

A graph Patrol Problem with a random attacker on edges

Thomas Lowbridge
School of Mathematical Sciences
University of Nottingham

February 6, 2018

Contents

1	Introduction to a random attacker patroller game with attacks on edges	1
2	Relaxation	3
3	Single-edge problem	4
3.1	Acyclic graphs	5
4	Index Heuristic on Acyclic graphs	6
4.1	Lower bound	7
5	Removing Directional problem	8
6	Converting undirected problem to directed problem	9
	Appendices	i
A	Theorem on Index	i
B	Directional graphs with attacks on nodes	ii

1 Introduction to a random attacker patroller game with attacks on edges

The model has a undirected graph $Q = (N, E)$ with a set of nodes labelled 1 to n , $N = \{1, \dots, n\}$ and a set of edges linking the nodes which are a subset of $E_{comp} = \{(i, j) \mid i, j \in N\}$. The adjacency matrix $a = (a_{i,j})_{i,j \in N}$ has $a_{i,j} = 1$ if i and j are adjacent and $a_{i,j} = 0$ if they are not adjacent. By definition we will use $a_{i,1} = 1 \quad \forall i \in N$.

Note. We note that we have no edge (i, i) , but the node is still adjacent to its self. Also note that edge $(i, j) = (j, i)$ Hence we will assume that $i \leq j$ to uniquely describe and edge.

An attacker will arrive at some continuous position along an edge on the graph. The attackers arrive according to a Poisson process with rate Λ and will choose to attack edge (i, j) with probability $p_{i,j}$, hence by Poisson thinning we will arrive on the edge with a Poisson process with rate $\lambda_{i,j} = p_{i,j}\Lambda$. Once a node is picked the attacker will pick a position along the unit length line according to some distribution function $f_{i,j}(x)$ (with support $x \in [0, 1]$ and c.d.f $F_{i,j}(x)$). The length of an attack is a random variable $X_{i,j}$, which measures the amount of time before completion.

The patroller, will use some walk (with possible waiting) to partol the graph. We assume that the walk along an edge is uniform in real time, and that the edges are of unit length. So after making the decision of which node to move to they will arrive one unit later, with linear (continuous) movement along the edge. With each completing attack costing, $c_{i,j}$

This means that once an edge is picked, it is ‘renewed’ as normal, however their may be a residual amount of attacks left behind the patroller, while all currently commencing attacks that are not complete when the patroller reaches them are caught. Because an edge can be traversed i to j or j to i , it will matter which way we just traversed as to where the residual amount of attacker may be left.

So we will formulate the state space as the delineation of separate edges. $\Omega = \{(\mathbf{s}, \mathbf{d}) \mid s_{i,j} = 1, 2, \dots \quad d_{i,j} = 0 \text{ or } 1 \forall i, j \in N\}$, where $\mathbf{s} = (s_{i,j})_{(i,j) \in E}$ has $s_{i,j}$ represent the number of time periods since we last decided to traverse the edge (i, j) and $d_{i,j} = 0$ if we traversed it from left to right, i.e we traversed (i, j) otherwise if $d_{i,j} = 1$ if we traversed it from right to left i.e we traversed (j, i) .

The $s_{i,j}$ increment by 1 unless the edge was just traversed, in which case it then resets to 1. The $d_{i,j}$ remain unchanged unless the edge was just traversed, in which case it becomes 0 if we traversed i to j and 1 if we traversed j to i . The current node is found by analysing which of the $s_{i,j} = 1$ and if $d_{i,j} = 0$ (then we are at node j) or if $d_{j,i} = 1$ (then we are node i). Let $l(\mathbf{s}, \mathbf{d})$ represent the current node found by the rule above, i.e argument of the min and then decide which side by using direction.

As the future of the process is independent of its past, the process can be formulated as a Markov Decision Process(MDP), where at the end of the period

the patroller chooses which adjacent node to visit (and thus cover an edge). Thus the action space is $\mathcal{A} = \{j \mid a_{l(\mathbf{s}, \mathbf{d}), j} = 1\}$, with a deterministic stationary policy, $\pi : \Omega \rightarrow \mathcal{A}$.

