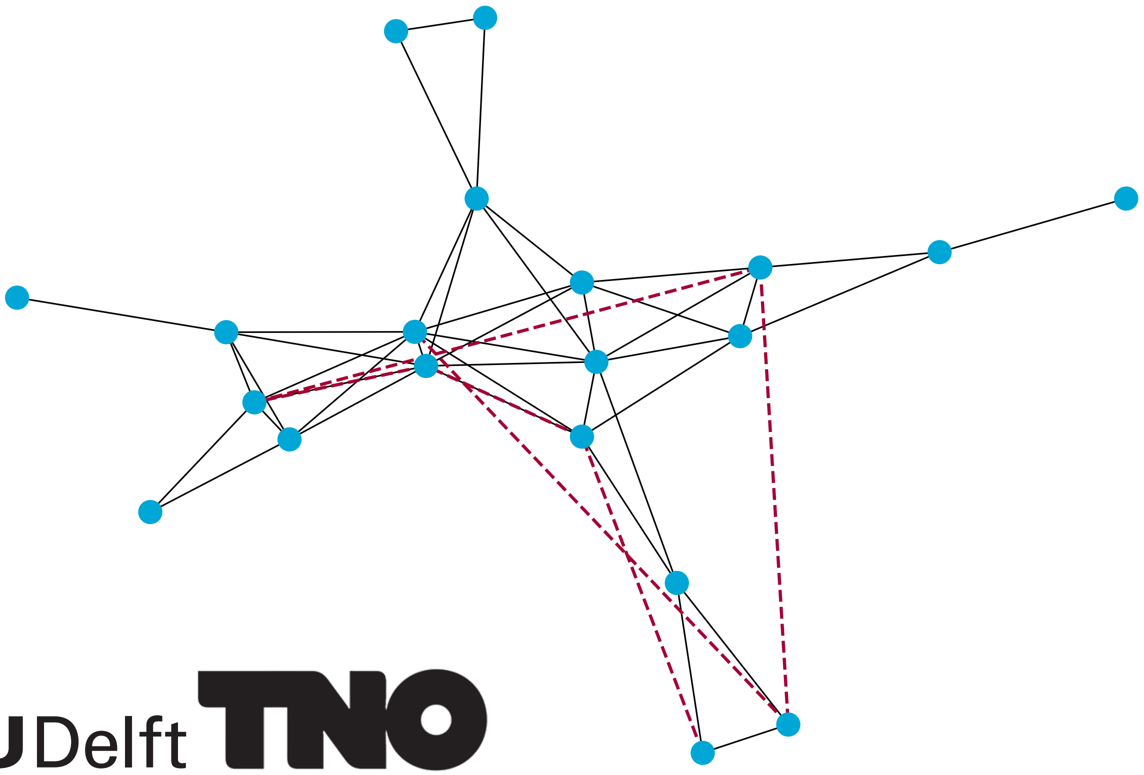
At a crossroads

T. Koning

Optimizing Markovian patrol strategies for perimeter defense



At a crossroads

by

T. Koning

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Wednesday July 17, 2024 at 10:00 AM.

Project duration:	April 22, 2024 – July 17, 2024	
Thesis committee:	Dr. ir. L. J. J van Iersel	TU Delft, supervisor
	C. J. Konings, MSc	TNO, supervisor
	Dr. J. G. Spandaw	TU Delft

Abstract

The security of sensitive areas against adversarial threats is a critical concern, necessitating the development of effective patrol strategies. This thesis addresses the problem of optimal patrolling in adversarial scenarios through the formulation of an analytical method to calculate the interception probability of a possible attacker in a graph-based patrolling game. By leveraging Markovian strategies, the research provides a robust framework for efficiently calculating interception probabilities and proposes a methodology to optimize patrol routes.

Firstly, the structure and modeling framework that is used in this thesis is set out. Subsequently, a probabilistic, a recursive, and a matrix-product method are derived to calculate the interception probability. This constitutes the core of this thesis. Lastly, the discussion is extended by applying the matrix-product method to practical examples. Monte Carlo simulations are used to compare the performance of different patrol strategies under varying conditions. The results illustrate how the developed methods enable detailed performance analyses, showing that the effectiveness of patrol strategies can vary based on the distribution of attack strategies.

Key findings highlight the impact of graph structure and attack strategy distributions on interception probabilities, the scalability and practicality of the matrix product method to calculate the interception probability, and the significant computational efficiency gained by using this approach. The research also outlines several avenues for future work, including the exploration of heterogeneous environments, optimization improvements, and multi-agent scenarios. Overall, this thesis contributes to the field of patrolling games by offering enhanced methods to evaluate and optimize Markovian patrol strategies, providing both theoretical methodologies and practical implementations that can be used to improve security in adversarial settings.

Contents

Glossary

$G = (V, E)$	The graph G with vertices V and edges E on which the patrolling game is played.
$\mathbf{A}$	The adjacency matrix of graph G. So, $a_{i,j} = 1$ if graph G has an edge from vertex i to vertex j and $a_{i,j} = 0$ otherwise.
n	The number of vertices in G, i.e. $n = \|V\|$.
$(X_t)_{t\geq 0}$	The discrete-time Markov chain underlying the Markov patrol.
$\mathbf{Q}$	The transition matrix corresponding to the Markov process $(X_t)_{t\geq 0}$. The entries are denoted as $q_{i,j}$.
$\boldsymbol{\mathcal{P}}_n$	The matrix containing the probabilities of visiting a state at least once in n steps plus the starting position. So, $(p_n)_{i,j} = \mathbb{P}(X_0 = j \cup \ldots \cup X_n = j \mid X_0 = i)$.
$\mathbf{I}$	The identity matrix.
d_p	The duration of the patrol in time steps. This includes the starting point, so a patrol includes the starting point and $d_p - 1$ movements.
d_a	The duration of the attack in time steps. It is assumed that $1 \leq d_a \leq d_p$.
$\boldsymbol{\varkappa}$	The probability distribution over the starting positions of the patrol. So, $\varkappa_i = \mathbb{P}(X_0 = i)$.
$\mathbf{Y} \in \mathbb{R}^{n\times d_p - d_a + 1}$	The attack probability distribution. So, $y_{v,t} = \mathbb{P}(\text{Attack on vertex } v \text{ starts at time } t)$.
$\mathcal{U}$	The probability that an attack is intercepted for a given patrol strategy and a given attack strategy defined on a graph G for a certain d_a and d_p.

1 Introduction

"Defending for intentional threats has been a central and longstanding focus for governments throughout the world", as Hunt and Zhuang so aptly state [1]. Although cyberspace security has become increasingly more important in recent years, physical security remains a crucial cornerstone of modern societies. Nevertherless, it remains difficult, if not impossible, to guarantee infallible safety. In general, there is too much to protect and too few resources to protect it with. Perhaps, if intelligence and security services could perfectly predict where and when each future attack would take place, the scarce resources could be placed just right. Such accurate prediction, of course, is an impossible task. In the case of patrolling, this means that not every part of the area of responsibility can be guarded at all times. A mathematical model can provide insight into the optimal choice of patrolling strategy when the ratio of resources to responsibilities is constrained.

In this paper, it is assumed that there are two adversarial parties and a single contained area of interest. One side controls the area and wishes to protect it against the adversary. Meanwhile, the other party seeks to either infiltrate or attack the same area. The problem is viewed from the perspective of the defending side. Patrolling the area is a possible security measure. For example, Figure 1.1 shows a graph representation of the 7th arrondissement of Paris, where the Eiffel Tower is located. This area is frequently targeted by individuals with disruptive, and sometimes even destructive intentions. Yet, without additional intelligence, it is unclear exactly when and where possible adversaries might want to attack. The core question then is how an optimal patrol strategy can be found when the location and time of the attack are unknown. In the literature, this is commonly referred to as a *patrolling game*.

Patrolling games have been extensively studied. In 2011, Alpern and Papadaki created a mathematical framework for modeling patrolling games and provided some analytical solutions [2]. Their model uses a graph representation of the environment with discrete time steps. Within this environment are a single patroller and a single attacker. An attack has a fixed duration and location, and the attacker is free to choose a place and time to attack. The attack fails if the patroller moves through the attack area while the attack is underway and succeeds otherwise. However, Alpern and Papadaki only derive analytical intercept probabilities for very small instances of the problem, i.e. for graphs with only a few vertices.

Subsequent papers relaxed certain assumptions or simplifications made by Alpern and Papadaki. For instance, Basilico et al. provide an algorithmic technique to find the optimal patrolling strategy for large instances of a similar problem on an infinite timescale [3]. The infinite timescale means that the setup differs slightly from that used in [2] and, in addition, Basilico et al. still conclude that finding optimal mixed patrol strategies is computationally difficult and requires the use of reduction strategies. Both [2] and [3] use discrete time steps. However, the timeframes used by the attacker and the patroller need not be aligned. Therefore, in a 2022 paper by Alpern et al. the problem was generalized to include continuous time steps [4]. Subsequently, a 2013 paper by Lin et al. introduced heterogeneous node values for the attacker [5]. An attack could be more impactful in certain places than in others, so adding heterogeneous node values adds realism in that regard. Furthermore, the time required to launch an attack might depend on the location. A 2019 paper by Yolmeh and Baykal-Gürsoy introduced

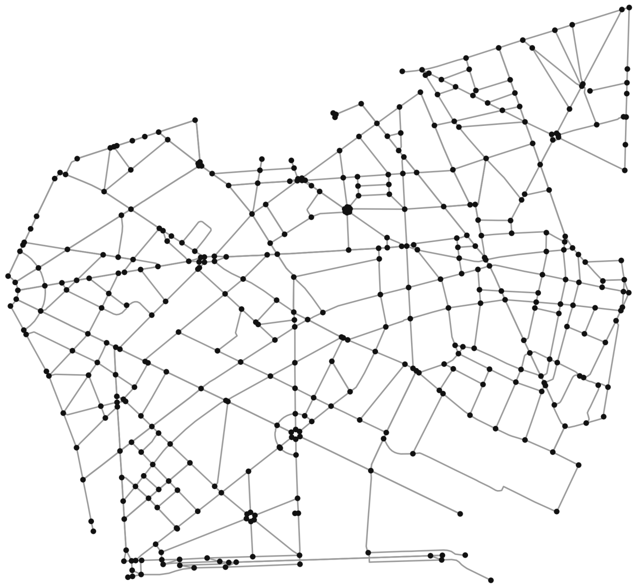


Figure 1.1: Graph representation of the map of the 7th arrondissement of Paris, France.

node-dependent attack times to account for this [6]. A core difficulty of patrolling games that is often mentioned in the literature, including the papers described above, is the size of the solution space for the defender “given that she must consider all potential patrol routes and/or schedules” [1]. Finding the probability of intercepting the attacker generally requires a probability distribution over all possible patrols, which is exactly the bottleneck in this case. As mentioned above, Alpern et al. mitigate this problem by only considering small, very specific instances of the problem [2].

In contrast, this thesis assesses whether the use of a Markovian patrol strategy allows efficient analytical calculation of the interception probability for large, general instances. In Chapter 3, three different methodologies are derived that can be used to calculate the interception probability without having to consider every possible patrol separately. In Chapter 4, it is then shown that the last methodology derived in Section 3.3 is indeed efficient enough to optimize for and compare different patrol strategies for large graph instances in limited time.

2

Model framework

As mentioned in the Introduction, the paper by Alpern et al. lays the foundational structure for patrolling games [2]. Although subsequent papers on the topic have generalized this structure, this thesis seeks to approach the problem from an entirely different mathematical angle. As such, the base structure is deemed the most appropriate for this research. The setup is as follows: A patrolling game is played on a graph $G = (V, E)$ with vertices V and edges E. The structure of this graph is represented using the adjacency matrix $\mathbf{A}$. Each game lasts for d_p time steps, which is the duration of a single patrol. Therefore, a pure strategy for the patroller is a deterministic walk of $d_p - 1$ steps plus a starting point on the graph. The walk is not restricted in any way. In other words, the patrol can start and end at any vertex in the graph and may take any possible path consisting of d_p time steps, which may include staying at the same vertex for consecutive time steps or returning to a vertex that has been visited before. Figure 2.1 shows an example of a possible pure patrol strategy for $d_p = 5$, where the patroller starts at vertex x_1. An attack takes d_a time steps to complete and is carried out in an unknown vertex v_a. If the patrol moves through the vertex under attack while the attack takes place, the attack is intercepted.

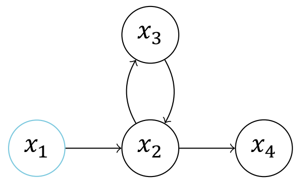


Figure 2.1: Example of a pure patrol strategy for $d_p = 5$.

A mixed strategy for the patroller is a probability distribution over all starting positions $v \in V$ and all possible walks of $d_p - 1$ steps starting in v. Thus, a patrol always visits d_p, not necessarily unique, vertices. A mixed strategy for the attacker is a probability distribution over all combinations of possible attack nodes and attack times. The difficulty lies in the exponential nature of the space of patrolling strategies. Suppose G is complete. Then, at each time step, the patroller has $|V|$ possible steps she can take. So, at worst, there are $|V|^{d_p}$ possible paths the patroller can take.

