as the polygonal one, as the idea is that any that are strictly greater are just worse for the patroller.

1.3 Star graph solution

As a special case of complete bipartite graph we have the star graph, $S_n = K_{1,n}$, that is a tree with one internal node (the centre) and n leaf nodes (the external nodes). Hence $V(S_n) = V(K_{1,n}) = \frac{m}{2n}$ for $m < 2n$ (by Theorem 15 (1) in [**?**]). This is achieved by the patroller forming a patrol which alternates between different external nodes and the centre, and embedded Hamiltonian patrol from C_{2n}, and the attacker attacking at all the external nodes with equal probability for a fixed time interval (or two consecutive time intervals if m is odd).

1.4 Elongating the star graph

The line and star graphs provided a good starting point for attempting to solve the problem for a general tree graph. If a more general version of the star graph can be solved, it may provide better bounds on tree graphs.

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 1.11 (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 figure 1.3, we note that $*$ nodes are symmetric and so we will be dealing with strategies that treat them as equal (see section 3.1 in [**?**]). We will from now assume that $n \geq 3$, as otherwise we are just dealing with the line graph, however we can think of $n = 2$ to get the results for the line graph for comparison.

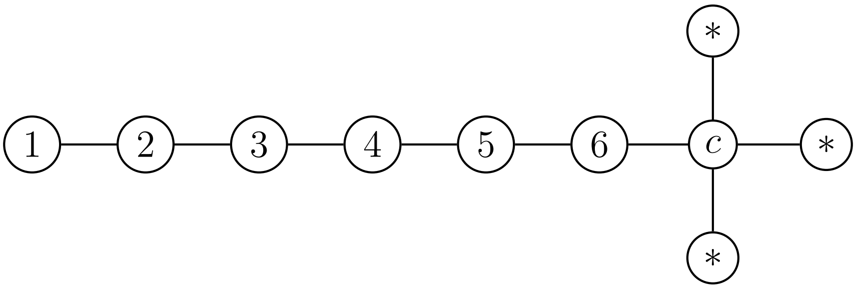


Figure 1.3: Labeling on the graph S_4^5.

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 to S_n^k.

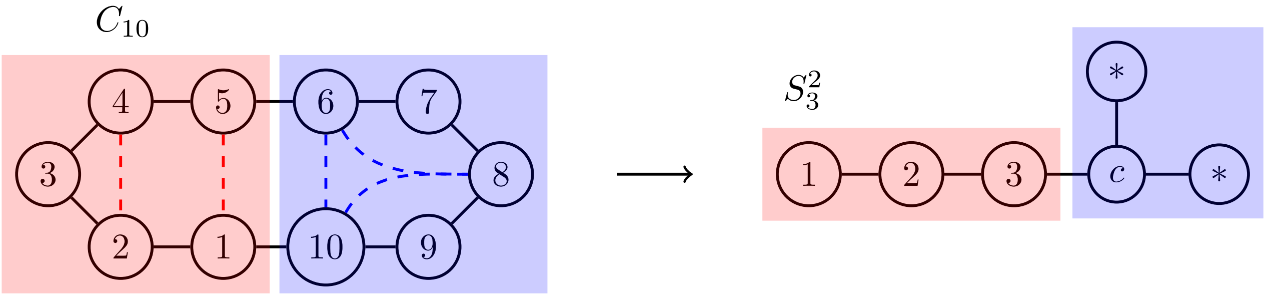


Figure 1.4: C_{10} can be simplified to S_3^2 by node identifying.

Definition 1.12 (Random Oscillation)**.** The simplification map from $C_{2(n+k)}$ to S_n^k is a mapping form nodes i and $2k+2-i$ to node i for $i = 1, ..., k$ and $2k, 2k+2, ..., 2k+2n-2$ to node c and $2k+2j-1$ to node $*$ for $j = 1, ..., n-1$.

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 1.13. *For $m \leq 2(n+k)$ following the Random Oscillation,*

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

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

The proof follows immediately from the simplification bound (Lemma 1 (4) in [**?**]) and the Hamiltonian solution (Theorem 13 (1) in [**?**]) i.e.

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

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

Note. We have the solution for $m = 2(n+k)$, but to be consistent with the regions in the line graph, we will pull this into the next region.

We could now consider applying the time-limited diametric attack to get bounds on the problem, $\bar{d} = n-1$ gives the bound $V \leq \max\{\frac{1}{2}, \frac{m}{2(n-1)}\}$, however this bound is not tight with our random oscillation bound, this is because we are not utilising all $*$ type nodes.

Instead of using just one $*$ node (which by symmetry will never yield optimal) we must utilise all the $*$ nodes equally, we can use weights on the distance from the centre. However when using these weights we must spread out the attack at each position in time, to ensure each potential attack in space and time has an equal weight.

Similiar to the proof of the bipartite graph bounds(Theorem 15 (1) in [**?**]), a timing issue arises when m is odd, however using two fixed time intervals fixes this problem. This holds even when m is even, hence to avoid the problem we will use a baseline spread of two consecutive time periods and then spread out from these according to the weight.

Definition 1.14 (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 for the attack, choose time intervals, $I, I+1, ..., I+2k+1$, each with equal probability for some fixed time interval, I (i.e starting attacks at $\tau, \tau+1, ..., \tau+2k+1$ equiprobably). If a $*$ node is chosen for the attack, choose time intervals, $I+k, I+k+1$, each with equal probability(i.e starting at times $\tau+k, \tau+k+1$ equiprobably).

A diagram showing how the potential attacks are spaced out in time can be seen in figure 1.5.

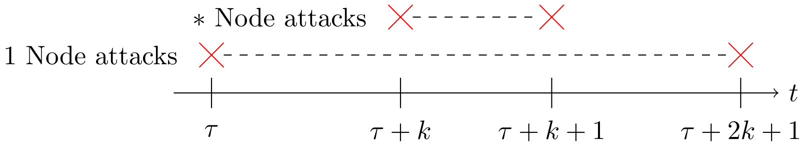


Figure 1.5: Time spacing of the time-delayed attack, for node 1 and $*$ nodes.

Note. By making the spread in time proportional to the weight of attacking the node, each potential attack in space and time has the same weight.

Lemma 1.15. *When $T \geq m+2k$, the upper bound $V \leq \max\{\frac{k+1}{n+k}, \frac{m}{2(n+k)}\}$ is guaranteed by the time-delayed attack.*

This bound is analogous to the diametric bound, and as long as we are in the range where $m \geq 2(k+1)$ so the lemma gives $V \leq \frac{m}{2(n+k)}$.

Theorem 1.16 (Solution in $m \geq 2(k+1)$)**.** *When $T \geq m+2k$, by the attacker using the time-delayed attack and the patroller using a random oscillation patrol, we achieve the value, when $2(k+1) \leq m \leq 2(n+k)$, of*

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

Hence we have a solution for 2 regions $m > 2(n+k)$ and $2(k+1) \leq m \leq 2(n+k)$, which we note are analogous to regions S_1 and S_2 in the line graph. Hence prompting definitions for our regions

- $S_1' = \{(n,m,k) \,|\, m > 2(n+k)\}$ which have values $V = 1$
- $S_2' = \{(n,m,k) \,|\, 2(k+1) \leq m \leq 2(n+k)\}$ which have values $V = \frac{m}{2(n+k)}$

An example of these regions and the elongated star graph's value can be seen in 1.6.