The transitions are purely deterministic, (\mathbf{s}, \mathbf{d}) with the decision to move to node $i \in \mathcal{A}$ transitions the state to $(\tilde{\mathbf{s}}, \tilde{\mathbf{d}})$. Where $\tilde{s}_{k,j} = s_{k,j} + 1$ if $k \neq i, j \neq i$ and $\tilde{s}_{k,j} = 1$ if $k = i$ or $j = i$. Also $\tilde{d}_{k,j} = d_{k,j}$ if $k \neq i, j \neq i$ and $\tilde{d}_{k,j} = 0$ if $j = i$ and $\tilde{d}_{k,j} = 1$ if $k = i$.

To write down the cost function, which will be dependent on the state (\mathbf{s}, \mathbf{d}) and the action to visit node i is chosen, we will look at the expected cost incurred at all edges and sum these costs for the next time period.

The cost incurred in the next time period at edge (k, j) is

$$\begin{aligned}
C_{k,j}(\mathbf{s}, \mathbf{d}, i) &= \begin{cases} c_{k,j} \lambda_{k,j} \int_0^1 f_{k,j}(y) \left(\int_0^{s_{k,j}} P(t-1 < X_{k,j} \leq t) dt \right) dy & \text{If same direction} \\ c_{k,j} \lambda_{k,j} \int_0^1 f_{k,j}(y) \left(\int_0^{s_{k,j}+1-2y} P(t-1 < X_{k,j} \leq t) dt \right) dy & \text{If opposite direction} \end{cases} \\
&= \begin{cases} c_{k,j} \lambda_{k,j} E_Y \left[\int_0^{s_{k,j}} P(t-1 < X_{k,j} \leq t) dt \right] & \text{If same direction} \\ c_{k,j} \lambda_{k,j} E_Y \left[\int_0^{s_{k,j}+1-2Y} P(t-1 < X_{k,j} \leq t) dt \right] & \text{If opposite direction} \end{cases} \\
&= \begin{cases} c_{k,j} \lambda_{k,j} \int_0^{s_{k,j}} P(t-1 < X_{k,j} \leq t) dt & \text{If same direction} \\ c_{k,j} \lambda_{k,j} E_Y \left[\int_0^{s_{k,j}+1-2Y} P(t-1 < X_{k,j} \leq t) dt \right] & \text{If opposite direction} \end{cases} \\
&= \begin{cases} c_{k,j} \lambda_{k,j} \int_{s_{k,j}-1}^{s_{k,j}} P(X_{k,j} \leq t) dt & \text{If same direction} \\ c_{k,j} \lambda_{k,j} E_Y \left[\int_{s_{k,j}-2Y}^{s_{k,j}+1-2Y} P(X_{k,j} \leq t) dt \right] & \text{If opposite direction} \end{cases} \tag{1}
\end{aligned}$$

Note. We note that the cost depends on whether we are going to traverse the edge in the same direction as last time or not. The conditions formally are

- Same direction: $l(\mathbf{s}, \mathbf{d}) \leq i$ and $d_{l(\mathbf{s}, \mathbf{d}), i} = 0$ or $l(\mathbf{s}, \mathbf{d}) > i$ and $d_{l(\mathbf{s}, \mathbf{d}), i} = 1$
- Opposite direction: $l(\mathbf{s}, \mathbf{d}) \leq i$ and $d_{l(\mathbf{s}, \mathbf{d}), i} = 1$ or $l(\mathbf{s}, \mathbf{d}) > i$ and $d_{l(\mathbf{s}, \mathbf{d}), i} = 0$

With the total cost incurred being $C(\mathbf{s}, \mathbf{d}, i) = \sum_{(k,j) \in E} C_{k,j}(\mathbf{s}, \mathbf{d}, i)$

We will now assume that the attack time is bounded, i.e $X_{k,j} \leq B_{k,j}$, where $B_{k,j} = \min\{b \mid b \in \mathbb{Z}^+, P(X_{k,j} \leq b) = 1\}$. Now are cost functions are going to be the same for $s_{k,j} \geq B_{k,j} + 1$ we can restrict our state space to having $s_{k,j} \leq B_{k,j} + 1$, so $\Omega = \{(\mathbf{s}, \mathbf{d}) \mid s_{k,j} = 1, 2, \dots, B_{k,j} \text{ } d_{k,j} = 0 \text{ or } 1 \forall (k, j) \in E\}$ we a slightly modified transition that $\tilde{s}_{k,j} = \min\{s_{k,j} + 1, B_{k,j} + 1\}$ if $k \neq i, j \neq i$.