For a given patrol and a given attack, it is easy to check whether the attack would be intercepted. Therefore, for small instances, it is possible to directly find the interception probability at different vertices. As mentioned above, the attack can be carried out at any vertex $v \in V$. In addition, it is assumed that any attack will start and end within the time span of a single patrol. That means that $d_a \leq d_p$, and for any d_a, the attack may only start at times $t \in \{0, \ldots, d_p - d_a\}$. The attack strategy space can thus be represented as a set or ordered pairs $\mathcal{A}_{d_p,d_a} = \{(v, t) \mid v \in V, t \in \{0, \ldots, d_p - d_a\}\}$ where $\sum_{a \in \mathcal{A}_{d_p,d_a}} \mathbb{P}(a) = 1$. The dependency of the attack strategy space on the attack duration d_a and the patrol duration d_p is clear from context, so the double subscript will subsequently be omitted. The patrol space $\mathcal{B}$ consists of all possible walks on G with a length of d_p, not necessarily unique, vertices. Suppose that A is the random variable that dictates which option the attacker selects, and B is the random variable that determines the specific patrol that is carried out. Then, the value of the game, $\mathcal{U}(A, B)$, is defined as the probability that the attack is intercepted for a given mixed patrol strategy and a given mixed attack strategy. In

Equation (2.1), the dependence of $\mathcal{U}$ on A and B is explicitly denoted. As the specific mixed strategies for the patroller and the attacker can be deduced from the context in which $\mathcal{U}(A, B)$ is used, it is used interchangeably with $\mathcal{U}$.

$$\mathcal{U}(A, B) = \sum_{b \in \mathcal{B}} \sum_{a \in \mathcal{A}} \mathbb{P}_{\mathcal{A}}(A = a)\, \mathbb{P}_{\mathcal{B}}(B = b) \mathbb{1}_{\{\text{patrol } b \text{ intercepts attack } a\}} \tag{2.1}$$

The objective is to find a probability distribution for B that achieves an optimal outcome for the patroller, given a mixed attacker strategy A. In this case, which aspect of the game is optimized depends on the specific situational context of the patroller. For example, she might be willing to risk a large uncertainty in the interception probability if the average value is high enough, or she might prefer a lower interception probability with more certainty. These aspects will be formalized and illustrated in Chapter 4. The most straightforward notion of optimality would be a mixed patroller strategy that maximizes $\mathcal{U}$ for a given mixed attacker strategy. Note that maximizing $\mathcal{U}$ with respect to the mixed patroller strategy A requires an efficient calculation of this value $\mathcal{U}$. However, as mentioned above, since the strategy space of the patroller grows exponentially, this quickly becomes intractable when either the graph or the number of time steps becomes too large. To alleviate this problem, the patrol is modeled as a Markov process in this thesis. This was also one of the research directions proposed by Alpern et al. [2]. Modeling the patrol as a Markov process means that, at every time step, the direction in which to continue the patrol is determined stochastically based on the patroller's current location alone. However, using a Markov chain does not immediately improve the complexity of finding the intercept probability and, in extension, $\mathcal{U}$. The naive method would be to iterate over all possible patrols, but this would not provide any improvement, as the number of patrols increases exponentially in the patrol duration and the graph size.

The patrol is thus modeled as a Markov chain $X = (X_k)_{k \geq 0}$ on the discrete state space V, i.e. the vertices of the graph. So, at each time step t, X_t is a random variable whose realization x_t determines the vertex at which the patroller will be at time step t. The probability of taking each individual step is determined by the time-homogeneous transition matrix $\mathbf{Q}$. Without loss of generality, the vertices $v \in V$ can be numbered $\{1, ..., |V|\}$. Now, a Markov chain also requires a probability distribution for the starting position, which is used in the next chapter. To that end, define:

Definition 2.0.1 (Probability distribution for the starting point of the patrol)**.**

$$\begin{bmatrix} \mathbb{P}(X_0 = 1) \\ \mathbb{P}(X_0 = 2) \\ \vdots \\ \mathbb{P}(X_0 = |V|) \end{bmatrix}^{\mathsf{T}} =: \boldsymbol{\varkappa}$$

Then, let $\mathbf{Y} \in \mathbb{R}^{|V| \times (d_p - d_a + 1)}$ be a matrix containing the probabilities of $\mathcal{A}$. In other words, $y_{\mathrm{v,t}} = \mathbb{P}_{\mathcal{A}}(A = (v, t))$. As mentioned above, a patrol intercepts an attack if the patrol moves through the vertex under attack when the attack takes place. Suppose that an attack on vertex v starts at time step t and ends at time step $t + d_a - 1$. The attack is then intercepted if $(X_t = v) \cup (X_{t+1} = v) \cup ... \cup (X_{t+d_a-1} = v)$. This can be used jointly to express the value $\mathcal{U}$ in terms of the Markov chain X and the attack probability matrix $\mathbf{Y}$:

$$\begin{aligned} \mathcal{U} &= \sum_{b \in \mathcal{B}} \sum_{a \in \mathcal{A}} \mathbb{P}_{\mathcal{A}}(A = a)\, \mathbb{P}_{\mathcal{B}}(B = b) \mathbb{1}_{\{\text{patrol } b \text{ intercepts attack } a\}} \\ &= \sum_{t=0}^{d_p - d_a} \sum_{v \in V} \mathbb{P}(X_t = v \cup ... \cup X_{t+d_a-1} = v)\, \mathbb{P}(Y = (v, t)) \\ &= \sum_{t=0}^{d_p - d_a} \sum_{v=1}^{|V|} \mathbb{P}(X_t = v \cup ... \cup X_{t+d_a-1} = v) y_{\mathrm{v,t}} \end{aligned} \tag{2.2}$$

Nevertheless, this does not yet directly solve the dimensionality problem. Without further simplifications, it is not directly clear how $\mathbb{P}(X_t = i \cup ... \cup X_{t+d_a-1} = i)$ can be determined for all time steps and vertices without iterating over all possible patrols. Thus, to find the value of the game, it would still be necessary to consider every individual patrol. Chapter 3 shows how this can be avoided by utilizing the Markovian structure of the patrol. Specifically, this chapter provides a probabilistic, a recursive, and a direct matrix-multiplicative method to find this probability without having to consider each individual patrol. As briefly mentioned above, Chapter 4 then shows how the characterization in Equation (3.21) can be used to compare different notions of an optimal patrol by applying gradient ascent.

3 Analytical solutions

In the previous chapter, the value $\mathcal{U}$ of the game was derived in Equation (2.2). For simplicity, Sections 3.1 through 3.3 assume that the duration of an attack equals the duration of a complete patrol cycle, i.e., $d_a = d_p$. In Section 3.4, this assumption is relaxed to allow $d_a \neq d_p$. Consequently, under the assumption $d_a = d_p$, any attack must start at $t = 0$, simplifying Equation (2.2) to:

$$\begin{aligned}
\mathcal{U} &= \sum_{t=0}^{d_p-d_a} \sum_{v=1}^{|V|} \mathbb{P}(X_t = v \cup \ldots \cup X_{t+d_a-1} = v) y_{\mathrm{v,t}} \\
&= \sum_{v=1}^{|V|} \mathbb{P}(X_0 = v \cup \ldots \cup X_{d_a-1} = v) y_{\mathrm{v,0}} \\
&= \sum_{v=1}^{|V|} \mathbb{P}(X_0 = v \cup \ldots \cup X_{d_a-1} = v) y_{\mathrm{v}} \\
&= \begin{bmatrix} \mathbb{P}\left(\cup_{t=0}^{d_a-1}(X_t = 1)\right) \\ \mathbb{P}\left(\cup_{t=0}^{d_a-1}(X_t = 2)\right) \\ \vdots \\ \mathbb{P}\left(\cup_{t=0}^{d_a-1}(X_t = |V|)\right) \end{bmatrix}^{\mathsf{T}} \mathbf{Y}
\end{aligned} \tag{3.1}$$

Now, the domain of $\mathbf{Y}$ consists of only the vertices of the graph, rather than a Cartesian product of the vertices and possible starting times. As such, it can be seen as a restriction of the general $\mathbf{Y}$ where the starting time of the attack is fixed at $t = 0$. A result of this is that $\mathbf{Y}$ is now a vector in $\mathbb{R}^{|V|}$ rather than a matrix, so the second index can be omitted. The difficulty currently lies in finding $\mathbb{P}(X_0 = v \cup \ldots \cup X_{d_a-1} = v)$ for all $v \in V$. This value represents the probability that the patroller visits vertex v at least once during its patrol. Section 3.1 provides a direct approach to find this probability using the inclusion-exclusion principle. It is also shown that, by conditioning on the starting position of the patrol, the problem can be reformulated to find a specific matrix $\boldsymbol{\mathcal{P}}_{d_a}$. Then, two propositions are derived and proven in Section 3.2, which jointly allow for a recursive expression of $\boldsymbol{\mathcal{P}}_{d_a}$. Lastly, in Section 3.3, this recursive expression is solved and a closed-form solution for $\boldsymbol{\mathcal{P}}_{d_a}$ is given.

3.1. Inclusion-exclusion

Recall that the vertices $v \in V$ can be numbered $\{1, \ldots, n\}$, where n is the cardinality of V. So, whenever a proof or derivation contains $\sum_{v=1}^{n}$, the summand indices refer to the vertices $v \in V$. In addition, $\odot$ is used to refer to the Hadamard (or element-wise) matrix product. This means that for any square matrix $\mathbf{H}$ and correctly sized identity matrix $\mathbf{I}$, $\mathbf{H} \odot \mathbf{I}$ equals the matrix $\mathbf{H}$ with all nondiagonal elements set to zero. As this operation will be used extensively in the upcoming sections, one last notation is introduced for all square matrices $\mathbf{H}$:

$$\mathbf{H}_D \coloneqq \mathbf{H} \odot \mathbf{I}$$

Now, direct application of the inclusion-exclusion principle to the Markov chain X results in the following:

$$\mathbb{P}\left(\bigcup_{t=0}^{d_p-1}(X_t = v)\right) = \sum_{t=0}^{d_p-1}\left((-1)^{t-1}\sum_{\substack{\mathcal{K}\subseteq\{0,\dots,d_p-1\}\\|\mathcal{K}|=t}}\mathbb{P}\left(\bigcap_{k\in\mathcal{K}}(X_k = v)\right)\right) \tag{3.2}$$

Note that inclusion-exclusion only provides an alternative characterization of $\mathbb{P}(\cup_t(X_t = v))$ for a single vertex v. Therefore, in the following subsections, the inclusion-exclusion operations are vectorized. In other words, it is applied simultaneously, yet separately, to all vertices v:

$$\begin{bmatrix}\mathbb{P}\left(\cup_t(X_t = 1)\right)\\ \mathbb{P}\left(\cup_t(X_t = 2)\right)\\ \vdots \\ \mathbb{P}\left(\cup_t(X_t = n)\right)\end{bmatrix}^\top = \begin{bmatrix}\sum\left((-1)^{t-1}\sum_{\mathcal{K}}\mathbb{P}\left(\cap_{k\in\mathcal{K}}(X_k = 1)\right)\right)\\ \sum\left((-1)^{t-1}\sum_{\mathcal{K}}\mathbb{P}\left(\cap_{k\in\mathcal{K}}(X_k = 2)\right)\right)\\ \vdots \\ \sum\left((-1)^{t-1}\sum_{\mathcal{K}}\mathbb{P}\left(\cap_{k\in\mathcal{K}}(X_k = n)\right)\right)\end{bmatrix}^\top$$

This characterization of (3.1) by itself is not directly helpful, as it still contains an exponentially growing number of probabilities of intersections of events. However, expanding and simplifying the equation for a few values of d_p, provides additional insight. The base case is when a patrol consists of a single time step, i.e. $d_p = 1$. In this scenario, only the starting point of the patrol is visited. It follows directly from Definition 2.0.1 that the probability distribution is then represented by $\boldsymbol{\varkappa}$. Sections 3.1.1 through 3.1.3 provide more elaborate derivations for $d_p = 2$, $d_p = 3$ and $d_p = 4$.