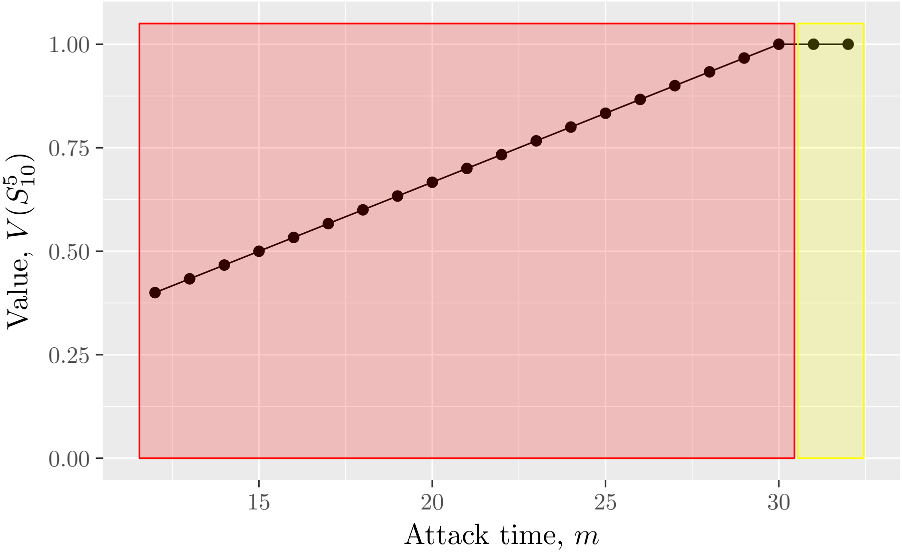


Figure 1.6: Value of the elongated star graph, S_{10}^5

We then seek solutions in the region $m < 2(k+1)$, however in this region we are below the random oscillation bound for the patroller and we can suggest some improvement (as in S_5 in the line graph) from the embedded Hamiltonian bound with. To see the issue and why the random oscillation can be improved we will look at the probability of interception under this patrol strategy.

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 appendix C.1),

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

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

Looking at the function (or the figure 1.7) it is clear that the issues are towards the end of the graph. This is because the next return under the random oscillation does not provide adequate coverage, hence we may wish to improve these end points of the graph. To do so we introduce the concepts of cycles which are played to guarantee capture of all attacks at all nodes within their cycle(*covering*).

Definition 1.17 (End-covering cycle)**.** Define a cycle of length m (if even) or $m-1$ (if odd) to be *end-covering* if one of the points along the cycle is a leaf node. Define the *half-length* of such cycles to be $\hat{m} = \lfloor \frac{m}{2} \rfloor$.

In [**?**] the authors improved the line graph's end nodes, which have poor interception probabilities, by introducing a end-covering cycle at each end. We shall do the same, though now more consideration needs to be taken on how to place these end-covering cycles.

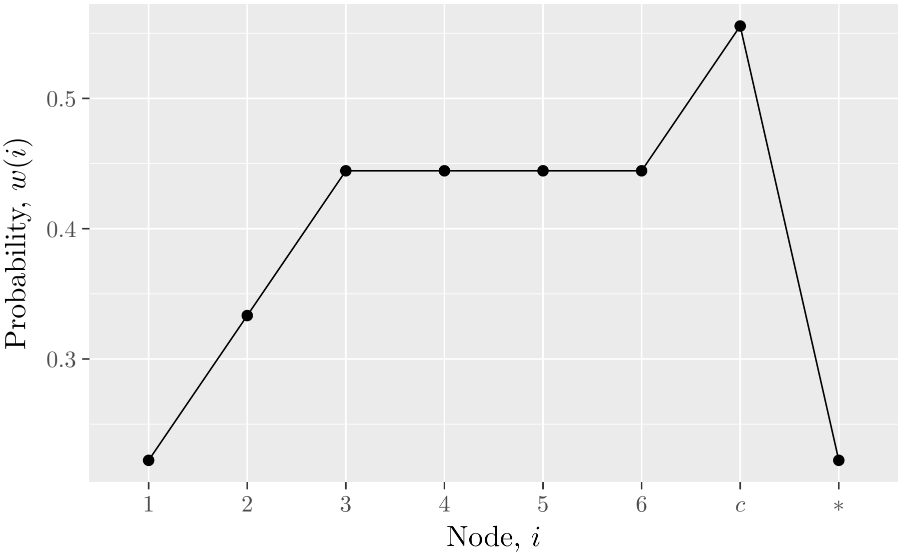


Figure 1.7: Interception probabilities of S_4^5 when $m = 4$.

We first classify nodes into types, we partition the node set, $N = L \cup M \cup R \cup S$, where

- Left nodes, $L = \left\{ i \,|\, i \leq \left\lfloor \frac{m}{2} \right\rfloor + 1, i \leq k+1 \right\}$,
- Middle nodes, $M = \left\{ i \,|\, \left\lfloor \frac{m}{2} \right\rfloor + 2 \leq i \leq n + k - \left\lfloor \frac{m}{2} \right\rfloor, i \leq k+1 \right\}$,
- Right nodes, $R = \left\{ i \,|\, i \geq n + k + 1 - \left\lfloor \frac{m}{2} \right\rfloor, i \leq k+1 \right\}$,
- Star node, $S = \{c, *\}$.

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

Then $V \geq w_{\min} \equiv w_{\min}^N = \min \left\{ w_{\min}^L, w_{\min}^M, w_{\min}^R, w_{\min}^S \right\}$, where $w_{\min}^X = \min\limits_{i \in X} w(i)$. We aim to use end-ensuring cycles to improve $w_{\min}$, which means improving the worst node sets, and hence improving the random oscillation bound.

Now from equation 1 we note some properties for each node set.

- $w_{min}^L = w(1)$, as for $i \in L$ we have that $w(i)$ is increasing in i.
- $w_{min}^M = w(\hat{m} + 2) = \frac{2m}{2(n+k)}$, 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 $\hat{m} + 2$ 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 $*$. We shall only consider improving strategies of the type above so only nodes, $1, \hat{m} + 2, k + 1, *$(if the set is non-empty) need to be considered.

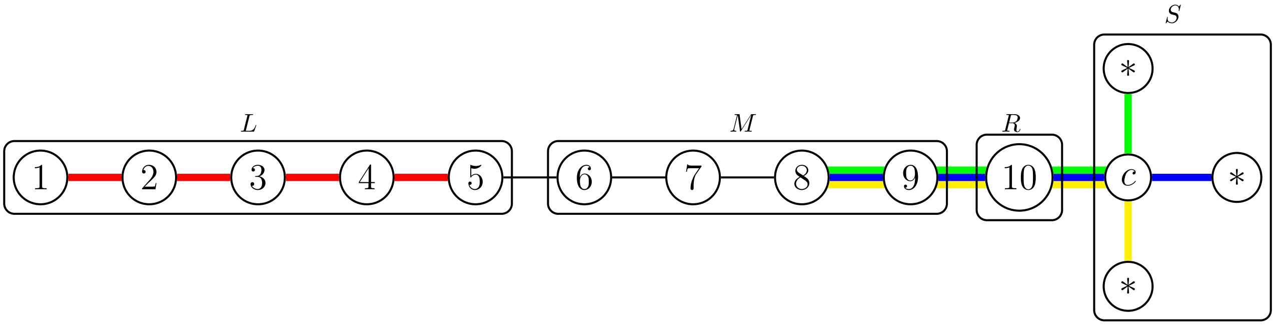
Similarly let $C_{\min}^X(\boldsymbol{\pi}) = \min\limits_{i \in X} P(\boldsymbol{\pi}, i)$, where $P(\boldsymbol{\pi}, i) = \max\limits_{I \subset \mathcal{T}} P(\boldsymbol{\pi}, [i, I])$, is the probability of capture under $\boldsymbol{\pi}$ and $C_{\min}(\boldsymbol{\pi}) \equiv C_{\min}^N = \min \left\{ C_{\min}^L, C_{\min}^M, C_{\min}^R, C_{\min}^S \right\}$. Then we seek to select $\boldsymbol{\pi}$ to get $C_{\min}(\boldsymbol{\pi}) > W_{\min} \equiv C_{\min}(\boldsymbol{\pi}_0)$, where $\boldsymbol{\pi}_0$ is the random oscillation strategy.

We will use P to be the probability that the random oscillation is played, and $Q = 1 - P$, to be the probability the improvement is played. We split, Q into $Q = Q_L + Q_M + Q_R + Q_S$, 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 $C^X_{\min} = PW^X_{\min} + q_X$.

Multiple choices for possible improvement exist, we first present the most simplest, but naive improvement. The improvement will begin by imagining the $*$ nodes are the ends of $n - 1$ lines and improve as in the improvment to the line's hamiltonian bound(section 5 in [**?**]). Because this approach is naive and is not fully utilising our graphical structure, we will adopt a combinatorial approach to which nodes to improve.

A naive improvement Let $\boldsymbol{\pi} = \boldsymbol{\pi}_N(Q_L, Q_S)$ denote the naive improvement policy. We will play the end-ensuring cycle, $\{1, ..., \hat{m} + 1, ..., 1\}$, with probability Q_L (a non-decreasing improvement), giving $q_L = Q_L$. We will play an end-covering cycle for each $*$ node, $\{*, c, ..., k + 3 - \hat{m}, ..., c, *\}$,(a non-decreasing improvement) with total probability Q_S, giving $q_S = \frac{Q_S}{n-1}$.

