Second Year Report

Thomas Lowbridge
School of Mathematical Sciences
University of Nottingham

June 8, 2018

Contents

1 Literature Review

1.1 Overview of Search games

Due to the high dependency on graphical structure, we provide a comprehensive guide to graph definitions in Appendix A.

1.2 Review of Strategic Patrolling games

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

- The graph $Q = (N, E)$ made of nodes, N ($|N| = n$), joined by edges, E, which can be represented by an adjacency matrix, A.
- The length of time over which the game takes place, the time-horizon T.
- The length of time the attack takes to complete, the attack-time m.

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

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

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

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

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

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

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

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

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

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

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

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

(with the change of $\mathcal{W}$ to $\mathcal{W}^*$ if we are playing the periodic game).

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

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

Many general properties for the game can be easily proven; such as that the one-off game is non-increasing in T (as it simply increases the size of $\mathcal{I}$) and introducing more edges for the same node set doesn't lower the value (it only increases the choice in the set $\mathcal{W}$ and $\mathcal{W}^*$). Also as $\mathcal{W}^* \subset \mathcal{W}$, it is obvious that $V^p(Q, T, m) \leq V^o(Q, T, m)$ (see [**?**] for details). So solving the one-off game gives an upper bound for the periodic game.

We shall now focus on the unrestricted, one-off game for the rest of this report.

1.2.1 Bounds and Tools

We shall provide a list of attacks and patrollers to give bounds on V which can be applied to all graphs. The lower bounds are given in terms of the patroller's "good" strategy against all attacker options, similarly the upper bounds are given in terms of the attacker's "good" strategy against all the patroller options. When we reach tightness between the bound these "good" strategies become an optimal solution.

General bounds(Patroller and Attacker):

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

Lemma 1.1 (General bounds)**.**

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

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

Decomposition(Patroller):

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

Lemma 1.2 (Decomposition lower bound)**.** *Consider decomposing Q into edge-preserving subgraphs Q_i for $i = 1, ..., k$ with values $V_i = V(Q_i)$ such that $Q = \bigcup_{i=1}^{k} Q_i$ then*

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

Furthermore in the case of a disjoint decomposition equality is reached

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

Example 1.3. For Q as seen in Figure **??**. Consider when $m = 3$, the decomposition of Q into the graphs Q_1 and Q_2 (as in Example Figure 1.1). $V_1 = V(Q_1) = 1$ as alternating between 1 and 2 can catch every attack. $V_2 = V(L_3) = \frac{3}{4}$ (as seen in [**?**]).

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

Simplification(Patroller and Attacker):

Definition 1.4 (Node Identification)**.** The operation of Node identification on two nodes, u and v, of a graph, $G = (N, E)$ into a single node w, is a mapping $f : N \to N'$ resulting in a new graph $G' = (N', E')$ where $N' = (N \backslash \{u, v\}) \cup \{w\}$ with $E' = E \setminus \{(u, v)\}$ if $(u, v) \in E$ and under the condition that $\forall x \in N$, $f(x) \in N'$ is incident to $e' \in E'$ iff $e \in E$ is incident to $x \in N$. Furthermore if a graph, Q, undergoes repeated node identification to become Q' then we say it has been simplified.

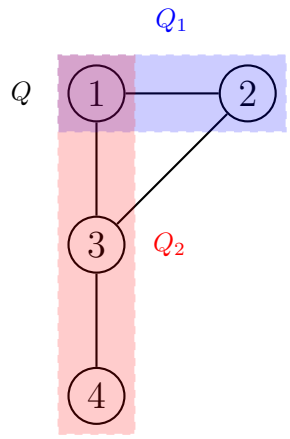


Figure 1.1: Decomposition of Q into Q_1 and Q_2.

Definition 1.5 (Embedded walk)**.** An *Embedded walk*, W', on Q' is the walk, W, done on Q under the simplification mapping of Q to Q'. i.e if $\pi : Q \to Q'$ is the simplification map, then $W' = \pi(W)$.