3.1.1. $d_p = 2$

Now, the patroller starts in a certain vertex and then takes a single step. Expanding Equation (3.2) gives:

$$\begin{aligned}\begin{bmatrix}\mathbb{P}(\cup_{t=0}^1(X_t = 1))\\ \mathbb{P}(\cup_{t=0}^1(X_t = 2))\\ \vdots \\ \mathbb{P}(\cup_{t=0}^1(X_t = n))\end{bmatrix}^\top &= \begin{bmatrix}\mathbb{P}(X_0 = 1)\\ \mathbb{P}(X_0 = 2)\\ \vdots \\ \mathbb{P}(X_0 = n)\end{bmatrix}^\top + \begin{bmatrix}\mathbb{P}(X_1 = 1)\\ \mathbb{P}(X_1 = 2)\\ \vdots \\ \mathbb{P}(X_1 = n)\end{bmatrix}^\top - \begin{bmatrix}\mathbb{P}\left(\cap_{t=0}^1(X_t = 1)\right)\\ \mathbb{P}\left(\cap_{t=0}^1(X_t = 2)\right)\\ \vdots \\ \mathbb{P}\left(\cap_{t=0}^1(X_t = n)\right)\end{bmatrix}^\top \\ &= \boldsymbol{\varkappa} + \begin{bmatrix}\mathbb{P}(X_1 = 1)\\ \mathbb{P}(X_1 = 2)\\ \vdots \\ \mathbb{P}(X_1 = n)\end{bmatrix}^\top - \begin{bmatrix}\mathbb{P}\left(\cap_{t=0}^1(X_t = 1)\right)\\ \mathbb{P}\left(\cap_{t=0}^1(X_t = 2)\right)\\ \vdots \\ \mathbb{P}\left(\cap_{t=0}^1(X_t = n)\right)\end{bmatrix}^\top\end{aligned}$$

Explicit calculation of this vector of probabilities requires careful consideration of the newly introduced terms. Since X represents a Markov random process, the probability of each step is dictated by the transition matrix **Q**. Therefore, this equation can be rewritten as follows:

$$\begin{aligned}\begin{bmatrix}\mathbb{P}(X_1 = 1)\\ \mathbb{P}(X_1 = 2)\\ \vdots \\ \mathbb{P}(X_1 = n)\end{bmatrix}^\top &= \sum_{v=1}^{n}\begin{bmatrix}\mathbb{P}(X_1 = 1 \mid X_0 = v)\,\mathbb{P}(X_0 = v)\\ \mathbb{P}(X_1 = 2 \mid X_0 = v)\,\mathbb{P}(X_0 = v)\\ \vdots \\ \mathbb{P}(X_1 = n \mid X_0 = v)\,\mathbb{P}(X_0 = v)\end{bmatrix}^\top \\ &= \sum_{v=1}^{n}\begin{bmatrix}q_{\mathrm{v},1}\varkappa_{\mathrm{v}}\\ q_{\mathrm{v},2}\varkappa_{\mathrm{v}}\\ \vdots \\ q_{\mathrm{v},\mathrm{n}}\varkappa_{\mathrm{v}}\end{bmatrix}^\top \\ &= \boldsymbol{\varkappa}\mathbf{Q}\end{aligned} \tag{3.3}$$

Lastly, consider the vector $[\mathbb{P}(X_0 = 1 \cap X_1 = 1), \ldots, \mathbb{P}(X_0 = n \cap X_1 = n)]$. It immediately becomes clear that each entry within this vector is the probability of starting in the corresponding Markov state and staying there. More formally:

$$\begin{aligned}
\begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_1 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_1 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_1 = n)) \end{bmatrix}^\top &= \begin{bmatrix} \mathbb{P}(X_1 = 1 \mid X_0 = 1)\,\mathbb{P}(X_0 = 1) \\ \mathbb{P}(X_1 = 2 \mid X_0 = 2)\,\mathbb{P}(X_0 = 2) \\ \vdots \\ \mathbb{P}(X_1 = n \mid X_0 = n)\,\mathbb{P}(X_0 = n) \end{bmatrix}^\top \\
&= \begin{bmatrix} q_{1,1}\varkappa_1 \\ q_{2,2}\varkappa_2 \\ \vdots \\ q_{\mathrm{n},\mathrm{n}}\varkappa_{\mathrm{n}} \end{bmatrix}^\top \\
&= \varkappa\,\mathbf{Q}_D
\end{aligned} \tag{3.4}$$

Taken together, this results in the following.

$$\begin{bmatrix} \mathbb{P}\left(\bigcup_{t=0}^{1}(X_t = 1)\right) \\ \mathbb{P}\left(\bigcup_{t=0}^{1}(X_t = 2)\right) \\ \vdots \\ \mathbb{P}\left(\bigcup_{t=0}^{1}(X_t = n)\right) \end{bmatrix}^\top = \varkappa(\mathbf{I} + \mathbf{Q} - \mathbf{Q}_D) \tag{3.5}$$

3.1.2. $d_p = 3$

Adding a third patrol step results in a number of additional terms, as can be seen in (3.6).

$$\begin{aligned}
\begin{bmatrix} \mathbb{P}(\bigcup_{t=0}^{2}(X_t = 1)) \\ \mathbb{P}(\bigcup_{t=0}^{2}(X_t = 2)) \\ \vdots \\ \mathbb{P}(\bigcup_{t=0}^{2}(X_t = n)) \end{bmatrix}^\top &= \begin{bmatrix} \mathbb{P}(X_0 = 1) \\ \mathbb{P}(X_0 = 2) \\ \vdots \\ \mathbb{P}(X_0 = n) \end{bmatrix}^\top + \begin{bmatrix} \mathbb{P}(X_1 = 1) \\ \mathbb{P}(X_1 = 2) \\ \vdots \\ \mathbb{P}(X_1 = n) \end{bmatrix}^\top + \begin{bmatrix} \mathbb{P}(X_2 = 1) \\ \mathbb{P}(X_2 = 2) \\ \vdots \\ \mathbb{P}(X_2 = n) \end{bmatrix}^\top - \begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_1 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_1 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_1 = n)) \end{bmatrix}^\top \\
&\quad - \begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_2 = n)) \end{bmatrix}^\top - \begin{bmatrix} \mathbb{P}((X_1 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_1 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_1 = n) \cap (X_2 = n)) \end{bmatrix}^\top \\
&\quad + \begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_1 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_1 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_1 = n) \cap (X_2 = n)) \end{bmatrix}^\top \\
&= \varkappa + \varkappa\mathbf{Q} + \varkappa\mathbf{Q}^2 - \varkappa\mathbf{Q}_D - \begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_2 = n)) \end{bmatrix}^\top \\
&\quad - \begin{bmatrix} \mathbb{P}((X_1 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_1 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_1 = n) \cap (X_2 = n)) \end{bmatrix}^\top + \begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_1 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_1 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_1 = n) \cap (X_2 = n)) \end{bmatrix}^\top
\end{aligned} \tag{3.6}$$

Certain terms can be found relatively easily. For instance, the reasoning used for equation (3.3), also holds for $t = 2$ in equation (3.7). The only difference is that, rather than using the one-step transition matrix $\mathbf{Q}$, the two-step transition matrix $\mathbf{Q}^2$ with entries $q^2_{\mathrm{i},\mathrm{j}}$ is used.

$$\begin{bmatrix} \mathbb{P}(X_2 = 1) & \mathbb{P}(X_2 = 2) & \ldots & \mathbb{P}(X_2 = n) \end{bmatrix} = \varkappa\,\mathbf{Q}^2 \tag{3.7}$$

Similarly,

$$
\begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_2 = n)) \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \mathbb{P}(X_2 = 1 \mid X_0 = 1)\, \mathbb{P}(X_0 = 1) \\ \mathbb{P}(X_2 = 2 \mid X_0 = 2)\, \mathbb{P}(X_0 = 2) \\ \vdots \\ \mathbb{P}(X_2 = n \mid X_0 = n)\, \mathbb{P}(X_0 = n) \end{bmatrix}^{\mathsf{T}}
= \begin{bmatrix} q_{1,1}^2 \varkappa_1 \\ q_{2,2}^2 \varkappa_2 \\ \vdots \\ q_{\mathrm{n,n}}^2 \varkappa_{\mathrm{n}} \end{bmatrix}^{\mathsf{T}}
= \boldsymbol{\varkappa}(\mathbf{Q}^2)_D \tag{3.8}
$$

And also

$$
\begin{bmatrix} \mathbb{P}((X_1 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_1 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_1 = n) \cap (X_2 = n)) \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \mathbb{P}(X_2 = 1 \mid X_1 = 1)\, \mathbb{P}(X_1 = 1) \\ \mathbb{P}(X_2 = 2 \mid X_1 = 2)\, \mathbb{P}(X_1 = 2) \\ \vdots \\ \mathbb{P}(X_2 = n \mid X_1 = n)\, \mathbb{P}(X_1 = n) \end{bmatrix}^{\mathsf{T}}
$$

Now, denote the probability vector found in Equation (3.3) by $\mathbf{w} \coloneqq \boldsymbol{\varkappa}\mathbf{Q}$. So, $w_{\mathrm{i}} \coloneqq \mathbb{P}(X_1 = i)$ and therefore:

$$
= \begin{bmatrix} q_{1,1} w_1 \\ q_{2,2} w_2 \\ \vdots \\ q_{\mathrm{n,n}} w_{\mathrm{n}} \end{bmatrix}^{\mathsf{T}}
= \boldsymbol{\varkappa}\, \mathbf{Q}\, \mathbf{Q}_D \tag{3.9}
$$

The last element to be determined contains an additional term:

$$
\begin{bmatrix} \mathbb{P}((X_0 = 1) \cap (X_1 = 1) \cap (X_2 = 1)) \\ \mathbb{P}((X_0 = 2) \cap (X_1 = 2) \cap (X_2 = 2)) \\ \vdots \\ \mathbb{P}((X_0 = n) \cap (X_1 = n) \cap (X_2 = n)) \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \mathbb{P}(X_2 = 1 \mid (X_1 = 1) \cap (X_0 = 1))\, \mathbb{P}(X_1 = 1 \mid X_0 = 1)\, \mathbb{P}(X_0 = 1) \\ \mathbb{P}(X_2 = 2 \mid (X_1 = 2) \cap (X_0 = 2))\, \mathbb{P}(X_1 = 2 \mid X_0 = 2)\, \mathbb{P}(X_0 = 2) \\ \vdots \\ \mathbb{P}(X_2 = n \mid (X_1 = n) \cap (X_0 = n))\, \mathbb{P}(X_1 = n \mid X_0 = n)\, \mathbb{P}(X_0 = n) \end{bmatrix}^{\mathsf{T}}
$$

$$
= \begin{bmatrix} \mathbb{P}(X_2 = 1 \mid (X_1 = 1) \cap (X_0 = 1)) q_{1,1} \varkappa_1 \\ \mathbb{P}(X_2 = 2 \mid (X_1 = 2) \cap (X_0 = 2)) q_{2,2} \varkappa_2 \\ \vdots \\ \mathbb{P}(X_2 = n \mid (X_1 = n) \cap (X_0 = n)) q_{\mathrm{n,n}} \varkappa_{\mathrm{n}} \end{bmatrix}^{\mathsf{T}}
= \begin{bmatrix} \mathbb{P}(X_2 = 1 \mid X_1 = 1) q_{1,1} \varkappa_1 \\ \mathbb{P}(X_2 = 2 \mid X_1 = 2) q_{2,2} \varkappa_2 \\ \vdots \\ \mathbb{P}(X_2 = n \mid X_1 = n) q_{\mathrm{n,n}} \varkappa_{\mathrm{n}} \end{bmatrix}^{\mathsf{T}}
$$

Now, since this Markov process is time-invariant, it holds that $\mathbb{P}(X_2 = i \mid X_1 = i) = \mathbb{P}(X_1 = i \mid X_0 = i)$. Therefore:

$$
= \begin{bmatrix} (q_{1,1})^2 \varkappa_1 \\ (q_{2,2})^2 \varkappa_2 \\ \vdots \\ (q_{\mathrm{n,n}})^2 \varkappa_{\mathrm{n}} \end{bmatrix}^{\mathsf{T}}
= \boldsymbol{\varkappa}(\mathbf{Q}_D)^2
$$

Combining these results gives:

$$\begin{bmatrix} \mathbb{P}(\cup_{t=0}^{2}(X_t = 1)) \\ \mathbb{P}(\cup_{t=0}^{2}(X_t = 2)) \\ \vdots \\ \mathbb{P}(\cup_{t=0}^{2}(X_t = n)) \end{bmatrix}^{\top} = \varkappa\left(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 - \mathbf{Q}_D - (\mathbf{Q}^2)_D - \mathbf{Q}\,\mathbf{Q}_D + (\mathbf{Q}_D)^2\right) \tag{3.10}$$

3.1.3. $d_p = 4$

When adding a fourth time step, the methodologies applied in the previous subsections can be used again. As the derivations remain virtually unchanged, a proof of equation (3.11) is omitted.

$$\begin{aligned} \begin{bmatrix} \mathbb{P}(\cup_{t=0}^{3}(X_t = 1)) \\ \mathbb{P}(\cup_{t=0}^{3}(X_t = 2)) \\ \vdots \\ \mathbb{P}(\cup_{t=0}^{3}(X_t = n)) \end{bmatrix}^{\top} &= \varkappa\Big(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3 - \mathbf{Q}_D - (\mathbf{Q}^2)_D - (\mathbf{Q}^3)_D - \mathbf{Q}\,\mathbf{Q}_D - \mathbf{Q}(\mathbf{Q}^2)_D \\ &\quad - \mathbf{Q}^2\,\mathbf{Q}_D + (\mathbf{Q}_D)^2 + (\mathbf{Q}^2)_D\,\mathbf{Q}_D + \mathbf{Q}_D(\mathbf{Q}^2)_D + \mathbf{Q}(\mathbf{Q}_D)^2 - (\mathbf{Q}_D)^3\Big) \\ &= \varkappa\Big(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3 - \mathbf{Q}_D - (\mathbf{Q}^2)_D - (\mathbf{Q}^3)_D - \mathbf{Q}\,\mathbf{Q}_D - \mathbf{Q}(\mathbf{Q}^2)_D \\ &\quad - \mathbf{Q}^2\,\mathbf{Q}_D + (\mathbf{Q}_D)^2 + 2(\mathbf{Q}^2)_D\,\mathbf{Q}_D + \mathbf{Q}(\mathbf{Q}_D)^2 - (\mathbf{Q}_D)^3\Big) \end{aligned} \tag{3.11}$$