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



Example Figure 1.1: The Naive Improvement on S^9_4 for $m = 9$. The red lines indicating the end-covering 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 $\hat{m} = 4$).

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

Combinatorial Improvement

Definition 1.18 (Combinatorial Improvement)**.** Let $\boldsymbol{\pi}_C(Q_L, Q_S)$ denote the combinatorial improvement policy, which improves the sets, L and S. An end-covering cycle, $\{1, ..., \hat{m} + 1, ..., 1\}$ is played with probability Q_L, so $q_L = Q_L$. Also;

Case i) If $R \neq \emptyset$ then $\hat{m} = n + r$, for some excess r. Then form an end-covering cycle on the nodes $\{n + k + 1 - \hat{m}, ..., k + 1, c, *, c, *, ..., c, *, c, k + 1, ..., n + k + 1 - \hat{m}\}$. 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-covering cycle is formed by choosing $\hat{m}$ of the $*$ nodes (now labelled $*_1, ..., *_{\hat{m}}$) each with equal probability and forming a cycle $\{*_1, c, *_2, ..., c, *_{\hat{m}}, c, *_1\}$. This construction is performed with probability Q_S and nodes in S (c is better than $*$). The actual improvement made for a $*$ node is

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

We leave the exact analysis to appendix C.3 but are interested in the bounds:

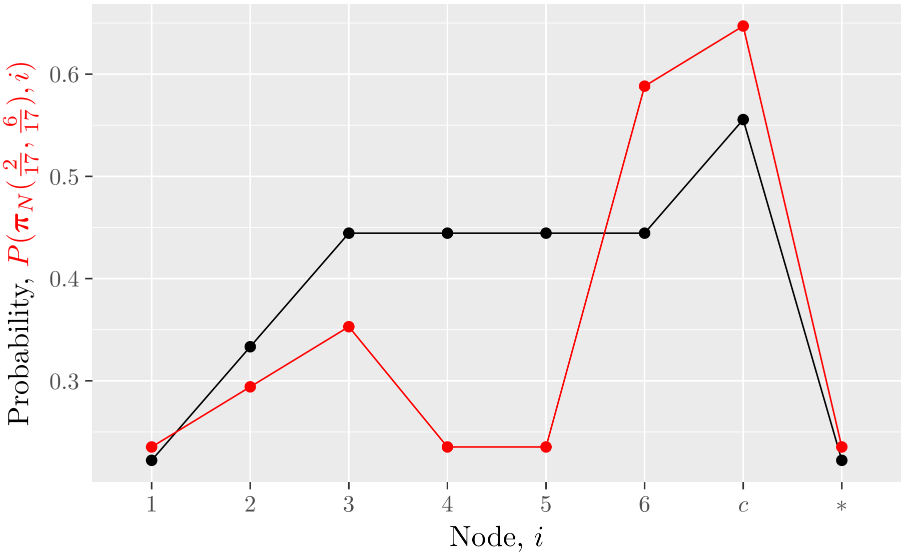


Figure 1.8: Interception probabilities of S_4^5 when $m=4$, with the red Probabilities showing the Naive Improvement Policy $\boldsymbol{\pi}_N\left(\frac{2}{17},\frac{6}{17}\right)$.

- If $M=\emptyset$,$R=\emptyset$ then $V\geq\frac{\hat{m}}{\hat{m}+n-1}$.
- If $M\neq\emptyset$,$R=\emptyset$ then $V\geq\frac{2m}{2(n+k)+m(1+\frac{n-1}{\hat{m}})}$.
- If $M\neq\emptyset$,$R\neq\emptyset$ then $V\geq\frac{2m}{2(n+k+m)}$.

It is conjectured in our future work (see section **??**) that the bound $V\geq\frac{2m}{2(n+k)+m(1+\frac{n-1}{\hat{m}})}$ is used in the region of $m<2(k+1)$. When m is even this bound becomes $V\geq\frac{2m}{2(2n+k-1)+m}$ which we will conjecture is tight with a multitude of attack strategies, however when m is odd due to the half-length, $\hat{m}\equiv\left\lfloor\frac{m}{2}\right\rfloor$, it makes the bound higher than when m is even, which will not be tight. We suggest some slight alteration when m is odd, but have to drop the use of end-covering cycles (we leave this idea to section **??**)

We will now deal with the case of $m\leq 2(n-1)$, so that all $*$'s nodes cannot be covered in one end-covering cycle. First looking at the $m=2k+1,2k$, we notice that by simplification of S_n^k into L_{k+1} and S_{n-1} we get the bound $V\geq\frac{1}{1+\frac{2(n-1)}{m}}=\frac{m}{m+2(n-1)}$.

When $m=2k+1$ we can propose an attack that 'augments' the time-delayed which simply removes one attack placed at 1. That is to place attacks at node 1 starting at times, $\tau,...,\tau+2k$ and $*$ nodes starting at times $\tau+k,\tau+k+1$ (equally we could use $\tau+k-1,\tau+k$) all with equal probability. Similarly if $m=2k$ we augment it to node 1 at times $\tau,...,\tau+2k-1$ and $*$ nodes at times $\tau+k-1,\tau+k$ all with equal probability.

Lemma 1.19. *Using the 'augmented' time-delayed we propose the attacker can get a bound of* $V\leq\frac{m}{m+2(n-1)}$ *and hence* $V(S_n^k)=\frac{m}{m+2(n-1)}$ *for* $m=2k+1,2k$

The idea of the proof for the 'augmented' time-delayed attack is the same as the time-delayed attack, but with a reduced number of attacks at node 1, to make sure the patroller can only get m by waiting here. No significant alteration to the proof takes place.

We leave the rest of the idea to solve in the region $m<2(k+1)$ to **??**, but respect that it may be difficult and careful consideration of attacker strategies may need to be taken when the random oscillation is not the best strategy for the patroller. This is analagous to the situtation in the region of S_5 for the line (see section 5 in [**?**]).

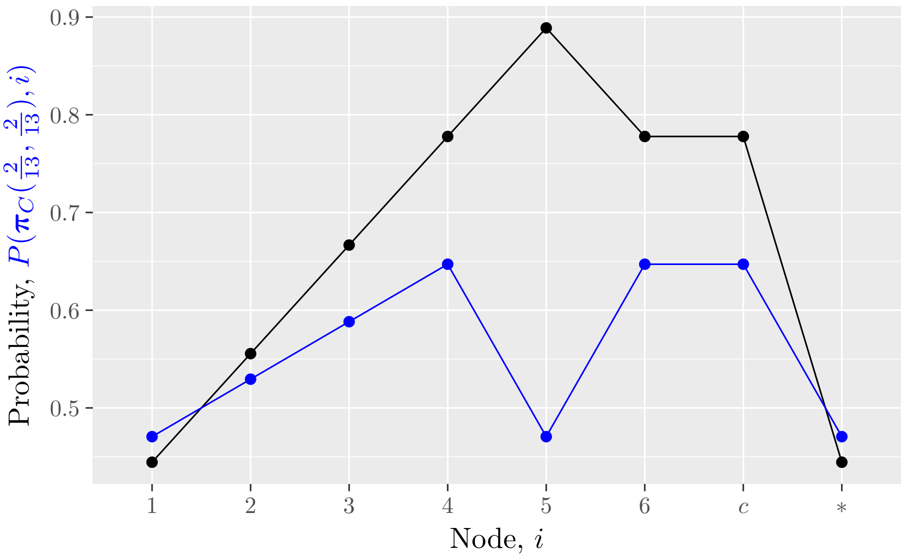


Figure 1.9: Interception probabilities of S_4^5 when $m = 8$, with the blue Probabilities showing the Choosing Improvement Policy $\boldsymbol{\pi}_C\left(\frac{2}{13}, \frac{2}{13}\right)$.