Lemma 1.6 (Simplification)**.** *If Q' is a simplified version of Q then $V(Q') \geq V(Q)$*

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

Example 1.7. For Q as seen in Figure **??**. Consider when $m = 3$, the Simplification of the graph by identifying $1, 2$ from Q to $Q' = L_3$ (as seen in Example Figure 1.2). Hence we can get the bound that $V(L_3) \geq V(Q)$ hence $V(Q) \leq \frac{3}{4}$

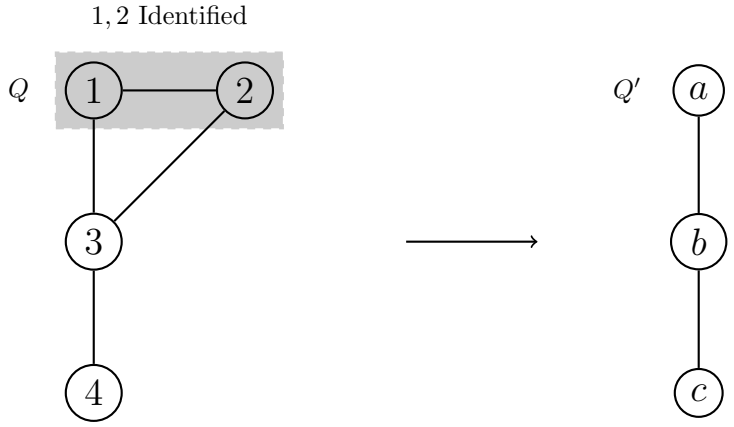


Figure 1.2: Simplifcation of Q to Q' by identification.

Diametric attack(Attacker)

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

Definition 1.8 (Graph Diameter)**.** The diameter of a graph Q is definded by $\bar{d} = \max_{i,i' \in N} d(i, i')$. The node pairs satisfying this are called diametrical.

Lemma 1.9. *By the attacker playing equally likely at a pair of diametrical at a random time interval, called the diametric attack, gives* $V \leq \max\left\{\frac{m}{2\bar{d}}, \frac{1}{2}\right\}$

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

Example 1.10 (Problem with diametric attack)**.** Consider the graph L_5 so $\bar{d} = 4$ for $T = m = 5$, then under the diametric attack, the patroller performing the patrol $\{1, 2, 3, 4, 5\}$ allows her to catch the two attacks. Hence the bound of $V \leq \frac{5}{8}$ given by lemma does not seem to hold.

Covering(Patroller) and Independence(Attacker):

Definition 1.11 (Covering)**.** A patrol, W, is called *intercepting* if it intercepts every possible attack at every node contained in the patrol, i.e all nodes visited by W are in any subpath of length m (i.e visits are at most m apart).

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

Definition 1.12 (Independence)**.** Two nodes, i and i', are called independent (under attack time, m) if any patrol intercepting an attack at i cannot also intercept an attack at i'.

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

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

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

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

Lemma 1.13 (Covering and Independence)**.**

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

1.2.2 Solved Graphs

We shall provide some information on graphs that have already been solved

Hamiltonian:

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

Definition 1.14 (Random Hamiltonian Patrol)**.** A *Random Hamiltonian Patrol* is a mixed strategy starting with equal probability at all nodes and following the Hamiltonian cycle.

Theorem 1.15 (Hamiltonian)**.** *If Q is Hamiltonian, by following the Random Hamiltonian Patrol (if feasible), the patroller can achieve $V \geq \frac{m}{n}$.*

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

Line:

A Line Graph, L_n, is a graph consisting of n nodes and $n-1$ edges connected in a straight line. While the line graph is complicated it has been solved across; Patrolling Games [**?**] and Patrolling a Border [**?**]. The cases in the solution require the division of the (n, m) space into sub-regions (inside which different strategies are adopted by the attacker and patroller).

However upon investigation one of the sub-regions and strategies does not produce the correct bound as reported. We bring this up as in section 2.1 we will provide a counter-example to show the error and provide an altered strategy giving the required bound.

Bipartite:

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

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

Definition 1.16 (Bipartite Attack)**.** The *Bipartite Attack* selects nodes equiprobably from the larger set B for a fixed time interval, I (or for the two time intervals, I and $I + 1$ equiprobably).