The objective of this MDP is to minimize the total long-run cost among all edges. As our state space and action space are finite it follows that we only need to consider deterministic stationary policies.

As our state transition is deterministic, so a deterministic stationary policy will induce a patrol pattern $\{\phi_\pi^m(\mathbf{s}_0, \mathbf{d}_0), m = 0, 1, 2, \dots\}$ where $\phi_\pi^m = \phi_\pi^{m-1} \circ \phi_\pi$ and because the state space is finite, the process will eventually revisit a previous state and then regenerate itself. Therefore applying a policy $\pi \in \Pi$ for an initial state $(\mathbf{s}_0, \mathbf{d}_0)$ we can write the long run cost of edge (k, j) as

$$V_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=0}^{N-1} C_{k,j}(\phi_\pi^m(\mathbf{s}_0, \mathbf{d}_0), \pi(\phi_\pi^m(\mathbf{s}_0, \mathbf{d}_0))) \quad (2)$$

We want to determine the optimal cost, over all edges, namely

$$C^{\text{OPT}}(\mathbf{s}_0, \mathbf{d}_0) = \min_{\pi \in \Pi} \sum_{(k,j) \in E} V_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0) \quad (3)$$

We will drop the dependence on the initial state, $(\mathbf{s}_0, \mathbf{d}_0)$ for C^{OPT} as from any initial state, as long as the graph is connected we can construct a feasible π to produce any patrol pattern, as $V_{k,j}$ depends entirely on the patrol pattern produced by π and $(\mathbf{s}_0, \mathbf{d}_0)$.

2 Relaxation

We will first relax the problem, to allow a larger class of policies allowing the patroller to travel on multiple edges (in a choice of direction) during each choice, as long as they on average choice to visit one or less edges. Let Π^{ME} denote the set of deterministic, stationary policies with $\pi : \Omega \rightarrow \{\boldsymbol{\alpha} \mid \alpha_{k,j} \in \{-1, 0, 1\}, (k, j) \in E\}$, where $\alpha_{k,j} = -1$ if and only if (k, j) is traversed from j to k , $\alpha_{k,j} = 0$ if and only if (k, j) is not traversed and finally $\alpha_{k,j} = 1$ if and only if (k, j) is traversed from k to j .

Because the same patrol pattern will repeat indefinitely, we can denote by $\mu_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0)$ the rate at which the edge (k, j) is traversed from k to j and similarly by $\nu_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0)$ the rate at which the edge (k, j) is traversed from j to k under the policy $\pi \in \Pi^{\text{ME}}$ with initial state $(\mathbf{s}_0, \mathbf{d}_0)$. This is just the number of directional visits to the edge divided by the length of the patrol pattern.

We now restrict ourselves to satisfy the total-rate constraint

$$\sum_{(k,j) \in E} \mu_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0) + \nu_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0) \leq 1, \quad \forall (\mathbf{s}_0, \mathbf{d}_0) \in \Omega \quad (4)$$

and we will form the collection of policies following the constraint as

$$\Pi^{\text{TR}} = \left\{ \pi \in \Pi^{\text{ME}} \left| \sum_{(k,j) \in E} \mu_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0) + \nu_{k,j}(\pi, \mathbf{s}_0, \mathbf{d}_0) \leq 1, \quad \forall (\mathbf{s}_0, \mathbf{d}_0) \in \Omega \right. \right\}$$

As previously discussed we can safely ignore the starting state, so for the future we will just write $V_{k,j}(\pi)$, $\mu_{k,j}(\pi)$ and $\nu_{k,j}(\pi)$. The relaxed problem can be formulated as

$$C^{\text{TR}} = \min_{\pi \in \Pi^{\text{TR}}} \sum_{(k,j) \in E} V_{k,j}(\pi) \quad (5)$$

and it follows, as $\Pi \subseteq \Pi^{\text{TR}}$ that $C^{\text{OPT}} \geq C^{\text{TR}}$