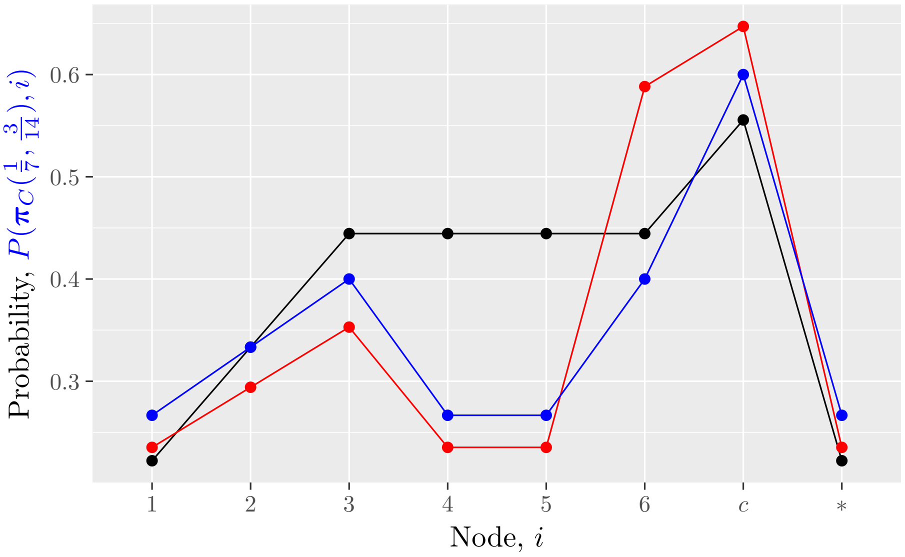


Figure 1.10: Interception probabilities of S_4^5 when $m = 4$, with the red Probabilities showing the Naive Improvement Policy $\boldsymbol{\pi}_N\left(\frac{2}{17}, \frac{6}{17}\right)$ and the blue Probabilities showing the Choosing Improvement Policy $\boldsymbol{\pi}_C\left(\frac{1}{7}, \frac{3}{14}\right)$.

1.5 Generalising the star graph

For now we will set aside the difficulty with the elongated star and look at generalising it, in which we can solve for values of m which match the random oscillation. We generalise further by allowing a general star graph to have all external nodes any distance from the centre node, again performed by subdivision of the edge an appropriate number of times.

Definition 1.20 (General star graph)**.** The *general star graph,* $S_n^{\boldsymbol{k}}$ ($\boldsymbol{k} \in \mathbb{N}^h$) is made from S_n, by performing subdivision on h of the initial edges each repeated k_i for each edge for $i = 1, ..., h$.

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 imagine that $\boldsymbol{k} \in \mathbb{N}^n$. For ease of notation we will also define define $|\boldsymbol{k}| \equiv \sum_{i=0}^{i=h} k_i$ and $k_{\max} \equiv \max_{i=1,2,..,h} k_i$.

The labelling will be done as in figure 1.11 and note that all nodes with the same label are symmetric and must be treated as such. We will refer to the branch made by k_i subdivisions of the edge as the i^{th} branch, which is made up of the nodes $n_{j,i}$ for $j = 1, ..., k_i + 1$, so $n_{j,i}$ is j^{th} node on i^{th} branch).

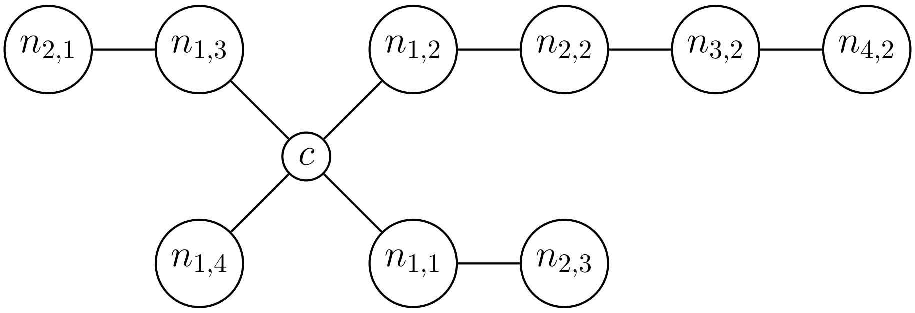


Figure 1.11: Labelling of $S_4^{1,3,1}$.

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 $C_{2(n+|\boldsymbol{k}|)}$, this can simplified to $S_n^{\boldsymbol{k}}$ by node identification. The identification mapping is analogous to that of the elongated star graph's simplification from a cyclic graph; Internal branch nodes are seen twice and the centre node is seen on every return, so the cycle graph can be made from this construction as in the example in figure 1.12.

The mapping is done as such:

- The centre is identified from nodes $1, 1 + 2(k_1 + 1), 1 + 2(\sum_{i=1}^{2}(k_i + 1)), ..., 1 + 2(\sum_{i=1}^{h}(k_i + 1), 1 + 2(|k| + h) + 2), ..., 1 + 2(|k| + h) + 2(n - h)$
- The first branch is identified from the nodes between 2 and $2(k_1 + 1)$ (Inclusive). $n_{1,i}$ for $i = 1, ..., k_1$ 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 $i = 1, ..., k_j$ 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 simplification mapping gives rise to the general random oscillation strategy (analogous to random oscillation on the elongated star).

Definition 1.21 (General random oscillation)**.** The *oscillation* on $S_n^{\boldsymbol{k}}$ is any embedded Hamiltonian patrol on $C_{2(n+|k|)}$ under the simplification above. The *random oscillation* on $S_n^{\boldsymbol{k}}$ is the embedded random Hamiltonian patrol on $C_{2(n+|k|)}$ under the simplification above.

Lemma 1.22. *For $m < 2(n+|k|)$ following the random oscillation*

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

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

The proof is straight forward by the application of the simplification (Lemma 1 (4) in [**?**]) and the Hamiltonian solution(Theorem 13 (1) in [**?**]).

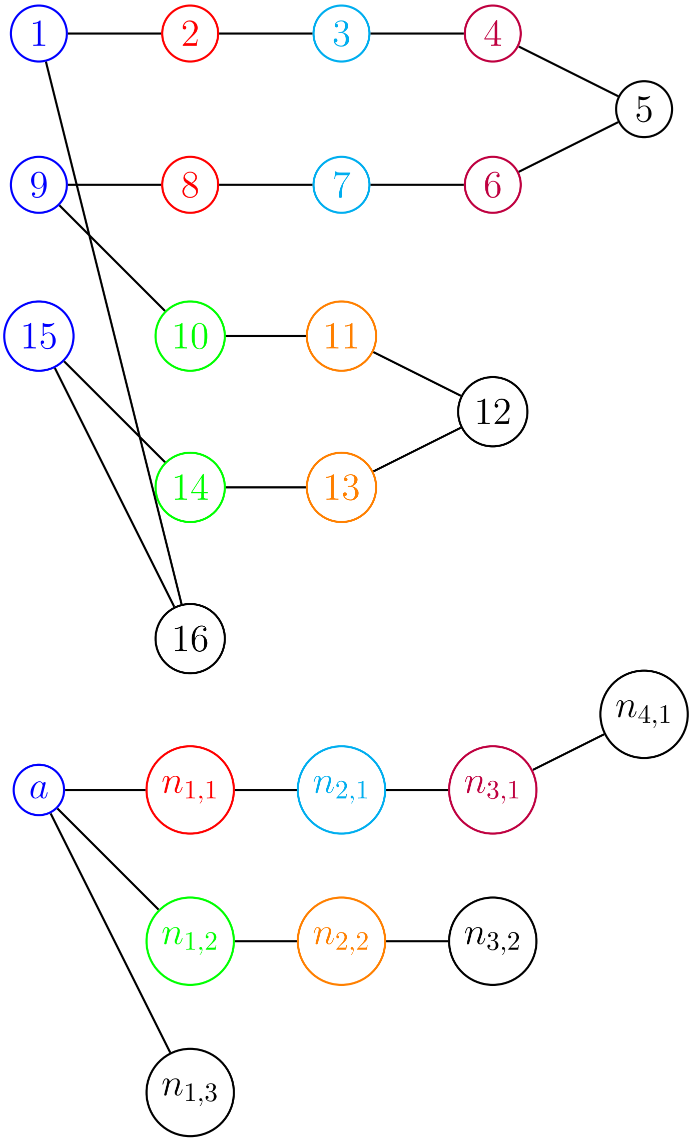


Figure 1.12: Simplification of C_{16} to $S_3^{3,2}$.

We will match the oscillation bound of $\frac{m}{2(n+|k|)}$, by further extending the time-delayed attack into the type-delayed attack, which will consider the length of the branches as the key to spacing out the attacks in time.