Theorem 1.17 (Bipartite)**.** *If Q is bipartite with $b \geq a$, by following the Bipartite Attack, the attacker can achieve $V \leq \frac{m}{2b}$*

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

Corollary 1.18 (Complete Bipartite)**.** *The value of the complete bipartite graph, $K_{a,b}$, with $b \geq a$, then $V = \frac{m}{2b}$*

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

1.3 Patrolling games with random attackers

A patrolling game is random attackers, $G = G(Q, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{c})$ is a minimizing game for the patroller. The parameters of the game are

- The graph $Q = (N, E)$ made of nodes, N ($|N| = n$), joined by edges, E, which can be represented by an adjacency matrix, A.
- A vector of attack time distributions, $\boldsymbol{X} = (X_1, ..., X_n)$.
- A vector of poisson arrival rates, $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_n)$.
- A vector of costs, $\boldsymbol{c} = (c_1, ..., c_n)$

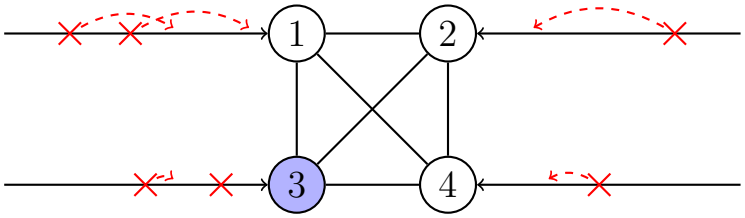


Figure 1.3: Example of $G = (K_4, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{c})$ with patroller currently at node 3

The attackers arrive at node, i, according a poisson process of rate λ_i, beginning their attack immediately, which lasts X_i time. The patroller detects all ongoing attacks when arriving at node i, the patroller then moves taking unit time to arrive at the next node (which can be the current node).

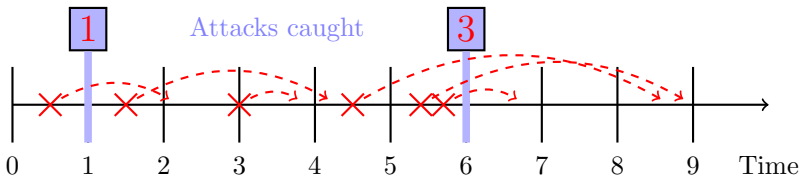


Figure 1.4: Example for a given node, when the patroller visits at times, $1, 5$

We can formulate such as problem as a Markov Decision Process(MDP) with state space, $\Omega = \{\boldsymbol{s} = (s_1, ..., s_n) \,|\, s_i = 1, 2, ... \text{ for } i = 1, ..., n\}$, where s_i denotes the time period since the decision to last visit node i was taken. Because the

patroller can only visit one node per time period, all s_i have distinct values. In particular, one s_i, the current node, has value 1. We can identify the current node by $l(\boldsymbol{s}) = \text{argmin}_i\, s_i$. The available decisions from state $\boldsymbol{s}$ are $\mathcal{A}(\boldsymbol{s}) = \{j | A_{l(\boldsymbol{s}),j} = 1\}$ and when node i is chosen by the patroller the state transitions to $\phi(\boldsymbol{s}, i) = \widetilde{\boldsymbol{s}}$, where $\widetilde{s}_i = 1$ and $\widetilde{s}_j = s_j + 1\ \forall j \neq i$.

Because the future of the process is independent of its past, it is only the current state that matters, we can formulate its movement as a Markov Chain(MC) and hence the patroller's problem is a MDP.

The patroller incurs costs for all attackers able to complete their attacks. For the decision the cost in the next time period is the sum of all costs incurred at all nodes, I.e $C(\boldsymbol{s}, i) = \sum_{j=1}^{j=n} C_j(\boldsymbol{s}, i)$, where $C_j(\boldsymbol{s}, i)$ is the cost at node j choosing to move to node i in the next time period. Hence

$$\begin{aligned} C_j(\boldsymbol{s}, i) &= c_j \lambda_j \int_0^{s_j} P(t-1 < X_j \leq t) dt \\ &= c_j \lambda_j \int_{s_j - 1}^{s_j} P(X_j \leq t) dt \end{aligned}$$

Note. $C_j(\boldsymbol{s}, i)$ is not dependent on i, the choice of i affects the future state (and hence future incurred costs)

With a countable infinite state space, Ω, problems of finding an optimal policy may exisit (See [Need reference from Peuterman]), so we bound the state space to make it finite. A reasonable assumption is to bound the attack times and define, the smallest interger for this bound

$$B_j \equiv \min\{k \,|\, k \in \mathbb{Z}^+, P(X_j \leq k) = 1\}$$

We now see that the cost function remains constant, $c_j \lambda_j$, for all $s_j \geq B_j + 1$ and so we restrict our state space to $s_j \leq B_j + 1$ and modify the transitions slightly so $\widetilde{s}_j = \min(s_j + 1, B_j + 1)\ \forall j \neq i$.