Some additional simplifications can be made. Since the Hadamard product is distributive, it is possible to define $\mathbf{Q}_O \coloneqq \mathbf{Q} - \mathbf{Q}_D$. Since $\mathbf{Q}_D$ contains the probabilities that the Markov process stays in the same state for one time step, $\mathbf{Q}_O$ only contains the probabilities that the Markov process leaves its current state after a single time step. In addition, using the distributive nature of the Hadamard product and the matrix product separately, the following terms can be rewritten as follows:

$$\begin{aligned} \mathbf{Q}\,\mathbf{Q}_D &= (\mathbf{Q}_D + \mathbf{Q}_O)\,\mathbf{Q}_D = (\mathbf{Q}_D)^2 + \mathbf{Q}_O\,\mathbf{Q}_D \\ \mathbf{Q}(\mathbf{Q}^2)_D &= (\mathbf{Q}_D + \mathbf{Q}_O)(\mathbf{Q}^2)_D = \mathbf{Q}_D(\mathbf{Q}^2)_D + \mathbf{Q}_O(\mathbf{Q}^2)_D \\ \mathbf{Q}^2\,\mathbf{Q}_D &= (\mathbf{Q}_D + \mathbf{Q}_O)(\mathbf{Q}_D + \mathbf{Q}_O)\,\mathbf{Q}_D \\ &= (\mathbf{Q}_D)^3 + (\mathbf{Q}_D)(\mathbf{Q}_O)(\mathbf{Q}_D) + (\mathbf{Q}_O)(\mathbf{Q}_D)^2 + (\mathbf{Q}_O)^2(\mathbf{Q}_D) \\ \mathbf{Q}(\mathbf{Q}_D)^2 &= (\mathbf{Q}_D + \mathbf{Q}_O)(\mathbf{Q}_D)^2 = (\mathbf{Q}_D)^3 + \mathbf{Q}_O(\mathbf{Q}_D)^2 \end{aligned}$$

Combined, this yields the following.

$$\begin{aligned} \begin{bmatrix} \mathbb{P}(\cup_{t=0}^{3}(X_t = 1)) \\ \mathbb{P}(\cup_{t=0}^{3}(X_t = 2)) \\ \vdots \\ \mathbb{P}(\cup_{t=0}^{3}(X_t = n)) \end{bmatrix}^{\top} &= \varkappa\Big(\mathbf{I} + (\mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3)_O - (\mathbf{Q}_D)^2 - \mathbf{Q}_O\,\mathbf{Q}_D - \mathbf{Q}_D(\mathbf{Q}^2)_D - \mathbf{Q}_O(\mathbf{Q}^2)_D \\ &\quad - (\mathbf{Q}_D)^3 - \mathbf{Q}_D\,\mathbf{Q}_O\,\mathbf{Q}_D - \mathbf{Q}_O(\mathbf{Q}_D)^2 - (\mathbf{Q}_O)^2\,\mathbf{Q}_D + (\mathbf{Q}_D)^2 \\ &\quad + 2\,\mathbf{Q}_D(\mathbf{Q}^2)_D + (\mathbf{Q}_D)^3 + \mathbf{Q}_O(\mathbf{Q}_D)^2 - (\mathbf{Q}_D)^3\Big) \end{aligned}$$

$$= \varkappa\Big(\mathbf{I} + (\mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3)_O - \mathbf{Q}_O\,\mathbf{Q}_D \\ -(\mathbf{Q}_D)^3 - \mathbf{Q}_D\,\mathbf{Q}_O\,\mathbf{Q}_D - \mathbf{Q}_O(\mathbf{Q}^2)_D - (\mathbf{Q}_O)^2\,\mathbf{Q}_D \\ + \mathbf{Q}_D(\mathbf{Q}^2)_D\Big) \tag{3.12}$$

This formulation still contains multiple $(\mathbf{Q}^2)_D$. However, this can be rewritten even further:

$$\begin{aligned}(\mathbf{Q}^2)_D &= ((\mathbf{Q}_D + \mathbf{Q}_O)^2)_D \\ &= ((\mathbf{Q}_D)^2)_D + (\mathbf{Q}_O\,\mathbf{Q}_D)_D + (\mathbf{Q}_D\,\mathbf{Q}_O)_D + ((\mathbf{Q}_O)^2)_D\end{aligned}$$

Furthermore, $(\mathbf{Q}_D)^n$ is a matrix power of a diagonal matrix, which itself is diagonal. Therefore, $((\mathbf{Q}_D)^2)_D = (\mathbf{Q}_D)^2$. In addition, since $\mathbf{Q}_D$ is diagonal and $\mathbf{Q}_O$ only has zeros on the diagonal, $\mathbf{Q}_O\,\mathbf{Q}_D$ and $\mathbf{Q}_D\,\mathbf{Q}_O$ only contain zeros on the diagonal. Therefore, $(\mathbf{Q}_O\,\mathbf{Q}_D)_D = (\mathbf{Q}_D\,\mathbf{Q}_O)_D = \mathbf{O}$. Combining this gives the following.

$$(\mathbf{Q}^2)_D = (\mathbf{Q}_D)^2 + ((\mathbf{Q}_O)^2)_D$$

And consequently:

$$\begin{aligned}\begin{bmatrix}\mathbb{P}(\cup_{t=0}^3(X_t = 1)) \\ \mathbb{P}(\cup_{t=0}^3(X_t = 2)) \\ \vdots \\ \mathbb{P}(\cup_{t=0}^3(X_t = n))\end{bmatrix}^\mathsf{T} &= \varkappa\Big(\mathbf{I} + (\mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3)_O - \mathbf{Q}_O\,\mathbf{Q}_D - (\mathbf{Q}_D)^3 - \mathbf{Q}_D\,\mathbf{Q}_O\,\mathbf{Q}_D \\ &\quad - \mathbf{Q}_O((\mathbf{Q}_D)^2 + ((\mathbf{Q}_O)^2)_D) - (\mathbf{Q}_O)^2\,\mathbf{Q}_D \\ &\quad + \mathbf{Q}_D((\mathbf{Q}_D)^2 + ((\mathbf{Q}_O)^2)_D)\Big) \\ &= \varkappa\Big(\mathbf{I} + (\mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3)_O - \mathbf{Q}_O\,\mathbf{Q}_D - \mathbf{Q}_D\,\mathbf{Q}_O\,\mathbf{Q}_D \\ &\quad - \mathbf{Q}_O(\mathbf{Q}_D)^2 - \mathbf{Q}_O((\mathbf{Q}_O)^2)_D - (\mathbf{Q}_O)^2\,\mathbf{Q}_D + \mathbf{Q}_D((\mathbf{Q}_O)^2)_D\Big) \\ &= \varkappa\Big(\mathbf{I} + (\mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3)_O - (\mathbf{Q}_D + \mathbf{Q}_O + \mathbf{I})\,\mathbf{Q}_O\,\mathbf{Q}_D - \mathbf{Q}_O(\mathbf{Q}_D)^2 \\ &\quad - \mathbf{Q}_O((\mathbf{Q}_O)^2)_D + \mathbf{Q}_D((\mathbf{Q}_O)^2)_D\Big) \\ &= \varkappa\Big(\mathbf{I} + (\mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3)_O - (\mathbf{Q} + \mathbf{I})\,\mathbf{Q}_O\,\mathbf{Q}_D \\ &\quad - \mathbf{Q}_O(\mathbf{Q}_D)^2 - \mathbf{Q}_O((\mathbf{Q}_O)^2)_D + \mathbf{Q}_D((\mathbf{Q}_O)^2)_D\Big)\end{aligned} \tag{3.13}$$

Furthermore,

$$\mathbf{Q}_D((\mathbf{Q}_O)^2)_D \overset{(3.1.1)}{=} (\mathbf{Q}_D(\mathbf{Q}_O)^2)_D = ((\mathbf{Q}_O)^2\,\mathbf{Q}_D)_D$$

The first equality makes use of Theorem 3.1.1 below, retrieved from [7]. The second equality follows from the fact that for a square matrix $\mathbf{A}_1$ and a diagonal matrix $\mathbf{A}_2$, $\mathbf{A}_1\mathbf{A}_2$ equals $\mathbf{A}_2\mathbf{A}_1$ on the diagonal.

Theorem 3.1.1. *Suppose* $\mathbf{A}, \mathbf{B}$ *are* $m \times n$ *matrices and* $\mathbf{D}$ *and* $\mathbf{E}$ *are diagonal matrices of sizes* $m \times m$ *and* $n \times n$*, respectively. Then,*

$$\begin{aligned}\mathbf{D}(\mathbf{A} \odot \mathbf{B})\mathbf{E} &= (\mathbf{DAE}) \odot \mathbf{B} = (\mathbf{DA}) \odot (\mathbf{BE}) \\ &= (\mathbf{AE}) \odot (\mathbf{DB}) = \mathbf{A} \odot (\mathbf{DBE})\end{aligned}$$

In addition, it holds that:

$$\begin{aligned}((\mathbf{Q}+\mathbf{I})(\mathbf{Q}_O\,\mathbf{Q}_D))_D &= (\mathbf{Q}\,\mathbf{Q}_O\,\mathbf{Q}_D)_D + (\mathbf{Q}_O\,\mathbf{Q}_D)_D \\ &= (\mathbf{Q}\,\mathbf{Q}_O\,\mathbf{Q}_D)_D \\ &= (\mathbf{Q}_O\,\mathbf{Q}_O\,\mathbf{Q}_D)_D + (\mathbf{Q}_D\,\mathbf{Q}_O\,\mathbf{Q}_D)_D \\ &= ((\mathbf{Q}_O)^2\,\mathbf{Q}_D)_D\end{aligned} \tag{3.14}$$

Therefore:

$$\mathbf{Q}_D((\mathbf{Q}_O)^2)_D - (\mathbf{Q}+\mathbf{I})\,\mathbf{Q}_O\,\mathbf{Q}_D = -((\mathbf{Q}+\mathbf{I})\,\mathbf{Q}_O\,\mathbf{Q}_D)_O \tag{3.15}$$

And consequently, equation (3.11) becomes:

$$\begin{aligned}\begin{bmatrix}\mathbb{P}(\bigcup_{t=0}^{3}(X_t = 1)) \\ \mathbb{P}(\bigcup_{t=0}^{3}(X_t = 2)) \\ \vdots \\ \mathbb{P}(\bigcup_{t=0}^{3}(X_t = n))\end{bmatrix}^{\mathsf{T}} &= \varkappa\left(\mathbf{I}+(\mathbf{Q}+\mathbf{Q}^2+\mathbf{Q}^3)_O - ((\mathbf{Q}+\mathbf{I})\,\mathbf{Q}_O\,\mathbf{Q}_D)_O - \mathbf{Q}_O(\mathbf{Q}_D)^2 - \mathbf{Q}_O((\mathbf{Q}_O)^2)_D\right) \\ &= \varkappa\left(\mathbf{I}+(\mathbf{Q}+\mathbf{Q}^2+\mathbf{Q}^3)_O - ((\mathbf{Q}+\mathbf{I})\,\mathbf{Q}_O\,\mathbf{Q}_D)_O - \mathbf{Q}_O(\mathbf{Q}^2)_D\right)\end{aligned} \tag{3.16}$$

These steps have shown that it is possible not only to obtain a closed-form solution for the desired probability but also to simplify it significantly. However, this simplification required a tedious amount of manual algebra and has not yet resulted in a general closed-form solution for all d_p and d_a. Nevertheless, there does appear to be a structure present. In particular, in every solution up to and including $d_p = 4$, $\varkappa$ can be factored out. This can also be directly deduced from equation (3.1). Indeed, by applying the law of total probability, part of this equation can be rewritten as follows:

$$\begin{aligned}\begin{bmatrix}\mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 1)\right) \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 2)\right) \\ \vdots \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = n)\right)\end{bmatrix}^{\mathsf{T}} &= \sum_{i=1}^{n}\begin{bmatrix}\mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 1) \,|\, X_0 = i\right)\mathbb{P}(X_0 = i) \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 2) \,|\, X_0 = i\right)\mathbb{P}(X_0 = i) \\ \vdots \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = n) \,|\, X_0 = i\right)\mathbb{P}(X_0 = i)\end{bmatrix}^{\mathsf{T}} \\ &= \sum_{i=1}^{n}\begin{bmatrix}\mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 1) \,|\, X_0 = i\right)\varkappa_i \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 2) \,|\, X_0 = i\right)\varkappa_i \\ \vdots \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = n) \,|\, X_0 = i\right)\varkappa_i\end{bmatrix}^{\mathsf{T}} \\ &= \begin{bmatrix}\mathbb{P}(X_0 = 1) \\ \mathbb{P}(X_0 = 2) \\ \vdots \\ \mathbb{P}(X_0 = n)\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 1) \,|\, X_0 = 1\right) \cdots\cdots \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = n) \,|\, X_0 = 1\right) \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 1) \,|\, X_0 = 2\right) \cdots\cdots \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = n) \,|\, X_0 = 2\right) \\ \vdots \\ \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = 1) \,|\, X_0 = n\right) \cdots\cdots \mathbb{P}\left(\bigcup_{t=0}^{d_a-1}(X_t = n) \,|\, X_0 = n\right)\end{bmatrix} \\ &= \varkappa\boldsymbol{\mathcal{P}}_{d_a}\end{aligned} \tag{3.17}$$

As such, to find a simple way to calculate the value $\mathcal{U}$ of the game, it is necessary to find a closed-form expression of the matrix $\boldsymbol{\mathcal{P}}_{d_a}$ as defined here. Upon closer inspection, it becomes clear that $\boldsymbol{\mathcal{P}}_{d_a}$ is a matrix containing the probabilities of visiting a certain state in the Markov chain given a starting position. No research could be identified where a direct expression of this matrix of probabilities is provided. Therefore, in sections 3.2 and 3.3, two methods are derived that allow direct calculation of this matrix for every $d_a = d_p \geq 1$.

3.2. Recursive matrix-equation

Let $X = (X_k)_{k\geq 0}$ be a Markov chain on the discrete state space Ω with a time-homogeneous transition matrix $\mathbf{Q}$. Furthermore, define:

Definition 3.2.1 (First passage time). T_ω is the time step where state $\omega \in \Omega$ is visited for the first time (first passage time step).