Definition 1.23 (Node types)**.** A *type i* node, is an external node which has been extended i times, that is the branch length. Then $k_{\max}$ is the maximum node type for $S_n^{\boldsymbol{k}}$.

Definition 1.24 (Type-delayed attack)**.** Let the *type-delayed attack*, be the attack that attacks at a type i node with probability $\frac{i+1}{n+\sum_{j=1}^{h} k_j}$ $\forall i$. For a fixed attack interval I, the attacks at a type i node choose an interval from the following with equal probability: $I+(k_{max}-i), I+(k_{max}-i)+1, ..., I+k_{max}+i+1$ $\forall i$ (i.e starting attacks at a type i node at times $\tau+(k_{max}-i)+1, ..., \tau+(k_{max}+i)+1$).

A diagram showing how potential attacks are spaced out in time, for each node of a given type, can be seen in figure 1.13.

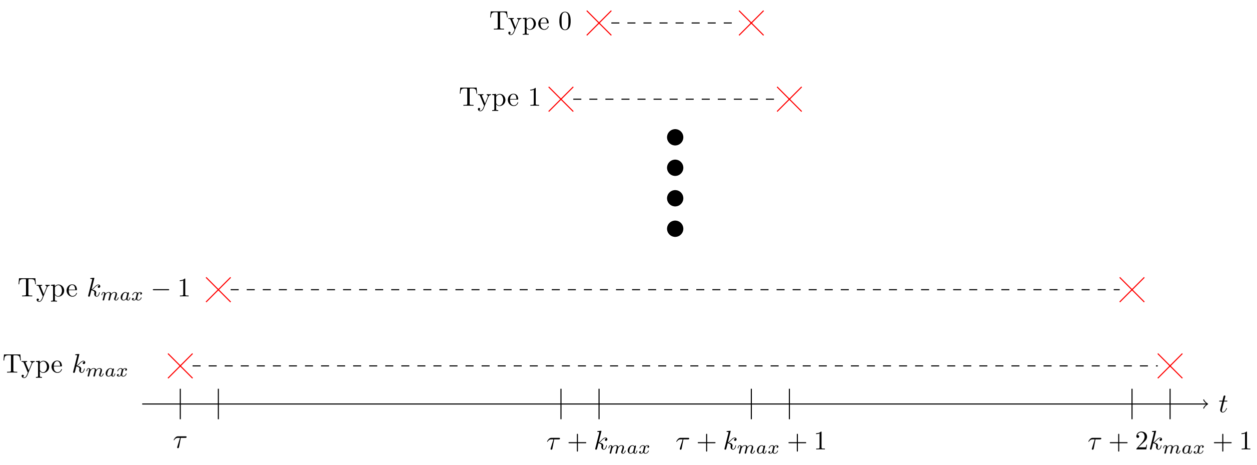


Figure 1.13: Time spacing of the type-delayed attack.

Theorem 1.25. *When $T \geq m+2k_{max}$, the upper bound is*

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

guaranteed by the type-delayed attack.

We leave the proof of this

Corollary 1.26 (Solution in $m \geq 2(k_{max}+1)$)**.** *When $T \geq m+2k_{max}$, by the attack using the type-delayed attack and the patroller using the random oscillation we achieve the value, when $2(k_{max}+1) \leq m \leq 2(n+|\boldsymbol{k}|)$,*

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

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

Just like in the elognated star graph, we have solved the problem for the analogous regions to S_1 and S_2 for the line. As in the elongated star graph we expect a multitude of issues moving to the region of $m < 2(k_{\max}+1)$, so for now we shall move onto an even more general result.

1.6 Joining star graphs by centralised connections

We now look at connecting by their centres the generalised star (with branches).

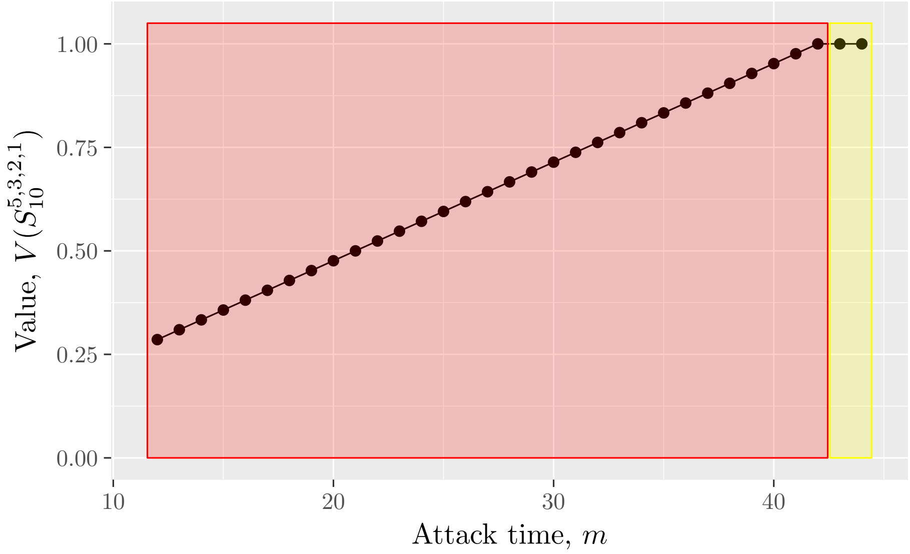


Figure 1.14: The value of the general star graph, $S_{10}^{5,3,2,1}$.

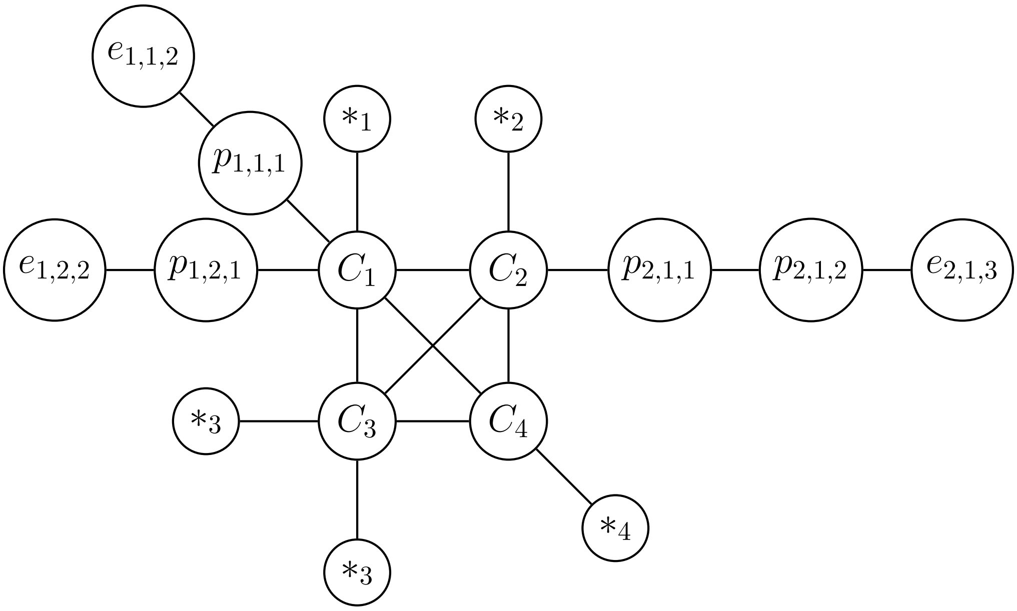


Figure 1.15: Example of labelling on $(S_3^{1,1}, S_2^2, S_2, S_1)$.

Definition 1.27. We define the *multi general p-star graph*, $(s_{n_1}^{\boldsymbol{k}_p}, ..., S_n^{\boldsymbol{k}_p}) \equiv \bigodot_{i=1}^{p} S_{n_i}^{\boldsymbol{k}_i}$, to be the p star graphs, $S_{n_i}^{\boldsymbol{k}_i}$ initially with disconnected centres which are now made adjacent by the introduction of connections between each combination of centres (i.e the complete graph of centres).

Because we have just joined graphs we know some solutions to, we can use decomposition for the patrollers strategy, and simplification for the attackers strategy.