Now we can consider the objective function for our MDP, we wish to minimize the long-run average cost incurred. We also know by [reference to peuterman], that we can just focus on the class of stationary, deterministic policies $\Pi = \{\pi : \Omega \to N \,|\, \pi(\boldsymbol{s}) \in \mathcal{A}(\boldsymbol{s})\}$. So we wish to solve

$$C^{\text{OPT}}(\boldsymbol{s}_0) \equiv \min_{\pi \in \Pi} \sum_{i=1}^{n} V_i(\pi, \boldsymbol{s}_0)$$

Where $V_i(\pi, \boldsymbol{s}_0)$ is the long-run average cost incurred at node i under the policy, π, starting from state, $(\boldsymbol{s}_0)$ defined by ,

$$V_i(\pi, \boldsymbol{s}_0) \equiv \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} C_i(\phi_\pi^k(\boldsymbol{s}_0), \pi(\phi_\pi^k(\boldsymbol{s}_0)))$$

Where $\phi_\pi^k(\boldsymbol{s}_0)$ is the state after k transitions starting from $(\boldsymbol{s}_0)$ under the policy π.

Because the transitions are deterministic and the state space is finite, we know that $\phi_\pi^k(\boldsymbol{s}_0)$ will repeat and induce a cyclic behaviour under the policy,π. We will define the patrol pattern to be exactly this, from $\boldsymbol{s}_0$ let $\boldsymbol{s}_R$ be the first state which is repeated then define $\psi_\pi^k = \phi_\pi^k(\boldsymbol{s}_R)$, so the patrol pattern is $\{\psi_\pi^k \,|\, k = 0, 1, ..., K-1\}$. We say the patrol pattern is of length K. We can rewrite the long-run average cost at a node to be

$$V_i(\pi, \boldsymbol{s}_0) = \frac{1}{K} \sum_{k=0}^{K-1} C_i(\psi_\pi^k, \pi(\psi_\pi^k))$$

We will assume we are dealing with a connected graph, otherwise we solved the connected parts separately. Therefore, because every state is reachable we see that $C^{\mathrm{OPT}}(\boldsymbol{s}_0)$ does not depend on the initial state and so $C^{\mathrm{OPT}} = C^{\mathrm{OPT}}(\boldsymbol{s}_0)$ is the same for all initial states.

Note. If we set $c_i = 1\ \forall i$ then we can interest the long-run average cost, as the probability of not detecting an attack.

We can now use standard techniques such as value iteration or linear programming to compute the optimal policy and long-run average cost. This is left to Appendix B. However such methods are slow are only computable for a small graph, therefore we seek to create a near-optimal heuristic policy that can run in a shorter space of time.

1.3.1 Problem relaxation

We first relax the problem of the patroller only being able to visit one adjacent node each time period, to the *Multi-Node*(MN) problem, where the patroller can visit multiple nodes each time period. We will denote the class of stationary, deterministic policies as

$$\Pi^{\mathrm{MN}} = \{\pi^n : \Omega \to \boldsymbol{\alpha} \,|\alpha_i = 0, 1 \text{ for } i = 1, ..., n\}$$

For ease of notation we will define $\pi_i : \Omega \to \alpha_i$, that is the resultant element map, which we will use when we only care about a single node.

Note. Our previous un-relaxed policies, $\pi : \Omega \to N$ can be converted to a MN policy by mapping $\boldsymbol{s}$ to $\boldsymbol{\alpha}$ where $\alpha_i = 1$ for $i = \pi(\boldsymbol{s})$ and $\alpha_i = 0$ for $i \neq \pi(\boldsymbol{s})$ to form a policy $\pi^n \equiv \pi$.