Definition 3.2.2 (Number of visits). $N_\omega^{(m)}$ is the number of visits of state $\omega \in \Omega$ in $m+1$ time steps, that is, the number of visits after taking m steps plus the starting point.

Definition 3.2.3 (Identity matrix). $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, where n is the cardinality of Ω.

Definition 3.2.4 (First passage probabilities). $\mathbf{P}_k$ is the matrix that contains the first visit probabilities after taking k steps given a starting point, so $(p_k)_{i,j} \coloneqq \mathbb{P}(T_j = k \,|\, X_0 = i)$.

Lastly, note that $\mathbb{P}(X_0 = j \cup \ldots \cup X_m = j \,|\, X_0 = i) = \mathbb{P}(N_j^{(m)} \geq 1 \,|\, X_0 = i)$.

Proposition 3.2.1. $\mathbb{P}(N_j^{(m)} \geq 1 \,|\, X_0 = i) = \sum_{k=0}^{m} (p_k)_{i,j}$

Proof.

$$\mathbb{P}(N_j^{(m)} \geq 1 \,|\, X_0 = i) = \sum_{k=0}^{m} \mathbb{P}(N_j^{(m)} \geq 1, T_j = k \,|\, X_0 = i)$$

This follows from a direct application of the law of total probability using the partition $\{T_j = k : 0 \leq k \leq m\} \cup \{T_j > k\}$. Note that $\mathbb{P}(N_j^{(m)} \geq 1, T_j > m \,|\, X_0 = i) = 0$, so the expression above follows directly. Then, applying the definition of conditional probability gives:

$$= \sum_{k=0}^{m} \mathbb{P}(T_j = k \,|\, X_0 = i)\, \mathbb{P}(N_j^{(m)} \geq 1 \,|\, T_j = k, X_0 = i)$$

However, if $T_j = k$ for $0 \leq k \leq m$, then the Markov process certainly visits state j at least once in m steps, namely at time k. So, $\mathbb{P}(N_j^{(m)} \geq 1 \,|\, T_j = k, X_0 = i) = 1$ and as such:

$$= \sum_{k=0}^{m} \mathbb{P}(T_j = k \,|\, X_0 = i)$$

$$= \sum_{k=0}^{m} (p_k)_{i,j}$$

□

Proposition 3.2.2. $\sum_{k=0}^{m} \mathbf{P}_k (\mathbf{Q}^{m-k})_D = \mathbf{Q}^m$

Proof.

The m step transition matrix $\mathbf{Q}^m$ is used. This means that:

$$q_{\mathrm{i,j}}^m = \mathbb{P}(X_m = j \,|\, X_0 = i)$$

Here, the law of total probability can be applied using the same partition over T_j as in the proof of proposition 3.2.1, after which the definition of conditional probability is used again.

$$= \sum_{k=0}^{m} \mathbb{P}(X_m = j,\, T_j = k \,|\, X_0 = i)$$

$$= \sum_{k=0}^{m} \mathbb{P}(X_m = j \,|\, T_j = k,\, X_0 = i)\, \mathbb{P}(T_j = k \,|\, X_0 = i)$$

$$\overset{(3.2.4)}{=} \sum_{k=0}^{m} \mathbb{P}(X_m = j \,|\, T_j = k,\, X_0 = i)(p_k)_{i,j}$$

Now, note that $\mathbb{P}(X_m = j \,|\, T_j = k,\, X_0 = i) = \mathbb{P}(X_m = j \,|\, (X_0 \neq j, \ldots, X_{k-1} \neq j, X_k = j),\, X_0 = i) = \mathbb{P}(X_m = j \,|\, X_k = j)$. The second equality follows directly from the Markov property. Therefore,

$$= \sum_{k=0}^{m} \mathbb{P}(X_m = j \,|\, X_k = j)(p_k)_{i,j}$$

$$= \sum_{k=0}^{m} q_{\mathrm{j,j}}^{m-k}(p_k)_{i,j}$$

Note that this holds for all $1 \leq i, j \leq n$. Combined with the fact that only the diagonal elements of $\mathbf{Q}^{m-k}$ are used, his can be written as a matrix-vector product and, therefore, it directly follows that:

$$\mathbf{Q}^m = \sum_{k=0}^{m} \mathbf{P}_k(\mathbf{Q}^{m-k})_D$$

□

Proposition 3.2.1 shows that the probability of visiting a state at least once can be written as the sum of the first passage probabilities and Proposition 3.2.2 provides a recursive way to find the first passage probabilities using the transition matrix $\mathbf{Q}$. If $m = 0$, then:

$$\mathbf{Q}^0 = \mathbf{P}_0\, \mathbf{Q}_D^0 \implies \mathbf{P}_0 = \mathbf{I}$$

Therefore, assuming $m \geq 1$:

$$\mathbf{Q}^m = \sum_{k=0}^{m} \mathbf{P}_k(\mathbf{Q}^{m-k})_D \iff \mathbf{Q}^m = \mathbf{P}_m + \sum_{k=0}^{m-1} \mathbf{P}_k(\mathbf{Q}^{m-k})_D$$

$$\implies \mathbf{P}_m = \mathbf{Q}^m - \sum_{k=0}^{m-1} \mathbf{P}_k(\mathbf{Q}^{m-k})_D$$

$$\implies \sum_{j=0}^{m} \mathbf{P}_j = \sum_{j=1}^{m} \left(\mathbf{Q}^j - \sum_{k=0}^{j-1} \mathbf{P}_k(\mathbf{Q}^{j-k})_D \right) + \mathbf{I}$$

$$\iff \boldsymbol{\mathcal{P}} = \sum_{j=1}^{m} \left(\mathbf{Q}^j - \sum_{k=0}^{j-1} \mathbf{P}_k(\mathbf{Q}^{j-k})_D \right) + \mathbf{I} \tag{3.18}$$

This method is an improvement compared to the inclusion-exclusion method as set out in the previous section. Equation (3.18) provides a direct and recursive method of calculating $\mathcal{P}_{d_a}$ for any $d_a \geq 1$. However, the recursive element is also what makes this method computationally expensive. The different $\mathbf{P}_i$ all need to be stored and reused at every subsequent time step. To account for this, Proposition 3.2.2 is expanded and solved explicitly in the next section.

3.3. Closed-form block matrix solution

Using block matrices, it is possible to rewrite equation 3.2.2 as a matrix-matrix equation. Note that in the proof, $m \geq 0$ is arbitrary. Therefore, the equation also holds for $k = 0, 1, \ldots, m-1$. Now, after substituting d_p for m and setting $d_p = 0$, it is possible to rewrite:

$$\begin{aligned} \mathbf{I} &= \sum_{k=0}^{0} \mathbf{P}_k(\mathbf{Q}^{-k} \odot \mathbf{I}) \\ &= \mathbf{I} + \sum_{k=1}^{m} \mathbf{O} \cdot \mathbf{P}_k \end{aligned} \tag{3.19}$$

Or, equivalently,

$$\begin{bmatrix} \mathbf{P}_0 & \mathbf{P}_1 & \ldots & \mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{O} \\ \vdots \\ \mathbf{O} \end{bmatrix} = \mathbf{I}$$

The same process can be applied for $d_p = 1$:

$$\begin{aligned} \mathbf{Q} &= \sum_{k=0}^{1} \mathbf{P}_k(\mathbf{Q}^{1-k} \odot \mathbf{I}) \\ &= \mathbf{P}_0(\mathbf{Q} \odot \mathbf{I}) + \mathbf{P}_1 + \sum_{k=2}^{m} \mathbf{O} \cdot \mathbf{P}_k \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{P}_0 & \mathbf{P}_1 & \ldots & \mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{Q}_D \\ \mathbf{I} \\ \mathbf{O} \\ \vdots \\ \mathbf{O} \end{bmatrix} = \mathbf{Q}$$

These two steps can then be combined to give the following system of matrix equations:

$$\begin{bmatrix} \mathbf{P}_0 & \mathbf{P}_1 & \ldots & \mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{Q}_D \\ \mathbf{O} & \mathbf{I} \\ \vdots & \mathbf{O} \\ \vdots & \vdots \\ \mathbf{O} & \mathbf{O} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{Q} \end{bmatrix}$$

Continuing this process up to and including $d_p - 1$ yields the following equation:

$$\underbrace{\begin{bmatrix} \mathbf{P}_0 & \mathbf{P}_1 & \dots & \mathbf{P}_m \end{bmatrix}}_{\mathbf{P}_\Sigma} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{Q}_D & \mathbf{Q}_D^2 & \cdots & \mathbf{Q}_D^{d_p-2} & \mathbf{Q}_D^{d_p-1} \\ \mathbf{O} & \mathbf{I} & \mathbf{Q}_D & & & \mathbf{Q}_D^{d_p-2} \\ \vdots & \ddots & \mathbf{I} & \ddots & & \vdots \\ & & \ddots & \ddots & \mathbf{Q}_D & \\ & & & \ddots & \mathbf{I} & \mathbf{Q}_D \\ \mathbf{O} & \cdots & \cdots & \cdots & \mathbf{O} & \mathbf{I} \end{bmatrix}}_{\mathbf{Z}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{Q} & \mathbf{Q}^2 & \dots & \mathbf{Q}^{d_p-1} \end{bmatrix}}_{\mathbf{Q}_\Sigma}$$

which will be denoted as $\mathbf{P}_\Sigma \mathbf{Z} = \mathbf{Q}_\Sigma$, where both $\mathbf{Z}$ and $\mathbf{Q}_\Sigma$ are known. Note that $\mathbf{Z}$ is an upper-triangular matrix with only ones on the diagonal. Therefore, all eigenvalues of $\mathbf{Z}$ equal one, which makes it invertible. In addition, $\mathbf{I} - \mathbf{Z}$ is an upper triangular matrix with only zeros on the diagonal, and therefore it is nilpotent [8, Proposition 8.18]. Note that it is possible to use the Neumann series to write the matrix inverse as an infinite sum, namely $\mathbf{Z}^{-1} = \sum_{k=0}^{\infty} (\mathbf{I} - \mathbf{Z})^k$ [9, Lemma 6.6]. Since $\mathbf{I} - \mathbf{Z}$ is nilpotent, $\exists N_0 > 0$ such that $\forall \eta > N_0$ it holds that $(\mathbf{I} - \mathbf{Z})^\eta = \mathbf{O}$. Therefore, $\mathbf{Z}^{-1} = \sum_{k=0}^{N_0} (\mathbf{I} - \mathbf{Z})^k$. Consequently:

$$\mathbf{P}_\Sigma = \mathbf{Q}_\Sigma \sum_{k=0}^{N_0} (\mathbf{I} - \mathbf{Z})^k$$

Lastly, let $\mathbf{I}_\Sigma$ be the $d_p|V| \times |V|$ matrix consisting of d_p identity matrices stacked on top of each other. Then, it directly follows that $\boldsymbol{\mathcal{P}} = \mathbf{P}_\Sigma \mathbf{I}_\Sigma$ and therefore:

$$\boldsymbol{\mathcal{P}}_{d_a} = \mathbf{Q}_\Sigma \left(\sum_{k=0}^{N_0} (\mathbf{I} - \mathbf{Z})^k \right) \mathbf{I}_\Sigma \tag{3.20}$$

3.4. Extension

In the previous chapter, the assumption was made that $d_a = d_p$. However, in reality, it is not a realistic assumption that an attack lasts the entire patrol. Therefore, this assumption is relaxed to $d_a \leq d_p$. To do so, the methodology of Section 3.3 is applied to equation (2.2):

$$\begin{aligned} \mathcal{U} &= \sum_{t=0}^{d_p-d_a} \sum_{v=1}^{n} \mathbb{P}(X_t = v \cup \dots \cup X_{t+d_a-1} = v) y_{\text{v,t}} \\ &= \sum_{t=0}^{d_p-d_a} \sum_{v,w=1}^{n} \mathbb{P}(X_t = w \cup \dots \cup X_{t+d_a-1} = w \mid X_t = v)\, \mathbb{P}(X_t = v) y_{\text{w,t}} \end{aligned}$$

Again, it should be noted that X represents a time-homogeneous Markov process. As such, it is time-invariant, which means that the conditional probabilities may be shifted

$$= \sum_{t=0}^{d_p-d_a} \sum_{v,w=1}^{n} \mathbb{P}(X_0 = w \cup \ldots \cup X_{d_a-1} = w \mid X_0 = v)\, \mathbb{P}(X_t = v) y_{\mathrm{w,t}}$$

$$= \sum_{t=0}^{d_p-d_a} \sum_{v,w=1}^{n} (p_{d_a})_{\mathrm{v,w}}\, \mathbb{P}(X_t = v) y_{\mathrm{w,t}}$$

$$= \sum_{t=0}^{d_p-d_a} (\varkappa \mathbf{Q}^t) \boldsymbol{\mathcal{P}}_{d_a} \mathbf{Y}_t \tag{3.21}$$

Here, $\mathbf{Y}_t$ represents column t of $\mathbf{Y}$. Since the matrix $\boldsymbol{\mathcal{P}}_{d_a}$ depends only on $\mathbf{Q}$ and d_a, the characterization derived in Equation (3.20) can still be used. The following section compares the asymptotic time complexity of this closed-form characterization to that of the naive method which considers every patrol individually. Afterward, Chapter 4 uses Equation (3.21) to find various patrol strategies for different underlying assumptions to illustrate some possible applications and to test the runtime complexity in practice.