Theorem 1.28 (Separable solution)**.**

For $2(K_{max}+1) \leq m \leq 2 \min_{i=1,...,p} \{n_i + \sum_{j=1}^{h_i} (\boldsymbol{k}_i)_j\}$,

$$V = \frac{m}{2\sum_{i=1}^{p}\left(n_i + \sum_{j=1}^{h_i}(\boldsymbol{k}_i)_j\right)} \equiv \frac{m}{2(|\boldsymbol{n}| + |\boldsymbol{K}|)}$$

where $\boldsymbol{K}$ *is the concatenation of all the* $\boldsymbol{k}_i$ *vectors for* $i = 1, ..., p$*, with* $K_{max} = \max \boldsymbol{K}$ *and* $\boldsymbol{n} = (n_1, ..., n_p)$.

Proof. Under decomposition with the Hamiltonian bound for the general star graphs, as $m \leq 2 \min_{i=1,...,p} \{n_i + \sum_{j=1}^{h_i} (\boldsymbol{k}_i)_j\}$

$$V \geq \frac{1}{\sum_{i=1}^{k} \frac{2(n_i + |\boldsymbol{k}_i|)}{m}} = \frac{m}{2\sum_{i=1}^{k}(n_i + |\boldsymbol{k}_i|)}$$

Now under simplification of the centre's, i.e $\bigodot_{i=1}^{p} S_{n_i}^{\boldsymbol{k}_i}$ to $S_{|\boldsymbol{n}|}^{\boldsymbol{K}}$, as $m \geq 2(K_{\max}+1)$

$$V \leq \frac{m}{2(|\boldsymbol{n}| + |\boldsymbol{K}|)}$$

So we have a tight upper and lower bound. □

Note. We note that this is not the Hamiltonian bound for such a graph and the Hamiltonian bound would be,

$$V = \frac{m}{2(|\boldsymbol{n}| + |\boldsymbol{K}|) + p}$$

Note. The idea of requiring the complete graph of connection between centres is not needed for the separable solution.

A Examples

A.1 Example of decomposition

Example A.1. For Q as seen in Figure A.1. Consider when $m = 3$, the decomposition of Q into the graphs $Q_1 \equiv L_2$ and $Q_2 \equiv L_3$. $V_1 = V(L_2) = 1$ 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 $V \geq \frac{1}{\frac{1}{V_1}+\frac{1}{V_2}} = \frac{1}{\frac{1}{1}+\frac{3}{4}} = \frac{4}{7}$.

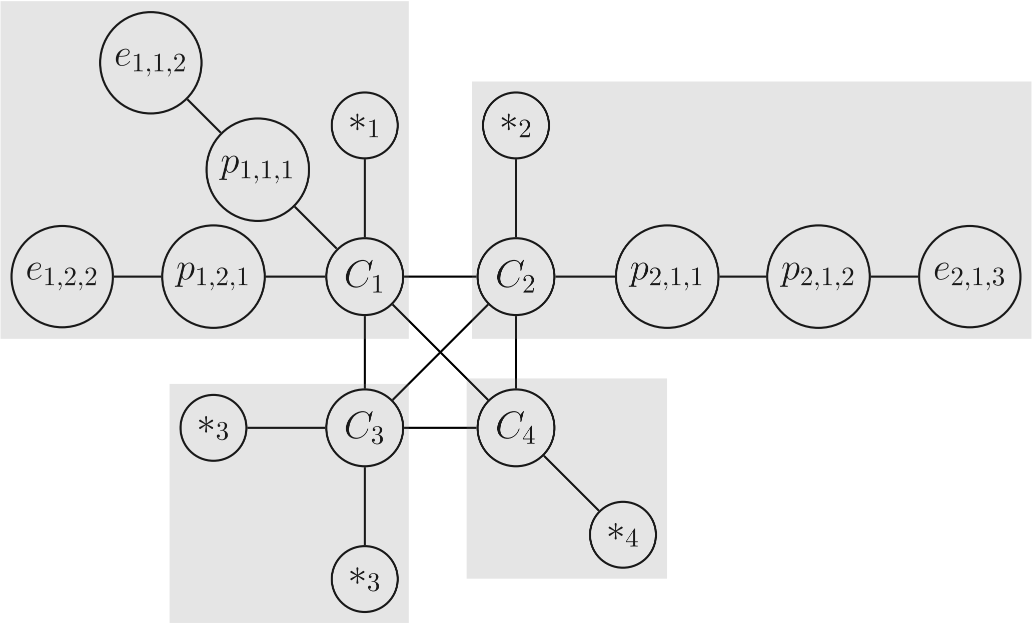


Figure 1.16: Example of decomposition on $(S_3^{1,1}, S_2^2, S_2, S_1)$.

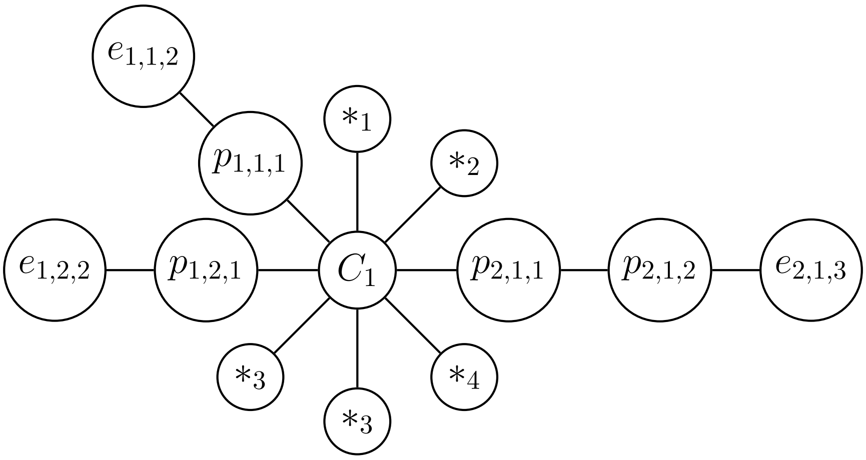


Figure 1.17: Example of simplification on $(S_3^{1,1}, S_2^2, S_2, S_1)$ to $S_8^{2,1,1}$.

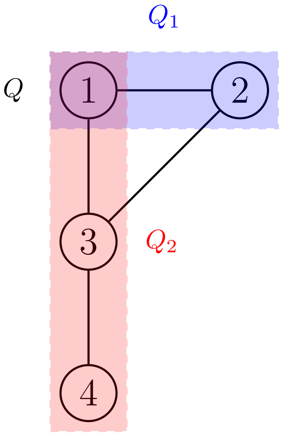


Figure A.1: Decomposition of Q into Q_1 and Q_2.

A.2 Example of simplification

Example A.2. For Q as seen in Figure A.2, when $m = 3$, the simplification of the graph by identifying nodes 1 and 2 simplifies Q to $Q' = L_3$. Hence we can get the bound that $V(L_3) \geq V(Q)$ so $V(Q) \leq \frac{3}{4}$.

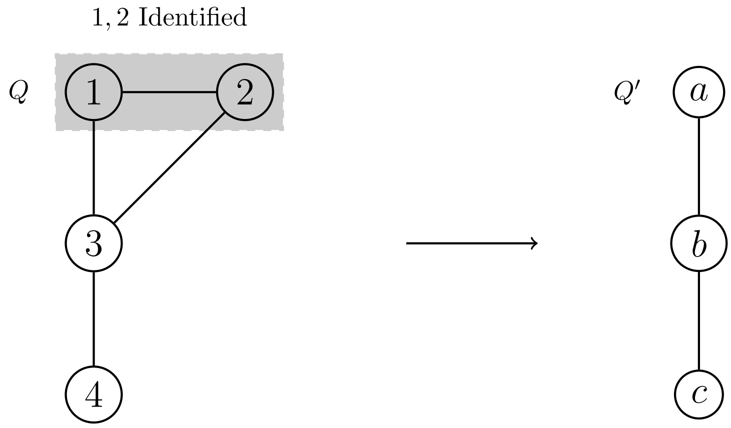


Figure A.2: Simplifcation of Q to Q' by identification.

B Patrolling games

B.1 Proof of diametric waiting time

Consider visit i to a end node capturing C_i of the attacks placed by the diametric attack, then the total number of attacks captured is $C = \sum_{i=0}^{\lfloor \frac{T-1}{\bar{d}} \rfloor} C_i$.