Like before π^n will induce a patrol pattern, $\psi^k_{\pi^n}$ with length K'. We define the long-run average visit rate at which node, i is visited to be under π^n starting from $\boldsymbol{s}_0$ as

$$\mu_i(\pi_i, \boldsymbol{s}_0) = \frac{1}{K'} \sum_{k=0}^{K'-1} \pi_i(\psi^k_{\pi^n})$$

Now we restrict ourselves to having a maximum long-run average visit rate of one. This is know as the *Total-Rate*(TR) constraint. So we restrict ourselves to obey this constraint and the set of policies,

$$\Pi^{\mathrm{TR}} = \left\{ \pi^n \in \Pi^{\mathrm{MN}} \;\middle|\; \sum_{i=1}^{N} \mu_i(\pi_i, \boldsymbol{s}_0) \leq 1 \,,\, \forall \boldsymbol{s}_0 \in \Omega \right\}$$

We define the Problems optimal minimum cost, $C^{\mathrm{TR}} = \min_{\pi^n \in \Pi^{\mathrm{TR}}} \sum_{i=1}^{n} V_i(\pi^n)$, similarly to before.

Note. We have dropped the dependency on the initial state, $\boldsymbol{s}_0$, as even though $V_i(\pi^n, \boldsymbol{s}_0)$ depends on it, the long-run average cost does not. We will similarly drop the notation on the long-run visit rate, using $\mu_i(\pi^n)$.

Again, $\exists\, \pi^n \in \Pi^{\mathrm{TR}}$ s.t $\pi \equiv \pi^n$, so $C^{\mathrm{OPT}} \geq C^{\mathrm{TR}}$

Secondly we relax the MN problem by introducing the TR constraint as a Lagrangian multiplier.

2 Strategic Patroller and Strategic Attacker

2.1 Problem and correction to line graph strategy

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

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

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

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

Example 2.1 (Problem with diametric attack)**.** Consider the graph L_5 so $\bar{d} = 4$ for $T = 20$,$m = 6$, then under the diametric attack, the patroller oscillating between diametrics points gets.

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

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

End Capture 4 attacks when finishing at node 1.

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

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

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

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

Let us count the pattern under this strategy,

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

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

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

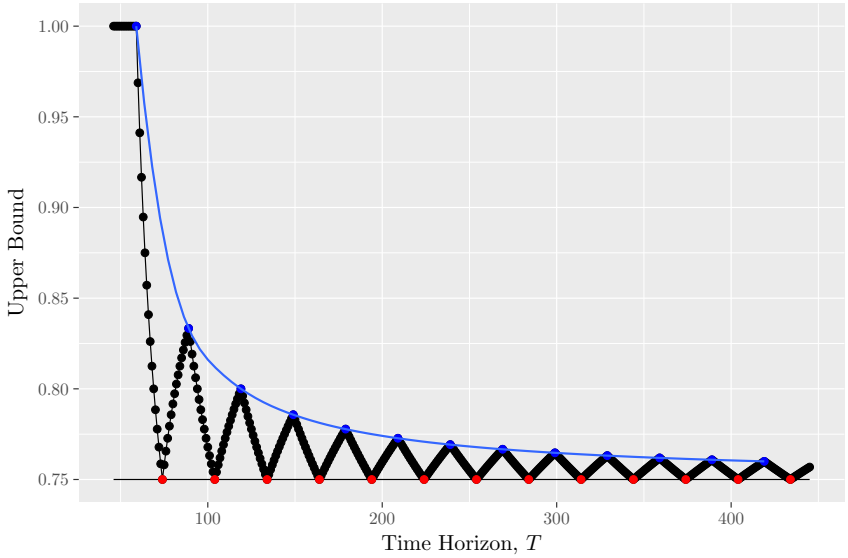


Figure 1: Best Upper Bound achievable under the diametric strategy

This gives

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

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

Lemma 2.2 (Condition on T for bound to hold)**.** *When $T = m - 1 + (k+1)\bar{d}$ for some $k \in \mathbb{N}_0$ then the diametric bound holds. Otherwise as $T \to \infty$ then the diametric bound holds.*

Proof: **??**

Fixing the diametric attack

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

Definition 2.3 (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.*

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

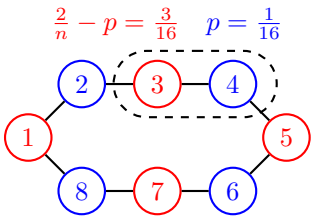
Lemma 2.4. *When $T \geq m + \bar{d} - 1$, the diametric bound $V \leq \max\{\frac{1}{2}, \frac{m}{2\bar{d}}\}$ is valid.*

Proof: **??**

2.2 New Tools

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

Definition 2.5 (Alternating Random Hamiltonian Patrol(ARHP))**.** An *Alternating Random Hamiltonian Patrol* (*ARHP*) is a mixed strategy following the Hamiltonian cycle but with a probability p of starting at "even" nodes and a probability of $\frac{2}{n} - p$ of starting at "odd" nodes.