3.5. Time Complexity

Now that a closed-form expression for $\mathcal{U}$ has been found, it is possible to objectively compare the time complexity of this method to the method which iterates over every patrol. For the analysis, it is assumed that the graph G is complete. So, as also mentioned in Chapter 2, the patroller has $|V|^{d_p}$ possible patrols she can take. Algorithm 1 shows the step-by-step breakdown of calculating the interception probability by considering every patrol. To reiterate, an attack is carried out on a single vertex $v \in V$, starts at a certain time step t and lasts until time step $t + d_a - 1$. Line 4 checks whether a given patrol would intercept a specific attack. So, at most, steps t up to and including $t + d_a - 1$ of the patrol would have to be considered to check whether the patrol visits vertex v during this interval. As such, line 4 is $\mathcal{O}(d_a)$ operations. The calculation of the probability in lines 5-7 is $\mathcal{O}(1)$, so this is dominated by checking whether the interception takes place at all. As can be seen in lines 2 and 3, this is repeated $(d_p - d_a + 1)|V|^{d_p+1}$ times. So, the iterative method which considers every patrol is $\mathcal{O}\left(d_a(d_p - d_a + 1)|V|^{d_p+1}\right)$.

Algorithm 1: Interception probability: iterative method

Output: Interception probability

1 probability ← 0
2 **foreach** *patrol in* $|V|^{d_p}$ *patrols* **do**
3   **foreach** *attack in* $(d_p - d_a + 1)|V|$ *attacks* **do**
4   intercepted ← $\mathbb{1}_{\{\text{patrol intercepts attack}\}}$
5   **if** *intercepted* **then**
6   probability ← probability + $\mathbb{P}$(patrol) $\mathbb{P}$(attack)
7   **end**
8   **end**
9 **end**
10 **return** probability

Algorithm 2 shows how Equation (3.21) can instead be used to calculate the interception probability. Line 3 requires careful consideration to find the time complexity. Recall that $\mathbf{Z}$ is an upper triangular matrix of dimensions $d_a|V| \times d_a|V|$. As stated in [10, Section 3.2], solving a standard triangular system of equations of this size is $\mathcal{O}(d_a^2|V|^2)$. However, since $\mathbf{Q}_\Sigma$ is of size $d_a|V| \times |V|$ rather than

$d_a|V| \times 1$ as in the standard case, it actually takes $\mathcal{O}(d_a^2|V|^3)$ time to find $\mathbf{P}_\Sigma$. Then $\mathbf{P}_\Sigma \in \mathbb{R}^{|V| \times d_a|V|}$ and $\mathbf{I}_\Sigma \in \mathbb{R}^{d_a|V| \times |V|}$. Since line 4 represents a standard matrix product, this is $\mathcal{O}(d_a|V|^3)$[1]. Lines 1-2 are $\mathcal{O}(1)$, so finding $\mathbf{P}_\Sigma$ dominates lines 1, 2 and 4. Lines 6 through 8 are all matrix-vector products, whereas line 9 is a matrix-matrix product. Since the matrices are all the same size, the operation in line 9 dominates those of lines 6 through 8. Both $\mathbf{Q}$ and $\mathbf{Q}^t$ are $|V| \times |V|$ matrices, so this operation is $\mathcal{O}(|V|^3)$. Therefore, lines 5 through 10 are $\mathcal{O}\left((d_p - d_a + 1)|V|^3\right)$. In total, then, the entire algorithm is $\mathcal{O}\left(d_a^2|V|^3 + (d_p - d_a + 1)|V|^3\right) = \mathcal{O}\left(\max\{d_a^2, d_p - d_a + 1\}|V|^3\right)$.

Algorithm 2: Interception probability: matrix-multiplication method

Output: Interception probability

1 probability ← 0
2 $Q^0 \leftarrow \mathbf{I}$
3 $\mathbf{P}_\Sigma \leftarrow$ solve $\mathbf{P}_\Sigma \mathbf{Z} = \mathbf{Q}_\Sigma$
4 $\boldsymbol{\mathcal{P}}_{d_a} \leftarrow \mathbf{P}_\Sigma \mathbf{I}_\Sigma$
5 **for** $t \in \{0, 1, \dots, d_p - d_a + 1\}$ **do**
6     $v_1 \leftarrow \boldsymbol{\varkappa} \mathbf{Q}^t$
7     $v_2 \leftarrow v_1 \boldsymbol{\mathcal{P}}_{d_a}$
8   probability ← $v_2 \mathbf{Y}_t$
9     $\mathbf{Q}^{t+1} \leftarrow \mathbf{Q}^t \mathbf{Q}$
10 **end**
11 **return** probability

So, while the highest-order term in the naive iterative method is $|V|^{d_p+1}$, this is only $|V|^3$ for the method resulting from Equation (3.21). Thus, the matrix-multiplication method of finding the interception probability as proposed in this thesis does indeed constitute a significant improvement in the asymptotic runtime complexity compared to the naive method that iterates over all patrols.

[1]This Big-O assumes a naive matrix product implementation. In practice, there are more specialized algorithms that can reduce the time complexity.

4 Application

In the previous chapter, a matrix $\mathcal{P}_{d_a}$ is derived which can be used to find the interception probability of a patrol given a starting distribution $\varkappa$ and an attack distribution $\mathbf{Y}$. As mentioned in Chapter 2, this result is derived with the purpose of using it to find an optimal patrol strategy for the patroller. In this chapter, different notions of optimality are discussed and optimized for to illustrate the potential optimization results.

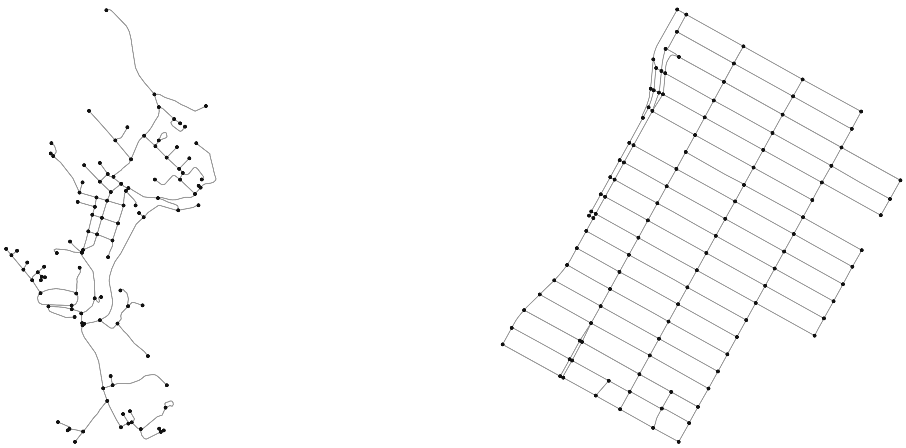
Recall that the patroller can choose the probability distribution $\varkappa$ over the starting positions and the transition matrix $\mathbf{Q}$, whereas the attacker can choose the probability distribution $\mathbf{Y}$ over the graph vertices and the attack times. Each of the following sections seeks to maximize an objective function with respect to $\varkappa$ and $\mathbf{Q}$. However, the assumptions that underlie this optimization and, consequently, how $\mathbf{Y}$ is implemented vary. In Section 4.1.1, it is assumed that the attacker is a rational actor who also seeks to maximize his expected outcome by minimizing the interception probability. In Section 4.1.2, it is no longer assumed that the attacker also seeks to optimize his strategy. Instead, it is assumed that he wishes to be as unpredictable as possible by attributing an equal probability to every possible combination of attack location and time. Lastly, in Section 4.1.3, the attacker strategy is only used to evaluate the effectiveness of the defender strategy, but no longer for optimization. Rather than maximizing the value $\mathcal{U}$ of the game, this section seeks to explicitly maximize the minimum visit probability over all vertices for the patroller.

The analyses presented in this chapter are facilitated by the rapid computation of $\mathcal{P}$ and $\mathcal{U}$, as enabled by the methodology derived in Chapter 3. At the end of Section 4.2, a brief analysis shows that the theoretical improvement in runtime found in Section 3.5 is verifiably achieved in practical applications, too.

4.1. Methodology

Two locations are used to illustrate how Equation (3.21) can be applied by assessing the effectiveness of different strategies and scenarios. Firstly, the Scottish town of *Tobermory* is used as an example of an organically shaped area. As can be seen in Figure 4.1a, the town is only sparsely connected. There are many dead ends, as well as areas that are only connected by one or two edges. In contrast, *Hell's Kitchen* in New York, as seen in Figure 4.1b, has a gridlike structure. This visual insight is also supported by the average degree of the vertices in both graphs. In the Tobermory graph, each vertex has on average 2.25 neighbors, whereas this is 3.45 for Hell's Kitchen. For a fair comparison, the two areas are similarly sized, with both graph representations containing 113 vertices. It is assumed that a patrol lasts for 50 time steps. This constitutes a significant size increase compared to the example given in [2], where $|V| = 6$, $d_p = 5$ and $d_a = 3$.

The subsections below briefly describe three different scenarios, the underlying assumptions for each, and the form of the objective function in each case. Objective function (4.1) is then optimized by using



(a) Tobermory, Scotland, UK

(b) Hell's Kitchen, New York, USA

Figure 4.1: Graph representations of the analyzed areas.

gradient descent-ascent, leading to the *Nash Heuristic* (NH) patrol strategy. Equations (4.2) and (4.3) are optimized using gradient ascent, resulting in the *Optimized for a Homogeneous Attacker* (OHA) and *Maximize the Minimum Visit probability* (MMV) patrol strategies, respectively [1].

In Section 4.2, these strategies are evaluated by calculating the probability of interception for different attack strategies. These attack strategies include a Nash heuristic attack strategy, which is also obtained by optimizing Equation (4.1) using gradient descent-ascent, and the homogeneous strategy used to obtain the OHA patrol strategy. In addition, Monte Carlo simulations are used to test the stability of the patrol strategies by randomly varying the attack strategy.

4.1.1. Nash Heuristic equilibrium

A core definition in game theory is that of a Nash equilibrium. In a multiplayer game, a Nash equilibrium is achieved if "each player's strategy is optimal against those of the others" [11]. Many, if not all, zero-sum games have a mixed Nash equilibrium, yet for large game-theoretic problems it is notoriously hard to find this equilibrium even if it exists. If a Nash equilibrium exists and the patroller sticks to the corresponding strategy, it is by definition impossible for the attacker to lower the value $\mathcal{U}$ of the game (the interception probability) by unilaterally diverging from his Nash equilibrium strategy. However, with the specific formulation of the game value $\mathcal{U}$ used in this paper, it is not directly clear whether a proper mixed Nash equilibrium exists. Proving or disproving that it exists is beyond the scope of this paper. Therefore, while applying gradient descent-ascent as explained in [12] could result in convergence to a (local) Nash equilibrium, this need not be the case. As such, the resulting strategy is referred to as the *Nash Heuristic* (NH) strategy rather than a proper Nash strategy.

Recall that the attacker seeks to minimize the probability of interception, that is, the value of the game. The patroller, on the other hand, wants to maximize this same interception probability by adjusting her strategy to that of the attacker. Equation (4.1) then shows the optimization problem that is considered in this case. Here, $\mathbf{Q}_\Sigma, \mathbf{Z}$ and $\mathbf{I}_\Sigma$ are as defined in Section 3.3. The constraints ensure that all entries of $\boldsymbol{\varkappa}$, $\mathbf{Q}$ and $\mathbf{Y}$ are probabilities and that the correct elements sum to unity. Furthermore, as mentioned in Chapter 2, $\mathbf{A}$ is the adjacency matrix of the graph G on which the patrolling game is played. Therefore, the constraint $0 \leq q_{i,j} \leq a_{i,j}$ ensures that $\mathbb{P}(X_{t+1} = v_2 \,|\, X_t = v_1) = 0$ if there is no edge between vertices v_1 and v_2.