Then leaving the initial node at time t gives us, $C_0 = \min(t, T-m+1)$, $C_i = \big(\min(t+i\bar{d}, m, T-(t+i\bar{d})\big)_+$ for $i \neq 0$.

We first note that C_0 is increasing in the region $t \leq T-m$ and constant for $t \geq T-m+1$. So C_0 is concave. Due to this being increasing, and therefore its only possible to have others increasing if $t \leq T-m$, we will make this an assumption.

Now we look at C_i and see it is increasing for the region $t \leq m-1-i\bar{d}$, constant for $m-i\bar{d} \leq t \leq T-1-m-i\bar{d}$ and decreasing for $T-m-i\bar{d} \leq t \leq T-i\bar{d}-1$. So C_i is concave. Hence C is concave and so finding the best choice for t is when the net increase is constant (or decreasing for the first time).

We see that C_0 always improves, contributing 1 to the net increase. C_i however is only increasing, and contributes 1, if $i \leq \frac{m-1-t}{\bar{d}} < 2$ (as $m \leq 2\bar{d}$), so only C_1 can possibly contribute an increase when $t \leq m-\bar{d}-1$.

C_i is decreasing, contributing -1 if $\frac{T-m-t}{\bar{d}} \leq i \leq \frac{T-1-t}{\bar{d}}$, with at most 2 C_i's being decreased (as the gap is $\frac{m}{2\bar{d}}$ and $\bar{d} < m \leq 2\bar{d}$).

This worst issue occurs when $\frac{T-m-t}{\bar{d}}$ or $\frac{T-1-t}{\bar{d}}$ are integers, meaning we have chosen $t = T-m-k\bar{d}$ or $t = T-1-k\bar{d}$ for some $k \in \mathbb{Z}$.

So overall increasing t to $t+1$, with $t \leq m-\bar{d}-1$, gives us a net increase of 1 when we have non-integers, 0 when we have integers and -1 if $t > m-\bar{d}-1$.

So we pick the upper concave region, as its about to go from increasing to decreasing, giving us a choice of

$t = m - \bar{d}$.

Note. $t = m - \bar{d} - 1$ is not net decreasing, but $t = m - \bar{d} - 1$ is net decreasing, so $t = m - \bar{d}$ is a choice for the maximum.

B.2 Proof of conditions on T for diametric attack

We first justify the counting formula,

$$\underbrace{m-\bar{d}}_{\text{Waiting initially}} \quad + \quad \underbrace{(m \times (M+1))_+}_{\text{Visits which get exactly } m \text{ attacks}}$$
$$+ \underbrace{\big(T - \big(m - 1 + (M+1)\,\bar{d}\big)\big)_+ + \big(T - \big(m - 1 + (M+2)\,\bar{d}\big)\big)_+}_{\text{Penultimate and final node visits}}$$

Where $M = \left\lfloor \frac{T-2m+1}{\bar{d}} \right\rfloor$.

Initially by waiting till time $m - \bar{d} - 1$ the patroller collect $m - \bar{d}$ potential attacks, they then leave and hit the next diametric node at time $m - 1$ getting them m potential attacks.

Then we notice that the last time we can arrive at diametric node to get m potential attacks is at time $T - m$. so how many diametric nodes do we get exactly m at , the initial one (at time $m - 1$) and then $\left\lfloor \frac{T-m-(m-1)}{\bar{d}} \right\rfloor = \left\lfloor \frac{T-2m+1}{\bar{d}} \right\rfloor = M$ additional ones. This is the second piece of the equation.

For the final piece we consider the times we are a diametric node, after we get m, we will be there at times $m - 1 + (M+1)\bar{d}, m - 1 + (M+2)\bar{d}, ...,$ getting us $T - m + 1 - (M+1)\bar{d}, T - m + 1 - (M+2)\bar{d}, ...$ As we know that $T - m + 1 - (M+1)\bar{d} < m \implies T - m + 1 - (M+3)\bar{d}, T - m + 1 - (M+4), ... < 0$, so we only have possibly the penultimate and final node visits.

Hence the result is as above.

Proof. Using $T = m-1+(k+1)\bar{d}$ in the formula gives, $M = \left\lfloor \frac{(k+1)\bar{d}-m}{\bar{d}} \right\rfloor = (k+1)+\left\lfloor \frac{-m}{\bar{d}} \right\rfloor = (k+1)-2 = (k-1)$ (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).

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

which is $m-\bar{d}+(mk)_+ + \big((k+1)\bar{d} - k\bar{d}\big)_+ + \big((k+1)\bar{d} - (k+1)\bar{d}\big)_+$ giving $m-\bar{d}+mk+\big(\bar{d}\big)_+ + (0)_+ = m(k+1)$. 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 $r = m - \bar{d}$ (i.e $T = 2m - 1 + k\bar{d}$).

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.

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

Then we notice two cases:

1. If $0 \leq r < m - \bar{d}$ then $M = k - 1$ and we get a fraction of potential attacks caught of $\frac{m(k+1)+2r}{2((k+1)\bar{d}+r)}$
2. If $m - \bar{d} \leq r < \bar{d}$ then $M = k$ and we get a fraction of potential attacks caught of $\frac{m(k+1)+r+m-\bar{d}}{2((k+1)\bar{d}+r)}$

Hence the best in the first case is $r = m - \bar{d} - 1$ for $\frac{m(k+3)-2\bar{d}-2}{2(k\bar{d}+m-1)}$ and in the second case $r = m - \bar{d}$ for $\frac{m(k+3)-2\bar{d}}{2(k\bar{d}+m)}$, hence it is clear the best is $r = m - \bar{d}$.

Then we show that the maximal subsequence tends to the bound as $T \to \infty$, i.e as $k \to \infty$. Well the maximal subsequence is $\frac{m(k+3)-2\bar{d}}{2(k\bar{d}+m)} \to \frac{m}{2\bar{d}}$ as $k \to \infty$, hence as the maximal subsequence tends down to the bound, it implies the result stated. □

B.3 Proof of time-limited diametric attack

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, $t_1 = \min\{t \,|\, W(t) = n_1\}$, 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): $t_l \in \{0, 1, ..., m-2\}$, to pick to maximize the number of attacks caught; or $t_L = \infty$, never leaving to get $\bar{d}$ attacks.

Leaving: Choosing $t_L \in \{0, 1, ..., m-2\}$ 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 $\left(m - 1 - \bar{d}\right)_+ \leq t_L \leq \bar{d} - 1$ 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 explicity it gives $V \leq \frac{1}{2}$ if $m < \bar{d}$, and $V \leq \frac{m}{2\bar{d}}$ is $m \geq \bar{d}$. □

B.4 Proof of time-delayed attack

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 becuase we either get:

$$k+m \text{ odd } \underbrace{\frac{1}{n+k}+...+\frac{1}{n+k}}_{\frac{k+m-t}{2}\text{times}}$$

$$k+m \text{ even } \underbrace{\frac{1}{n+k}+...+\frac{1}{n+k}}_{\frac{k+m-1-t}{2}-1\text{times}}+\tfrac{1}{2(n+k)}$$

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 in fact the game will be over) and we have no incentive not to do it (IN FACT IF THE CONDITION ABOUT LIMITED number of $*$ nodes is brought up it is in fact 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 decide 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 yield.

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

($+\frac{1}{n+k}$ 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

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

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

Hence it is better to move and 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 are 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 scenarios 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. □

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 there is no point in moving and in fact 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 there is no point in moving and in fact 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 in fact 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 additional 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 latter, 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 attacks).

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 $r \geq 1$) □

Lemma B.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 [Missing condition for being in the 'middle of game']

Note. This payoff does not affect the initial 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

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

.