Example Figure 2.1: C_8 with the blue nodes being "even" nodes started at with probability $\frac{1}{16}$ and the red nodes being "odd" nodes started at with probability $\frac{3}{16}$.

Lemma 2.6. *When n and m are both even, following the Alternating Random Hamiltonian Patrol, if feasible, gives the same lower bound as the random Hamiltonian patrol, i.e $V \geq \frac{m}{n}$.*

Proof. 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

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

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

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 **??**).

2.3 Star graph solution

As a special case of Complete bipartite 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 the Complete bipartite corollary(1.18)). This is achieved by the patroller forming a patrol which alternates between different external nodes and the centre, and the attacker attacking at all the external nodes with equal probability (for a fixed time interval).

2.4 Extending the star graph

2.5 Joining star graphs by centralised connections

3 Strategic Patroller with Random Attackers and Local-observations

3.1 Altering problem to accommodate local information

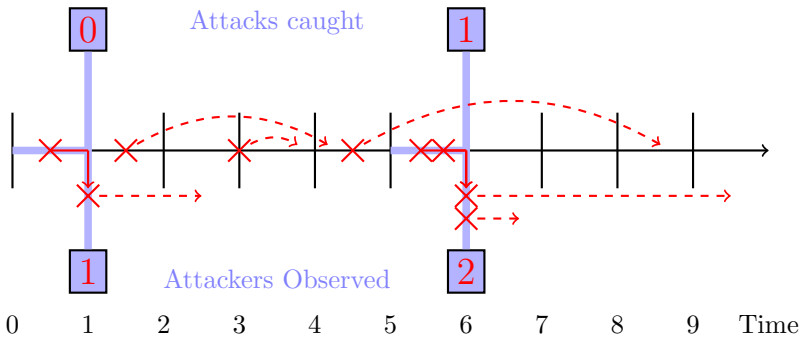


Figure 3.1: Example of timing

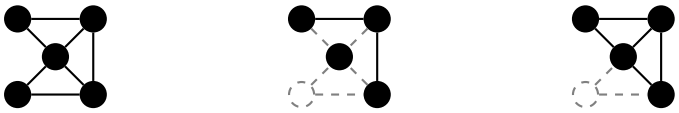
Appendices

A Graph Definitions

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

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

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



(a) Q (b) Q_1 (c) Q_1

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

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

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

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

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

Example A.5. For the graph Q as in Figure 2:

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

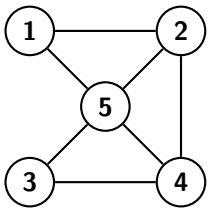


Figure 2: Graph, Q

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

Hence we would call the graph Q Hamiltonian.

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

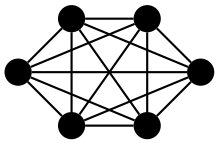
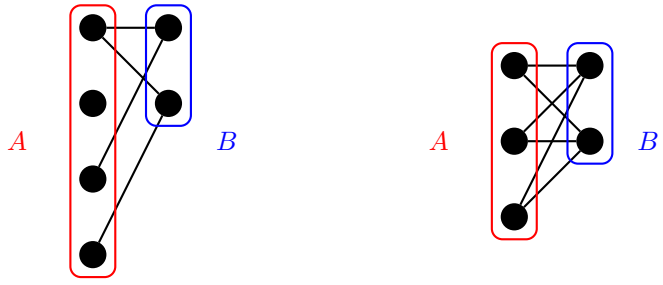


Figure A.2: The complete graph of 6 nodes,K_6.

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

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



(a) Q

(b) $K_{3,2}$

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

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

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

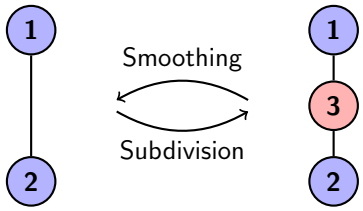


Figure A.4: Subdivision and Smoothing of the edge $\{1, 2\}$

B Optimal Solution for a Random Attacker