Secondly we relax the problem again, by incorporating the total-rate constraint into the objective function as a Lagrangian multiplier, $\omega \geq 0$ and becoming the Lagrangian relaxation

$$\begin{aligned} C(\omega) &= \min_{\pi \in \Pi^{\text{ME}}} \left\{ \sum_{(k,j) \in E} V_{k,j}(\pi) + \omega \left(\sum_{(k,j) \in E} \mu_{k,j}(\pi) + \nu_{k,j}(\pi) - 1 \right) \right\} \\ &= \min_{\pi \in \Pi^{\text{ME}}} \left\{ \sum_{(k,j) \in E} V_{k,j}(\pi) + \omega(\mu_{k,j}(\pi) + \nu_{k,j}(\pi)) \right\} - \omega \end{aligned} \quad (6)$$

As policies in Π^{TR} satisfy the total-rate constraint, we get that

$$\begin{aligned} C^{\text{TR}} &= \min_{\pi \in \Pi^{\text{TR}}} \sum_{(k,j) \in E} V_{k,j}(\pi) \geq \min_{\pi \in \Pi^{\text{TR}}} \left\{ \sum_{(k,j) \in E} V_{k,j}(\pi) + \omega \left(\sum_{(k,j) \in E} \mu_{k,j}(\pi) + \nu_{k,j}(\pi) - 1 \right) \right\} \\ &\geq \min_{\pi \in \Pi^{\text{ME}}} \left\{ \sum_{(k,j) \in E} V_{k,j}(\pi) + \omega \left(\sum_{(k,j) \in E} \mu_{k,j}(\pi) + \nu_{k,j}(\pi) - 1 \right) \right\} = C(\omega) \end{aligned}$$

So we have that $C^{\text{OPT}} \geq C^{\text{TR}} \geq C(\omega)$ for all $\omega \geq 0$.

Under the Lagrangian relaxation, we note that the problem splits up into a problem involving separate edges, with edge (k, j) wanting to minimize $V_{k,j}(\pi) + \omega(\mu_{k,j}(\pi) + \nu_{k,j}(\pi))$, where ω can be interpreted as the service charge for visiting in either direction. We now aim to solve this problem

3 Single-edge problem

Consider a single edge problem, namely the Lagrangian relaxation problem with the subscript removed. We aim to solve

$$\min_{\pi \in \Pi^{\text{ME}}} V(\pi) + \omega(\mu(\pi) + \nu(\pi)) \quad (7)$$

For this problem we consider a similar MDP as previously described but with the state just being how long since we last visited the edge and in which direction

we last visited it. Our policy for such a problem is going to be stationary and deterministic.

3.1 Acyclic graphs

We will now consider only Acyclic graphs, so that it makes sense to have to visit the edge in the opposite direction to the prior visit. This limits our decision to being binary, i.e wait or visit the edge. Then we can form an average cost for choosing to visit the edge every m time units,

$$f(m) = \frac{c\lambda E_Y \left[\int_0^{m+1-2Y} P(X \leq t) dt \right] + \omega}{m} \quad (8)$$

$$\begin{aligned} f(m+1) - f(m) &= \frac{1}{m(m+1)} \left(mc\lambda E_Y \left[\int_0^{m+2-2Y} P(X \leq t) dt \right] + m\omega \right. \\ &\quad \left. - \left((m+1)c\lambda E_Y \left[\int_0^{m+1-2Y} P(X \leq t) dt \right] + (m+1)\omega \right) \right) \\ &= \frac{1}{m(m+1)} \left(c\lambda \left(mE_Y \left[\int_{m+1-2Y}^{m+2-2Y} P(X \leq t) dt \right] - E_Y \left[\int_0^{m+1-2Y} P(X \leq t) dt \right] \right) - \omega \right) \end{aligned}$$

Thus as the patroller will be indifferent from choosing every $m+1$ over m when this is zero, it suggests an index

$$W(m) = c\lambda \left(mE_Y \left[\int_{m+1-2Y}^{m+2-2Y} P(X \leq t) dt \right] - E_Y \left[\int_0^{m+1-2Y} P(X \leq t) dt \right] \right) \quad (9)$$