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 $x = 0$ $\left(k+m-t-2s-1-(x-2s)_+\right)_+ = (k+m-t-2s-1)_+ = (3k+1+r-t-2s)_+ \leq (2k+1+r-2s)_+$,the payoff 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 occuring 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 well 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

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

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

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

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

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

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

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

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 $x geq 1$ then the payoff is

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

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 except 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. □

Theorem B.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 $q + k + 2 > k + m = 3k + 2 + r$

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

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

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

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

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

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. □

B.5 Proof of type-delayed attack

We begin with a lemma used in the proof

Lemma B.3. *When $k_{max} - i \leq t \leq k_{max} + i$ we get that*
$$\frac{(k_{max}+1+i-t)_+}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)} + \frac{(m-2(i+1))_+}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)} \geq \frac{(k_{max}-i+m-t-1)_+}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)}$$

Proof. If $m \geq 2(i+1)$, then we get
$$\frac{k_{max}+1+i-t+m-2(i-1)}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)} = \frac{k_{max}-i+m-t-1}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)} \geq \frac{(k_{max}-i+m-t-1)_+}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)}$$

If $m \leq 2(i+1)$, then we get
$$\frac{k_{max}+1+i-t}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)} \geq \frac{(k_{max}+1+i-t+m-2(i+1))_+}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)} = \frac{(k_{max}-i+m-t-1)_+}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)}$$

□

The main proof of the attack lemma's future payoff

Note. The difference in future payoffs 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 B.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:*

$$\begin{array}{ll} \dfrac{(k_{max}+1+i-t)_+}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)}+\dfrac{(m-2(i+1))_+}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)} & \textit{if } k_{max}-i\leq t\leq k_{max}+i \\ \dfrac{(k_{max}-i+m-t-1)_+}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)} & \textit{if } t\geq k_{max}+i+1 \end{array}$$

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

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

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

$$\underbrace{\frac{\min(2(j+1),(k_{max}+j+m-(t_1+i+2+j)+1)_+)}{2(j+1)}\times\frac{j+1}{n+\sum\limits_{j=1}^{h}k_j}}_{\text{Immediate reward at type } j}$$
$$+\underbrace{\frac{(k_{max}-j+m-(t_1+i+2+j)-1)_+}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)}}_{\text{Future reward from type } j}$$
$$=\frac{\min(2(j+1),(k_{max}+m-t_1-i-1)_+)}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)}+\frac{(k_{max}-i+m-t_1-2j-3)_+}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)}$$

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

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

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

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

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

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

Moving gives

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

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

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

Hence with either option the best that the patroller can do is $\frac{1}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)} + \frac{(m-2(i+1))_+}{2\left(n+\sum\limits_{j=1}^{h}k_j\right)}$, 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

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

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

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

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 $\frac{q}{2\left(n+\sum\limits_{j=1}^{h} k_j\right)}$

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

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

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

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

Hence by mathetmatical 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 recieving 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

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

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 $\frac{k+1}{n+\sum_{j=1}^{h} k_j}$. □

B.6 Generalised ARHP

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

Definition B.5 (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 $i = 1,..,k$, where $\sum\limits_{i=1}^{k} p_i = \frac{k}{n}$.

Lemma B.6. *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 $i = 1,...,k$. Therefore by following the k type BRHP, $\boldsymbol{\pi}_{BRHP}$, with probability p_i at "type i" for $i = 1,...,k$. Then

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

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

Hence as it holds for all pure attacks

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

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 $k' \in \mathbb{N}$.

C Improving random oscillations

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 $2(i-1) < 2(n+k+i-1), 2(i-1) = 2(n+k+1-i),$ 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 analysis

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

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

   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 $C_{min}(\alpha) = \min\{C^L_{min}, C^M_{min}, C^S_{min}\} = \min\{Pw(1)+q_L, Pw(\lfloor \frac{m}{2}+2 \rfloor), Pw(*)+q_S\} = P\frac{m}{2(n+k)} + \min\{q_L, P\frac{m}{2(n+k)}, q_S\}$. As the patroller wishes to select q_L, q_S to maximize this probability, the problem becomes

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

   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 $C_{min}(\alpha) = \min\{C^L_{min}, C^M_{min}, C^R_{min}, C^S_{min}\} = \min\{Pw(1)+q_L, Pw(\lfloor \frac{m}{2}+2 \rfloor), Pw(k+1)+q_S, Pw(*)+q_S\} = P\frac{m}{2(n+k)} + \min\{q_L, P\frac{m}{2(n+k)}, q_S\}$. 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

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

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 analysis

First we will note that the cycle in i) is of length $2(n - \hat{m} + 1) + 2(n-1) = 4n - 2\hat{m} = 4n - 2(n+r) = 2n + 2r = 2\hat{m} \leq m$ and so is end-covering

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

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

   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 $C_{min}(\beta_2) = \min\{C^L_{min}, C^M_{min}, C^S_{min}\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2), Pw(*) + q_S\} = P\frac{m}{2(n+k)} + \min\{q_L, P\frac{m}{2(n+k), q_S}$. As the patroller wishes to select q_L, q_S to maximize this probability the problem becomes

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

   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 $C_{min}(\beta_1) = \min\{C^L_{min}, C^M_{min}, C^R_{min}, C^S_{min}\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2), Pw(k+1) + q_S, Pw(*) + q_S\} = \min\{Pw(1) + q_L, Pw(\lfloor \frac{m}{2} \rfloor + 2), Pw(*) + q_S\} = P\frac{m}{2(n+k)} + \min\{q_L, P\frac{m}{2(n+k)}, q_S\}$ 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

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

   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 improvement extension

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 $Q_S = \frac{n-1}{\frac{m'}{2}} q_S$).

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

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

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 $(1 - \frac{m+2(n-1)}{m} q_L) \frac{m}{2(n+k)} + \min\{q_L, (1 - \frac{m+2(n-1)q_L}{m}) \frac{m}{2(n+k)}\}$ 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 Optimal Solution for a Random Attacker

To solve the MDP model, it is first worht noting that it belongs to the class of multichain models as in Chapter 9 of [**?**], as a stationary deterministic policy may have multiple recurrent MC's. To determine C^{OPT} we need to solve the following equations for $g(\boldsymbol{s})$ and $h(\boldsymbol{s})$ known as the Multichain Optimality equations in [**?**] (Equations (9.1.1) and (9.1.2))

$$g(\boldsymbol{s}) = \min_{i \in \mathcal{A}(\boldsymbol{s})} \{g(\phi(\boldsymbol{s}, i))\} \quad \forall \boldsymbol{s} \in \Omega$$
$$g(\boldsymbol{s}) + h(\boldsymbol{s}) = \min_{i \in \mathcal{B}(\boldsymbol{s})} \{C(\boldsymbol{s}, i) + h(\phi(\boldsymbol{s}, i))\} \quad \forall \boldsymbol{s} \in \Omega$$

Where $\mathcal{B}(\boldsymbol{s})$ is the subset of $\mathcal{A}(\boldsymbol{s})$ in which actions attain the minimum of the first equation, so that the system of equations is nested. The quantity $g(\boldsymbol{s})$ represents the long-run average cost starting from state $\boldsymbol{s}$ and $h(\boldsymbol{s})$ is a bias term from starting at state $\boldsymbol{s}$ and can be interpreted as the transient cost.

This allows the Primal linear program to be defined as

$$\max \sum_{\boldsymbol{s} \in \Omega} \alpha(\boldsymbol{s}) g(\boldsymbol{s})$$

subject to

$$g(\boldsymbol{s}) \leq g(\phi(\boldsymbol{s}, i)) \quad \forall i \in \mathcal{A}(\boldsymbol{s}) \quad \forall \boldsymbol{s} \in \Omega$$

and

$$g(\boldsymbol{s}) + h(\boldsymbol{s}) \leq C(\boldsymbol{s}, i) + h(\phi(\boldsymbol{s}, i)) \quad \forall i \in \mathcal{A}(\boldsymbol{s}) \quad \forall \boldsymbol{s} \in \Omega$$

For some choice of $\alpha(\boldsymbol{s}) > 0$ with $\sum_{\boldsymbol{s} \in \Omega} \alpha(\boldsymbol{s}) = 1$.

As discussed in **??**, C^{OPT} is the same for all states, as long as the graph is connected, and so $g = C^{\text{OPT}}$ is the same for all states. Hence equation one is always solved and $\mathcal{B}(\boldsymbol{s}) = \mathcal{A}(\boldsymbol{s})$ so our problem reduces to a linear problem.

$$\max g$$

subject to

$$g + h(\boldsymbol{s}) \leq C(\boldsymbol{s}, i) + h(\phi(\boldsymbol{s}, i)) \quad \forall i \in \mathcal{A}(\boldsymbol{s}) \quad \forall \boldsymbol{s} \in \Omega$$

This multichain linear program idea can be used for all finite state space MDP's and so it be easily adapted to deal with a change in cost and state space as in **??**. We also note that it is possible to determine the attack using the dual linear program (see equations 9.3.4, 9.3.5, 9.3.6 in [**?**])