[1] Optimization is done in Python using the PyTorch library. The full code can be found at https://github.com/koningt/MarkovianPatrolling.git.

$$\begin{aligned}
\max_{\varkappa,\mathbf{Q}} \min_{\mathbf{Y}} \quad & \sum_{t=0}^{d_p-d_a} \boldsymbol{\varkappa}\, \mathbf{Q}^t (\mathbf{Q}_\Sigma\, \mathbf{Z}^{-1}\, \mathbf{I}_\Sigma)\, \mathbf{Y}_t \\
\text{s.t.} \quad & 0 \le \varkappa_\mathrm{i} \le 1 && \forall i \in \{1, \dots, |V|\} \\
& 0 \le y_{\mathrm{v,t}} \le 1 && \forall v \in \{1, 2, \dots, |V|\}, \forall t \in \{0, 1, \dots, d_p - d_a\} \\
& 0 \le q_{\mathrm{i,j}} \le a_{i,j} && \forall\, i, j \in \{1, \dots, |V|\} \\
& \textstyle\sum_j q_{\mathrm{i,j}} = 1 && \forall\, i \in \{1, \dots, |V|\} \\
& \textstyle\sum_i \varkappa_\mathrm{i} = 1 \\
& \textstyle\sum_{v,t} y_{\mathrm{v,t}} = 1
\end{aligned} \tag{4.1}$$

4.1.2. Unknown attacker rationality

The previous section is based on a process where both the attacker and the defender seek to anticipate what the other will do, and in doing so hope to find a mixed strategy such that neither party can unilaterally alter the interception probability. A different dynamic is obtained when assuming that the attacker wants to be as unpredictable as possible by making all attack possibilities equiprobable. So, let $\overline{\mathbf{Y}} \in \mathbb{R}^{n\times(d_p-d_a+1)}$ be the homogeneous distribution of the attacker. Stated differently, $\overline{y}_{\mathrm{v,t}} = \mathbb{P}_{\mathcal{A}}(A = (v,t)) = \frac{1}{n(d_p-d_a+1)}$. The objective function of the optimization problem as shown in Equation (4.2) is quite similar to that of Section 4.1.1. The main difference is that the attacker strategy is fixed in this case, so optimizing the objective function will not return an optimized $\mathbf{Y}$. The resulting $\boldsymbol{\varkappa}$ and **Q** are jointly called the *Optimized for a Homogeneous Attacker* (OHA) patrol strategy.

$$\begin{aligned}
\max_{\varkappa,\mathbf{Q}} \quad & \sum_{t=0}^{d_p-d_a} \boldsymbol{\varkappa}\, \mathbf{Q}^t\, \boldsymbol{\mathcal{P}}_{d_a}\, \overline{\mathbf{Y}}_t \\
\text{s.t.} \quad & 0 \le \varkappa_\mathrm{i} \le 1 && \forall i \in \{1, \dots, |V|\} \\
& 0 \le q_{\mathrm{i,j}} \le a_{i,j} && \forall\, i, j \in \{1, \dots, |V|\} \\
& \textstyle\sum_j q_{\mathrm{i,j}} = 1 && \forall\, i \in \{1, \dots, |V|\} \\
& \textstyle\sum_i \varkappa_\mathrm{i} = 1
\end{aligned} \tag{4.2}$$

4.1.3. Maximize minimum visit probability

The last approach can be seen as a risk avoidant approach. The attacker can observe the defender's patrol strategy and attack at the place and time where the patroller is least likely to pass through. Note that $\left(\boldsymbol{\varkappa}\, \mathbf{Q}^t\, \boldsymbol{\mathcal{P}}_{d_a}\right)_v$ contains the probability that, throughout the duration of an attack starting at time t, vertex v is visited at least once. Therefore, $\min\left(\boldsymbol{\varkappa}\, \mathbf{Q}^t\, \boldsymbol{\mathcal{P}}_{d_a}\right)$ is the smallest visit probability for this interval, and so $\min\left\{\min\left(\boldsymbol{\varkappa}\, \mathbf{Q}^t\, \boldsymbol{\mathcal{P}}_{d_a}\right) \mid t = 0, 1, \dots, (d_p - d_a)\right\}$ is the smallest visit probability for all attack intervals. This corresponds to the smallest possible interception probability, and thus the smallest possible value of the game. The resulting optimization problem is shown in Equation (4.3). Maximizing the smallest visit probability is then equivalent to ensuring that no vertex is visited significantly less frequently, avoiding exploitable blind spots. The resulting patrol strategy is henceforth referred to as the *Maximize Minimum Visit probability* (MMV) patrol strategy.

$$\begin{aligned} \max_{\varkappa,\mathbf{Q}} \quad & \min\left\{\min\left(\varkappa \mathbf{Q}^t \mathcal{P}_{d_a}\right) \mid t = 0, 1, \dots, (d_p - d_a)\right\} \\ \text{s.t.} \quad & 0 \le \varkappa_i \le 1 & \forall i \in \{1, \dots, |V|\} \\ & 0 \le q_{i,j} \le a_{i,j} & \forall\, i, j \in \{1, \dots, |V|\} \\ & \textstyle\sum_j q_{i,j} = 1 & \forall\, i \in \{1, \dots, |V|\} \\ & \textstyle\sum_i \varkappa_i = 1 \end{aligned} \tag{4.3}$$

4.2. Results

Optimization of the objective functions in Sections 4.1.1 through 4.1.3 resulted in different possible $\varkappa$ and $\mathbf{Q}$, jointly called a patrol strategy, each with their own underlying assumptions. However, these assumptions need not hold up in reality. An attacker could, for instance, use a homogeneous random draw to decide where and when to attack, while the defender had assumed that the attacker would rationally optimize the distribution over his strategy space. It is therefore important to check how well each defense strategy holds up when the assumptions underlying the strategy are not met. Some core results are illustrated and explained below.

As mentioned above, it might be possible for the attacker to observe the movements of the defender and choose a place and time to attack when the probability of interception is lowest. However, the interception probability can never be lower than the smallest visit probability as set out in Section 4.1.3. Figure 4.2 shows this minimum visit probability for the three different patrol strategies for $d_a = 5$ (4.2a) and $d_a = 10$ (4.2b), respectively. Interestingly, the OHA patrol strategy has minimum visit probabilities of zero. Therefore, this patrol contains specific vertices which will never be visited during a certain attack window. It should be noted that, even though the MMVP patrol is specifically optimized for this metric, the NH patrol strategy performs almost as well if not better in most cases. Specifically, in Hell's Kitchen with an attack duration of 10 time steps, the NH strategy has a significantly higher minimum visit probability, as can be seen in Figure 4.2b. The reason for this is not entirely clear. A possible explanation is that objective function 4.3 converged to a suboptimal local maximum, but proving this would require additional analyses.

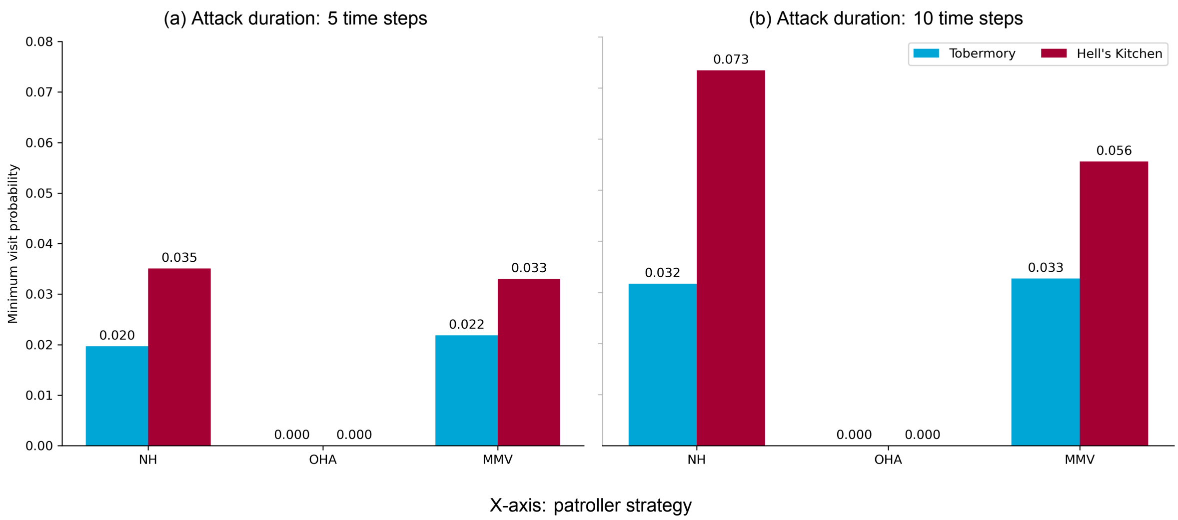


Figure 4.2: Bar plot showing the minimum visit probabilities for different optimized patrol strategies. For each patrol strategy and graph, the minimum is taken over all vertices for all possible attack intervals ($d_a = 5$ or $d_a = 10$) during the patrol ($d_p = 50$).

Two additional Monte Carlo simulations were run to test the stability of the three different patrol strategies. Algorithm 3 briefly describes the procedure that was followed to obtain the results. The first round of simulation iterations uses a Gaussian distribution $\mathcal{N}(0, 1)$ to randomly and independently draw each entry of $\mathbf{Y}$, whereas the second round uses a Cauchy distribution $\mathcal{C}(0, 1)$. The softmax function is then applied to the matrix $\mathbf{Y}$ to ensure that the entries sum to unity. In the Gaussian simulations, the attack probabilities $y_{\mathrm{v,t}}$ are roughly homogeneously distributed. However, a Cauchy distribution has significantly heavier tails. As a result, it is more likely that the random draw for a specific entry of $\mathbf{Y}$ will result in a notably higher value than the random draw for the other entries. Combined with the exponentiation used when applying the softmax function, the attacker is then likely to attribute a high attack probability to a single vertex and time step.

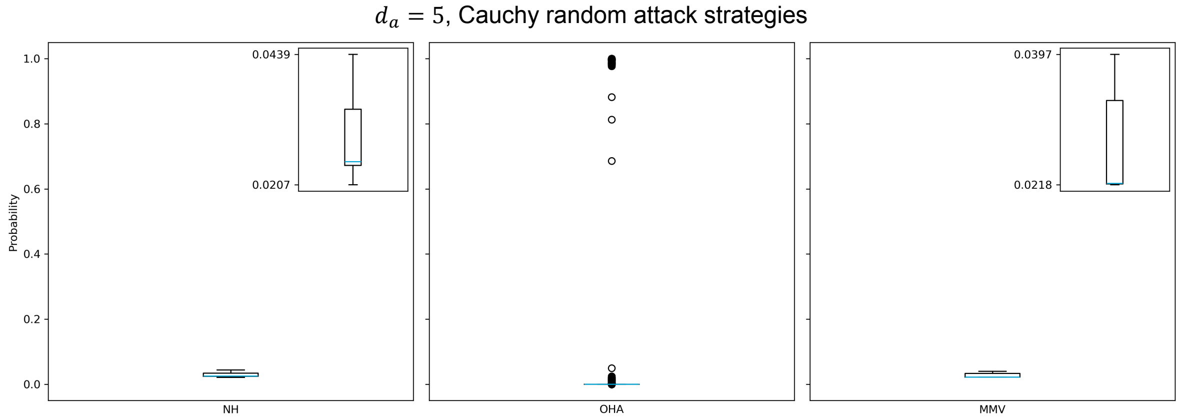


Figure 4.3: Boxplots with the interception probabilities of three patrol strategies in Tobermory for 10 000 Cauchy $\mathcal{C}(0, 1)$ random attack strategies.

Again, a few results stand out. Firstly, as can be seen in Figure 4.3, the OHA patrol strategy is significantly less stable than the NH and MMV strategies when the attack strategy resembles a Cauchy distribution. In other words, there are certain vertices and time steps where interception is almost certain when using the OHA strategy, but also many vertices and time steps where interception is almost guaranteed not to happen. In particular, the median interception probability is 6.84×10^{-11}. This is an expected outcome, as the OHA strategy optimizes for an unweighted sum of interception probabilities. In other words, a high probability of interception in some nodes can compensate for a low probability in other nodes. This can also be seen in Figure 4.5. For every attack window of five time steps during the patrol, the probability is calculated for each vertex that a patrol will visit that vertex. Then, for every patrol strategy and vertex, the average is taken over the probabilities for all attack windows. Since the majority of probabilities are quite small, the square roots of the probabilities are shown to create a clearer distinction between the vertices. The figure clearly shows that, while the average vertex visit probabilities of the NH and MMV strategies are distributed quite homogeneously, the OHA strategy does indeed visit certain vertices with a very high probability on average but most other vertices with a probability close to zero.

At the same time, when comparing Figure 4.3 with the scenario in which the attack strategy is drawn from a Gaussian distribution as visualized in Figure 4.4, the distributions for the MMV strategies appear to be quite similar. In fact, a Wilcoxon signed rank test does not indicate that the distributions are significantly different (p =.752)[2]. The same result holds for $d_a = 10$ (p =.272). Interestingly, the OHA strategy is significantly affected by the attack strategy, but is not significantly affected by the graph structure if the attack strategy is drawn from a Gaussian distribution. Although not visualized, a t-test showed that, for the OHA strategy, the Tobermory interception probability neither differs significantly from that of Hell's Kitchen for $d_a = 5$ (p =.652) nor for $d_a = 10$ (p =.939). Consequently, these results tentatively indicate that the MMV strategy is more robust against different attack distributions, whereas the graph-optimized OHA strategies yield a more stable interception probability against different graph structures if the attack strategy is drawn from a Gaussian distribution.

These results certainly do not provide a comprehensive comparison of the three patrol strategies under varying circumstances. Instead, they are merely meant to illustrate the results that can be obtained by

[2]The differences between the two results were not normally distributed, so a t-test could not be used.

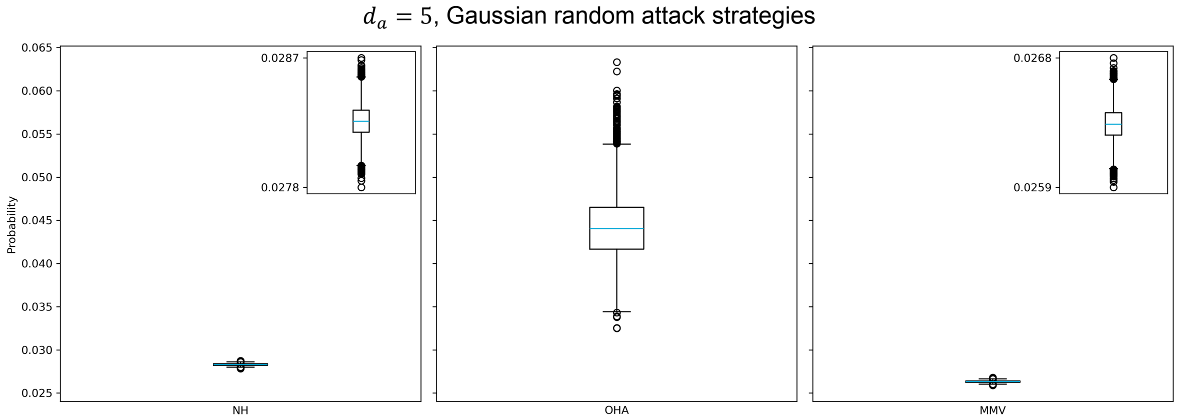


Figure 4.4: Boxplots with the interception probabilities of three patrol strategies in Tobermory for 10 000 $\mathcal{N}(0, 1)$ random attack strategies.