We will now take notice that if $m \geq B+1$ then we get a reduction of the index

to

$$\begin{aligned}
W(m) &= c\lambda \left(mE_Y[1] - E_Y \left[\int_0^{m+1-2Y} P(X \leq t) dt \right] \right) \\
&= c\lambda \left(1 - 2Y + E_Y \left[\int_0^{m+1-2Y} 1 - P(X \leq t) dt \right] \right) \\
&= c\lambda \left(E_Y \left[1 - 2Y + \int_0^{m+1-2Y} P(X > t) dt \right] \right) \\
&= c\lambda \left(1 + E_Y \left[-2Y + \int_0^B P(X > t) dt \right] \right) \\
&= c\lambda \left(1 + E_Y[-2Y + E[X]] \right) = c\lambda(1 + E[X - 2Y]) \quad (10)
\end{aligned}$$

Note. We may want to check that this is never negative. As $Y \leq 1$ we note that $2E[Y] \leq 2$ and so we just require that $1 + E[X] \geq 2 \implies E[X] \geq 1$. It is very clear that if we have a mean completion time of less than 1, then we will never expect to catch anyone.

We will now attempt to prove 3 statements, as in the theorem

Theorem 3.1. *For $W(m)$ as described above:*

- $W(m)$ is non-decreasing
- if $\omega \in [W(m-1), W(m)]$ then the optimal policy is to visit every m times, for $m = 0, 1, 2, \dots$
- Moreover if $\omega \geq c\lambda(1 + E[X - 2Y])$ then it optimal to never visit at all

The proof is in Appendix A

4 Index Heuristic on Acyclic graphs

We use the definition of the index, as suggested, affixing an edge subscript. We implement it as the index when edge (k, j) has not been visited for m time periods (i.e decision to last traverse the edge occurred m time units ago)

$$W_{k,j}(m) = c\lambda \left(mE_{Y_{k,j}} \left[\int_{m+1-2Y_{k,j}}^{m+2-2Y_{k,j}} P(X_{k,j} \leq t) dt \right] - E_{Y_{k,j}} \left[\int_0^{m+1-2Y_{k,j}} P(X_{k,j} \leq t) dt \right] \right) \quad (11)$$

To implement a heuristic based on this index, called the *Index-Heuristic*(IH), we choose the initial node, by assuming that the graph has been neglected for a

long time. Setting $S_{k,j} = B_{k,j} + 1$ initially and choosing the edge with the with the highest index.

??? Once an edge is selected, the initial direction needs to be selected- This feels like it should just depend on the distributions skew to not leave as many possible attackers behind ???

Once an edge and direction are selected, the patroller looks at all adjacent edges (or equivalently nodes) and calculates the indices of these edges to decide which one to traverse, by choosing the highest. Mathematically when in state \mathbf{s} (Note. No directional information is kept, as we know we are always traversing in the opposite direction to before) we will choose edge $(l(\mathbf{s}), j)$ where $W_{l(\mathbf{s}),j}(s_{l(\mathbf{s}),j}) = \min_{j' \in \mathcal{A}(\mathbf{s})} W_{l(\mathbf{s}),j'}(s_{l(\mathbf{s}),j'})$

4.1 Lower bound

Recall that $C^{\text{OPT}} \geq C^{\text{TR}} \geq C(\omega)$. In this section we will compute the tightest bound so that $C^{\text{TR}} = \max_{\omega \geq 0} C(\omega)$.

As $W_{k,j}(m)$ is the per-visit cost which makes the patroller indifferent from choosing to visit the edge (k, j) every m or $m + 1$ time units. So for a per-visit cost define

$$M_{k,j}(\omega) = \begin{cases} \infty & \text{if } \omega \geq W_{k,j}(B_{k,j} + 1) \\ \min\{m \mid W_{k,j}(m) > \omega\} & \text{otherwise} \end{cases} \quad (12)$$

From theorem 3.1 this $M_{k,j}(\omega)$ represents the optimal interval between visits to edge (k, j) when the patrol visit costs ω . According to this definition there are multiple optimal intervals, so we will break ties by choosing the longest interval.

Consequently we can write that $C(\omega) = \sum_{(k,j) \in E} C_{k,j}(\omega) - \omega$ where

$$\begin{aligned} C_{k,j}(\omega) &= f_{k,j}(M_{k,j}(\omega)) \\ &= \begin{cases} c_{k,j} \lambda_{k,j} & , \text{if } \omega \geq W_{k,j}(B_{k,j} + 1) \\ c_{k,j} \lambda_{k,j} E_Y \left[\frac{\int_0^{M_{k,j}(\omega)} P(X_{k,j} \leq t) dt}{M_{k,j}(\omega)} \right] + \omega & , \text{otherwise} \end{cases} \end{aligned}$$

The function $C_{k,j}(\omega)$ represents the optimal long-run cost for edge (k, j) if the patroller is charged ω per visit. Now we have that

- $C_{k,j}(\omega)$ is non-decreasing in ω , as the edge can always do better with a smaller service charge.

