A Graph Patrol Problem with Locally-Observable Random Attackers

Thomas Lowbridge and David Hodge

University Of Nottingham,UK

June 15, 2018

Outline

- Literature review
  - Introduction to Game
    - Pure game
    - Mixed game
  - Solved Graphs
    - Hamiltonian graphs
    - Line graph
- Problem with line graph strategy
- Correction of line graph strategy
- Extension of correction strategy
- Future work

Game setup

A Patrolling game with random attackers, $G = G(Q, \boldsymbol{X}, \boldsymbol{b}, \boldsymbol{\lambda}, \boldsymbol{c})$ is made of 4 major components.

- A Graph, $Q = (N, E)$, made of nodes, N ($|N| = n$), and a set of edges, E and an adjacency matrix, A.
- A vector of observable capacities, $\boldsymbol{b} = (b_1, ..., b_n)$
- 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, $\boldsymbol{c} = (c_1, ..., c_n)$

Game setup

A Patrolling game with random attackers, $G = G(Q, \boldsymbol{X}, \boldsymbol{b}, \boldsymbol{\lambda}, \boldsymbol{c})$ is made of 4 major components.

- A Graph, $Q = (N, E)$, made of nodes, N ($|N| = n$), and a set of edges, E and an adjacency matrix, A.
- A vector of observable capacities, $\boldsymbol{b} = (b_1, ..., b_n)$
- 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, $\boldsymbol{c} = (c_1, ..., c_n)$

The game is played over an infinite time horizon, $\mathcal{T} = 0, 1, ...$
A patroller's policy in the game is a walk (with waiting) on the graph,

$$W : \mathcal{T} \to N.$$

With the patroller moving instantaneously between nodes and attackers who arrive when the patroller is present waiting till the next time period to attack(who are observed as suspicious by the patroller).

Markov Decision Process(MDP) formulation

This game is a MDP with states $(\boldsymbol{s}, \boldsymbol{v})$, where s_i is the time since node i was last chosen to be visited and v_i is how many local-observations where observed when node i was last visited.
The current node can be identified by $l(\boldsymbol{s}) = \text{argmin}_i \, \boldsymbol{s}$. The current node will have $s_i = 1$.
Like in [Insert reference to paper], we will bound the attack times to create a finite state space, and define
$B_j \equiv \min\{k | k \in \mathbb{Z}^+, P(X_j \leq k) = 1\}$. Initially we assume that $b_i \in \mathbb{Z}_0^+ \cup \{\infty\}$, but to make a finite state space we will assume that b_i is finite.
$\Omega = \{(\boldsymbol{s}, \boldsymbol{v}) | s_i = 1, 2, ..., B_i + 1 \text{ and } v_i = 0, 1, ..., b_i \, \forall i = 1, ..., n\}$.

Making state space finite

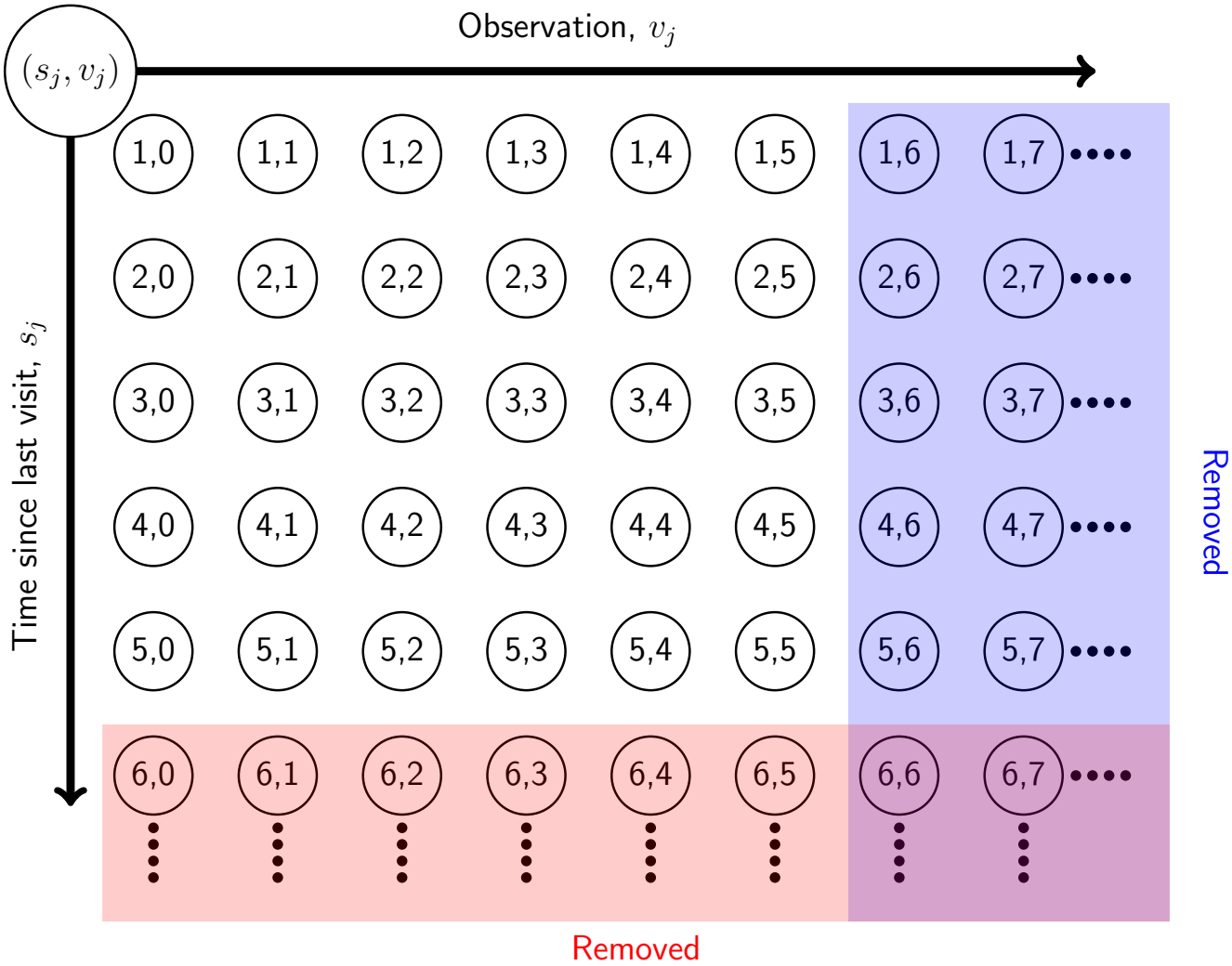

Figure 1: State space diagram, with $b_j = 5$ and $B_j = 4$ (e.g. $X_j \leq 3.7$)

Markov Decision Process(MDP) formulation

The available actions from any state is $\mathcal{A}(\boldsymbol{s}) = \{j | A_{l(\boldsymbol{s}),j} = 1\}$. A transition from a state with a chosen action is $\phi(\boldsymbol{s}, \boldsymbol{v}, i) = (\widetilde{\boldsymbol{s}}, \widetilde{\boldsymbol{v}})$ where $\widetilde{s}$ has; $\widetilde{s}_i = 1$ and $\widetilde{s}_j = \min\{s_j + 1, B_j + 1\} \forall j \neq i$ and $\widetilde{v}$ has; $\widetilde{v}_i \sim TPo(\lambda_i, b_i)$ and $\widetilde{v}_j = v_i \forall j \neq i$.
Where $TPo(\lambda, b)$ is the Poisson distribution truncated at the value b. I.e

$$P(TPo(\lambda, b) = \begin{cases} P(Po(\lambda) \text{ if } i \neq b \\ P(Po(\lambda) \geq i) \text{ if } i = b \\ 0 \text{ Otherwise} \end{cases}$$

Markov Decision Process(MDP) formulation

The cost at a node is zero if that node is chosen, otherwise it is the cost of arrivals finishing in the next time period, plus the cost of local-observations finishing in the next time period.

$$C_j(\boldsymbol{s}, \boldsymbol{v}, i) = \begin{cases} c_j \lambda_j \int_{s_j - 1}^{s_j} P(X_j \leq t) dt \\ + c_j v_j P(s_j - 1 < X_j \leq s_j) \text{ if } j \neq i \\ 0 \text{ if } j = i \end{cases}$$

.

Due to the finiteness of the state space, we can just focus on stationary, deterministic policies, $\pi : \Omega \to \mathcal{A}$ collected into Π, the set of stationary, deterministic policies.

Markov Decision Process(MDP) formulation

The patroller wants to choose a policy such that the long-run average cost is minimized. I.e. they want to find

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

where,

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

Where $V_i(\pi, \boldsymbol{s}_0, \boldsymbol{v}_0)$ is the long-run average cost incurred at node i under the policy, pi and $\phi_\pi^k(\boldsymbol{s}_0, \boldsymbol{v}_0)$ is the state after k transitions starting from $(\boldsymbol{s}_0, \boldsymbol{v}_0)$ under the policy π.