- $C_{k,j}(\omega)$ is piecewise linear with turning points occuring at $\omega = W_{k,j}(m)$ for $m = 1, 2, \dots, B_{k,j} + 1$
- $C_{k,j}(\omega)$ is concave as for $\omega \neq W_{k,j}(m)$ the function $M_{k,j}(\omega)$ remains constant and $C'_{k,j}(\omega) = \frac{1}{M_{k,j}(\omega)}$ which is non-increasing in ω

Consequently $C(\omega) = \sum_{(k,j) \in E} C_{k,j}(\omega) - \omega$ is also piecewise linear and concave.

Therefore it is straightforward to calculate $\max_{\omega \geq 0} C(\omega)$, as the optimal solution is either a point or a line segment. When $\omega \neq W_{k,j}(m)$ we have $C'(\omega) = \sum_{(k,j) \in E} \frac{1}{M_{k,j}(\omega)} - 1$. That is $C'(\omega)$ is a step function that changes at values of $W_{k,j}(m)$ for some $(k,j) \in E, m$ and is non-increasing. In the case that the optimal solution is unique, named ω^* such that

$$\begin{aligned} \omega < \omega^* &\iff C'(\omega) > 0 \iff \sum_{(k,j) \in E} \frac{1}{M_{k,j}(\omega)} > 1, \\ \omega > \omega^* &\iff C'(\omega) < 0 \iff \sum_{(k,j) \in E} \frac{1}{M_{k,j}(\omega)} < 1. \end{aligned}$$

In the case where the optimal solution consists of a line segment we need to find ω^* s.t $C'(\omega^*) = 0 \iff \sum_{(k,j) \in E} \frac{1}{M_{k,j}(\omega^*)} = 1$

5 Removing Directional problem

An idea to remove the problem with the direction for an edge is to assume all the edges are directed, that is the edge (k,j) can only be used from k to get to j . Then we simply have a state space $\Omega = \{\mathbf{s} \mid s_{k,j} = 1, 2, 3, \dots \forall (k,j) \in E$. That are state space is just dependent on what time we last went to visit it. Hence a very simple transition

Note. We no longer assume that (k,j) has $k \leq j$, as we now allow the full gambit of combinations.

This will lead to a slightly easier cost function to work with

$$\begin{aligned} C_{k,j}(\mathbf{s}, i) &= c_{k,j} \lambda_{k,j} \int_0^1 f_{k,j}(y) \left(\int_0^{s_{k,j}} P(t-1 < X_{k,j} \leq t) dt \right) dy \\ &+ = c_{k,j} \lambda_{k,j} \int_0^{s_{k,j}} P(t-1 < X_{k,j} \leq t) dt \\ &= c_{k,j} \lambda_{k,j} \int_{s_{k,j}-1}^{s_{k,j}} P(X_{k,j} \leq t) dt \end{aligned}$$

This leads us to suggest that it will follow the same idea as the decision on when to visit nodes, and we can use the edge-vertex dual to convert between these two problems

Definition 5.1 (Edge-vertex dual). A *edge-vertex dual* of a directed graph G , called $EV(G)$, is made of a vertex set $V_{EV(G)} = E_G$ and whose edge set is made up of a directed edge between $e_1, e_2 \in V_{EV(G)}$ if in G the edge e_1 's head meets the tail of e_2 at some node.

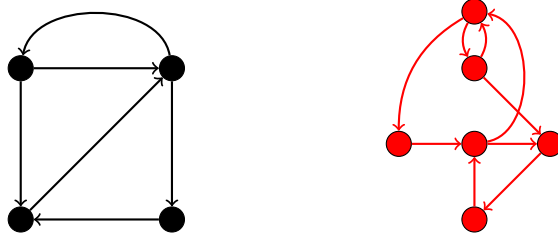


Figure 5.1: A graph G and its directed edge-vertex dual, $EV(G)$

This gives us a direction conversion between the current problem and a related problem to that which has already been solved. See Appendix B for an explanation of the slight alteration to the directional graphs with random attackers on nodes problem.