Problem Relaxation: Multiple Node

We now relax the problem to that of a patroller who can visit multiple nodes in the next timer period.
We will call this Problem the *multiple node* (MN) problem. To extend ourselves to this problem we will create a new class of polices, Π^{MN} which will allow the patroller to visit any set of nodes during the next time period.

$$\pi : \Omega \rightarrow \{\boldsymbol{\alpha} \,|\, \alpha_i \in \{0, 1\} \text{ for } i = 1, ..., n\}$$

Where $\alpha_i = 1$ if the patroller will visit node i in the next time period and $\alpha_i = 0$ if the patroller will not visit node i in the next time period. Note. $\Pi \subset \Pi^{\text{MN}}$.

Total-rate constraint

We now define the long-run rate at which the patroller visits node i, following π

$$\mu_i(\pi, \boldsymbol{s}_0, \boldsymbol{v}_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \alpha_{\psi_\pi^k(\boldsymbol{s}_0,\boldsymbol{v}_0),i}$$

We now impose the *total-rate* constraint, that is the long-run overall visit rate to all nodes is no greater than 1.

$$\sum_{i=1}^{n} \mu_i(\pi, \boldsymbol{s}_0, \boldsymbol{v}_0) \leq 1 \quad \forall(\boldsymbol{s}_0, \boldsymbol{v}_0) \in \Omega$$

and we denote the set of policies, which satisfy this constraint by Π^{TR} Again $\Pi \subset \Pi^{\mathsf{TR}}$ and we seek to find C^{TR}.

Lagrangian relaxation

We now relax the problem again by incorporating the total rate constraint into the objective function, with a Lagrange multiplier, $\omega \geq 0$. This forms

$$C(\omega) = \min_{\pi \in \Pi^{\mathsf{MN}}} \left\{ \sum_{i=1}^{n} V_i(\pi) + \omega \left(\sum_{i=0}^{n} \mu_i(\pi) - 1 \right) \right\}$$
$$= \min_{\pi \in \Pi^{\mathsf{MN}}} \sum_{i=1}^{n} (V_i(\pi) + \omega \mu_i(\pi)) - \omega$$

By incorporating the total-rate constraint as a Lagrange multiplier we can drop the constraint, so that the patroller can choose to visit any number of nodes in each time period (admittedly costing ω).

Linking reduced problems

We have that for any $\omega \geq 0$

$$C^{\mathsf{TR}} = \min_{\pi \in \Pi^{\mathsf{TR}}} \sum_{i=1}^{n} V_i(\pi) \geq \min_{\pi \in \Pi^{\mathsf{TR}}} \left\{ \sum_{i=1}^{n} V_i(\pi) + \omega \left(\sum_{i=0}^{n} \mu_i(\pi) - 1 \right) \right\}$$
$$\geq \min_{\pi \in \Pi^{\mathsf{MN}}} \left\{ \sum_{i=1}^{n} V_i(\pi) + \omega \left(\sum_{i=0}^{n} \mu_i(\pi) - 1 \right) \right\} = C(\omega)$$

The first inequality follows as the total-rate constraint is obeyed in Π^{TR} and hence the $\omega \left(\sum_{i=0}^{n} \mu_i(\pi) - 1 \right) \leq 0$ and the second holds, as the total rate constraint is dropped and we have got a bigger state space, so we can only do better and hence possibly get a lower value.
Hence we have a string of inequalities $C^{\mathsf{OPT}} \geq C^{\mathsf{TR}} \geq C(\omega)$.

Single node problem

We now want to find $C(\omega) = \min_{\pi \in \Pi^{\text{MN}}} \sum_{i=1}^{n} (V_i(\pi) + \omega \mu_i(\pi)) - \omega$ and we can split this problem up to just solve at each node.
That is consider a every node, i, try to minimize $V_i(\pi) + \omega \mu_i(\pi)$, where ω can be interpreted as the service charge for visiting.
For now we drop the node subscript and just try to find the following

$$\min_{\pi \in \Pi^{\text{MN}}} V(\pi) + \omega \mu(\pi)$$

Deterministic attack times

Before looking at solving the single node problem, we will assume the attack times are deterministic, to make the cost function easier to handle. When we use $X_j = x_j$ we get a cost function of.
For $s_j < B_j$

$$C_j(\boldsymbol{s}, \boldsymbol{v}, i) = 0$$

For $s_j = B_j$

$$C_j(\boldsymbol{s}, \boldsymbol{v}, i) = \begin{cases} c_j \lambda_j R_j + c_j v_j \text{ for } i \neq j \\ 0 \text{ for } i = j \end{cases}$$

For $s_j = B_j + 1$

$$C_j(\boldsymbol{s}, \boldsymbol{v}, i) = \begin{cases} c_j \lambda_j \text{ for } i \neq j \\ 0 \text{ for } i = j \end{cases}$$