Now we have converted the problem into one which can be ‘solved’ by using the already developed heuristic, with the same index idea as the cost function for an edge (k, j) is the same as the node i which represents the edge in our edge-vertex dual.

Hence for we have a solution for our problem, convert to the edge-vertex dual and then solve as before.

Note. The true problem with have an undirected graph in the edge attacker set up is that the patroller can use it in both directions. The question is now can we convert the undirected graph to a directed graph with the same properties.

6 Converting undirected problem to directed problem

We now look at converting an undirected edge (k, j) with $k \leq j$ into two directed edges (k, j) and (j, k) . Well as the Poisson process for arrivals on this edges is $\lambda_{k,j}$, we can thin this between the two lines so that the arrival on the forward direction is $\overrightarrow{\lambda}_{k,j} = \frac{\lambda_{k,j}}{2}$ and the arrival on the backwards direction is $\overleftarrow{\lambda}_{k,j} = \frac{\lambda_{k,j}}{2}$.

Now for the forward direction we will assume the distribution of arrivals holds true, i.e $\overrightarrow{f}_{k,j}(y) = f_{k,j}(y)$, however on the backwards we need to flip the distri-

bution, as this arrow will want to see it forward for it (which is backwards for the forward) so $\overleftarrow{f_{k,j}}(y) = f_{k,j}(1 - y)$.

Note. This means if the distribution $f_{k,j}(y)$ is symmetric then we will have the same distribution on the conversion.

Now we can perform this on each edge (k, j) and then convert to the edge-vertex dual and use the already developed heuristic, however we may run into an issue on equivalence to our original problem.



Figure 6.1: Converting an undirected edge to directed

The problem with this type of conversion is that the attackers on the edges now think of them as unique whereas in the initial undirected they are not unique to the direction you traverse the edge. A simple idea to solve this problem could be to add up the indices on the two edges (However at first glance this would only seem to work if the distribution is symmetric, due to the attackers positioning)

For now we will assume that the distribution is symmetric, then summing the indices created in the edge-vertex dual's nodes (the pair of nodes that correspond to a single edge in the original problem) may allow us to make a decision if we are using the simple heuristic of which edge to traverse now, as both of these will be incurred if we allow the process to continue. This is the same as just playing on the graph's edge-vertex dual, without directional problems.

Appendices

A Theorem on Index

Proof. We will prove each statement separately

- To see its non-decreasing observe that

$$W(m+1) - W(m) = c(m+1)\lambda \left(E_Y \left[\int_{m+2-2Y}^{m+3-2Y} P(X \leq t) dt - \int_{m+1-2Y}^{m+2-2Y} P(X \leq t) dt \right] \right) \geq 0$$

we note that the fact that it is greater than 0 is due to the fact that $P(X \leq t)$ is a non-decreasing function.

To derive that m is optimal for $\omega \in [W(m-1), W(m)]$ we shall show that the objective function, $f(m)$ is non-increasing for $m' \leq m$ and non-decreasing for

$m' \geq m$. Now

$$f(m') - f(m'-1) = \frac{1}{m'(m'-1)}(W(m'-1) - \omega) \leq \frac{1}{m'(m'-1)}(W(m'-1) - W(m)) \leq 0$$

and

$$f(m'+1) - f(m') = \frac{1}{m'(m'+1)}(W(m') - \omega) \geq \frac{1}{m'(m'+1)}(W(m') - W(m)) \geq 0$$

Finally we note that $f(m+1) \leq f(m) \iff \omega \geq W(m)$. It is not optimal to visit at all if $f(m+1) \leq f(m) \forall m$, or equivalently if

$$\omega \geq W(m) \forall m \iff \omega \geq \sup_{m=0,1,\dots} W(m) = \lim_{m \rightarrow \infty} W(m) = c\lambda(1 + E[x - 2Y])$$

□

B Directional graphs with attacks on nodes

Consider the standard problem, the only thing we need to note is that the adjacency matrix need not be symmetric, and this means actions are limited, however the developed heuristic still remains the same.

We will however assume that the directional graph has no 'closed states', i.e that every state is adjacent to some other state (other than itself), this is however included in the assumption that the graph is connected as no node should be unreachable from any node.