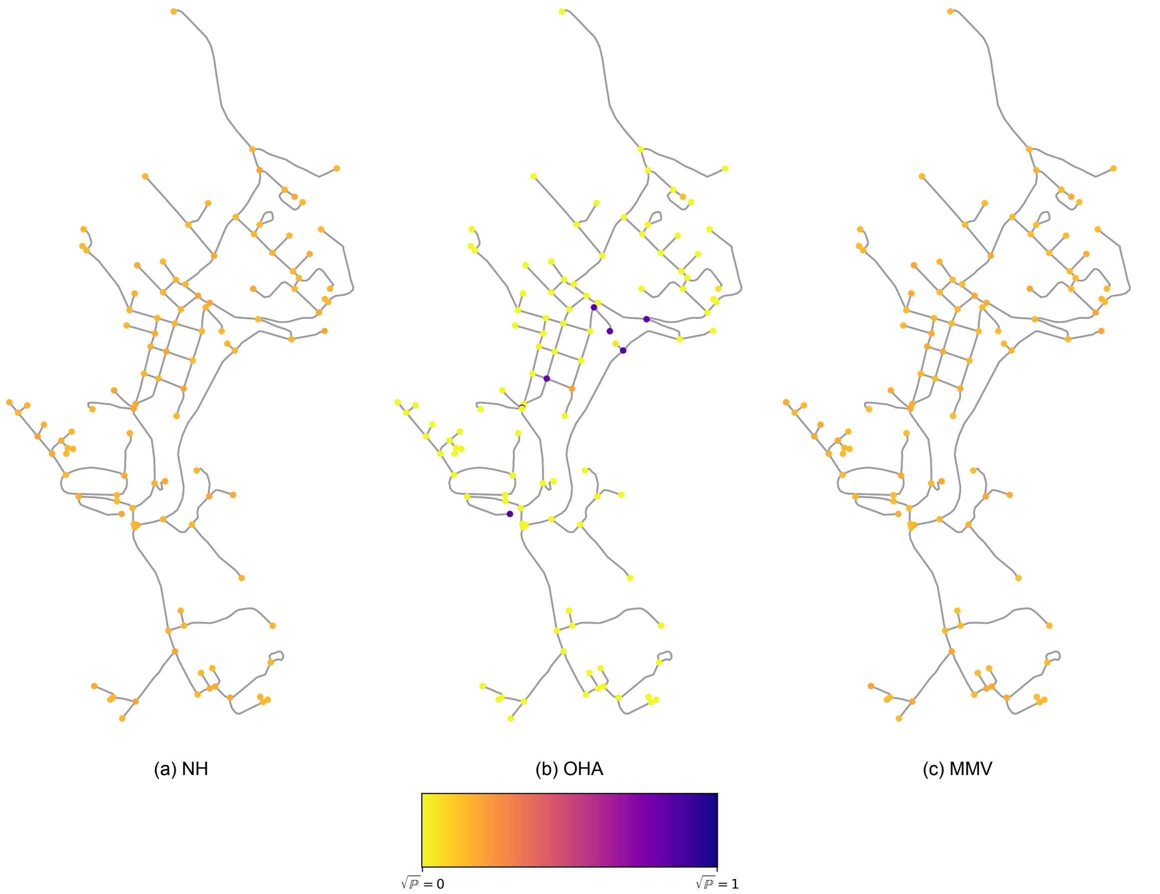


Figure 4.5: A heatmap showing the square root of the average visit probabilities for the NH, OHA and MMV patrol strategies. The averages are over all possible attack intervals ($d_a = 5$) during the patrol ($d_p = 50$).

using the method provided in this paper, specifically in Chapter 3. Because the interception probability is calculated analytically, gradient ascent could directly be applied to the the objective functions in Sections 4.1.1 through 4.1.3. In addition, due to the lower computational complexity required to find $\mathcal{U}$ by using Equation (3.21), it was possible to run extensive Monte Carlo simulations in a limited time.

Section 3.5 already showed the theoretical improvement in asymptotic runtime. Although this was confirmed in practice by observation during the simulations discussed above, it is important to make the improvement concrete by quantizing it. Recall that for a patrol that lasts 50 time steps ($d_p = 50$) the patroller starts at a vertex and then decides 49 times to either take a step or remain in place. Therefore, the total number of unique patrols corresponds to the sum of all entries in the 49th power of the adjacency matrix. This means that there are 2.1199×10^{31} different patrols in Tobermory and 1.0159×10^{35} in Hell's Kitchen. Thus, it does not require further explanation that it is intractible to calculate $\mathcal{U}$ by considering each patrol. In contrast, constructing matrix $\boldsymbol{\mathcal{P}}_{d_a}$ and then calculating $\mathcal{U}$ using equation (3.21) takes less than 0.5 seconds for both graphs[3], as can be seen in Figure 4.6. During this analysis, $\mathbf{Q}$, $\boldsymbol{\varkappa}$ and $\mathbf{Y}$ were kept homogeneous. Since the complexity of the operations of the matrix method only depends on d_a, d_p, and $|V|$, the results in Figure 4.6 are deemed representative of all possible $\mathbf{Q}$, $\boldsymbol{\varkappa}$ and $\mathbf{Y}$ that can be defined on the two graphs for $d_p = 50$. As such, it can be concluded that the lower computational complexity is achieved not only in theory as demonstrated in Section 3.5, but also in practice.

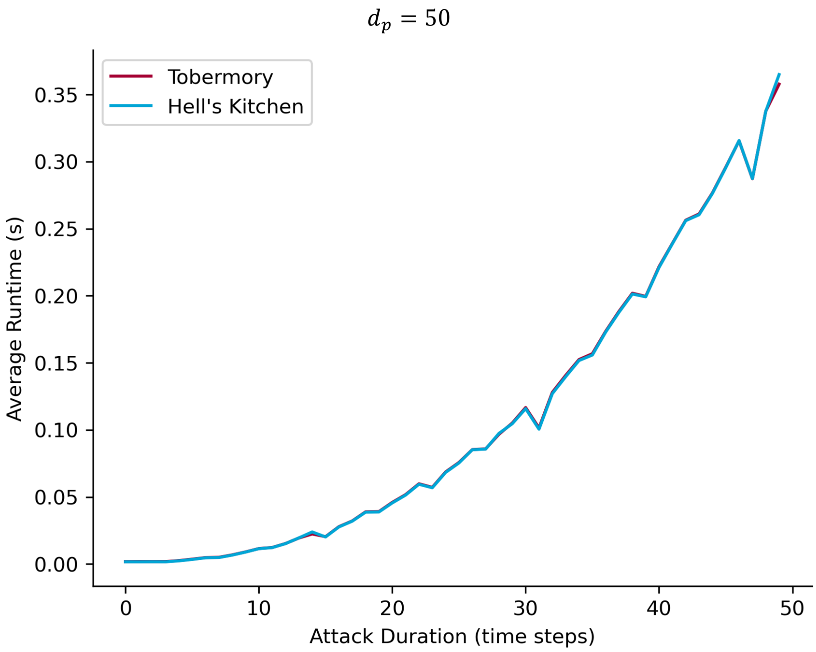


Figure 4.6: Average runtime (100 iterations) required to calculate $\mathcal{U}$ by using the methodology in Section 3.3 to build $\boldsymbol{\mathcal{P}}_{d_a}$ and subsequently applying Equation (3.21) for $d_a \in \{1, 2, ..., 50\}$.

[3]Code executed with GPU acceleration on a MacBook Pro with an M3 Pro chip (18 GPU cores).

Algorithm 3: Pseudocode of the Monte Carlo simulation algorithm used to evaluate the NH, OHA, and MMV patrol strategies.

```
for graph $G = (V, E) \in$ {Tobermory, Hell's Kitchen} do
    for $d_a \in \{5, 10\}$ do
        for patrol strategy $(\mathbf{Q}, \varkappa) \in$ {NH, OHA, MMV} do
            /* Note that each combination of $d_a$ and graph $G$ has its own
               unique NH, OHA and MMV strategies */
            for probability distribution $\mathcal{D} \in \{\mathcal{N}(0,1), \mathcal{C}(0,1)\}$ do
                for $i \in \{1, 2, \ldots, 10000\}$ do
                    seed ← SeedList[i]
                    /* Use fixed seeds to ensure a fair comparison */
                    for $v \in V, t \in \{0, \ldots, d_p - d_a\}$ do
                        $y_{\mathrm{v,t}} \leftarrow \mathcal{D}$ random draw
                    end
                    $\mathbf{Y} \leftarrow$ SoftMax($\mathbf{Y}$)
                    $\boldsymbol{\mathcal{P}}_{d_a} \leftarrow \mathbf{Q}_\Sigma \mathbf{Z}^{-1} \mathbf{I}_\Sigma$
                    /* Methodology from Section 3.3 */
                    probability ← $\sum_{t=0}^{d_p - d_a} \varkappa \mathbf{Q}^t \boldsymbol{\mathcal{P}}_{d_a} \mathbf{Y}_t$
                    /* Methodology from Section 3.4 */
                    append probability to values[scenario]
                end
            end
        end
    end
end
```

5
Conclusion

This thesis has explored the problem of optimal patrolling strategies in adversarial scenarios using mathematical models and simulations. By focusing on a contained area of interest where a patroller must guard against potential attacks, various patrol strategies have been formulated and analyzed through the lens of patrolling games. The central challenge lies in balancing limited patrol resources against extensive areas requiring protection, making it crucial to determine an effective patrol strategy.

Building on the foundational framework of Alpern et al. [2], this research extends it by incorporating Markovian strategies, which allow for the efficient calculation of interception probabilities. This approach significantly reduces computational complexity, allowing patrol strategies to be derived using gradient descent methods.

Chapter 3 serves as the core body of this research, presenting the mathematical derivation of the different methods to calculate the interception probability. These analytical methods, coupled with numerical simulations, provide a robust means of evaluating and optimizing patrol strategies. The gradient ascent method applied to the Markovian model proved effective in finding suitable strategies within a reasonable computational timeframe.

Chapter 4 demonstrates some practical applications of the methods developed in Chapter 3. The results of preliminary Monte Carlo simulations highlight how the expression of the interception probability as stated in Equation (3.21) enables performance comparisons of various patrol strategies. For instance, while specific patrol strategies like OHA may perform well under certain conditions, their effectiveness can vary depending on the specific probability distribution on all possible attack strategies. This variation is effectively captured and analyzed using the equations of Chapter 3.

Several key findings that emerged from this thesis are:

- Analytical and Numerical Approaches: The gradient ascent method applied to the Markovian model proved effective in finding suitable patrol strategies within a reasonable computational timeframe.
- Impact of Attack Strategy Distributions: Analysis showed that the distribution of attack strategies can significantly influence the interception probability. As might be expected, there is no single effective patrol strategy and a new suitable strategy may need to be determined for each unique instance by comparing several potential strategies.
- Scalability and Practicality: As was shown in Chapter 4, the methods developed in Sections 3.3 and 3.4 are scalable and can be applied to larger instances of patrolling games. This addresses one of the core difficulties of the patrolling game problem - the exponential growth of possible patrol routes. By leveraging the Markovian structure, this bottleneck was mitigated, making the approach practical for real-world applications.

In conclusion, this thesis has developed and provided preliminary validation of a method that can be used to improve the efficiency and effectiveness of patrol strategies. The insights gained from this research not only advance the theoretical understanding of patrolling game problems but also offer practical solutions to improve security in various adversarial settings. The methodologies presented in Chapter 3, and their applications demonstrated in Chapter 4, underscore the practical relevance and potential of these strategies in real-world scenarios.

5.1. Discussion

As was mentioned in the introduction, much research has already been done on patrolling games. This thesis sought to contribute to this body of research by expanding on the Markovian framework to model the problem. This does mean, however, that the different extensions that can be found in the literature have not yet been incorporated into this methodology. Certain extensions can likely be incorporated relatively easily. For instance, Equation (3.20) can be directly modified for a time-inhomogeneous Markov chain. On the one hand, using a time-dependent transition matrix adds additional dimensions of freedom to the patroller's strategy space, which might improve the interception probability. However, a trade-off is the significantly higher computational complexity required to find the optimal patrol strategy since the number of parameters over which to optimize will scale at a rate of $d_p|V|$ instead of $|V|$.

At the same time, the computational complexity of the optimization procedure is already quite high. The Markovian patrol method set forth in this thesis does ease the exponential time-complexity associated with considering every patrol individually, but it still requires a significant number of high-cost operations during the optimization phase. For example, when using gradient descent, the matrix $\boldsymbol{\mathcal{P}}_{d_a}$ as used in Equation (3.20) has to be recalculated after every step. Especially for larger instances, this can result in slow convergence.

Furthermore, the restriction to Markovian patrol strategies can significantly affect the domain of possible probability distributions over the patrols. Consider the scenario where $d_a = d_p = 5$ on the graph drawn in Figure 5.1. The attack could then be intercepted with absolute certainty if the patroller always chooses patrol $(x_1, x_2, x_3, x_2, x_4)$, corresponding to the patrol shown in Figure 2.1. However, both edge (x_2, x_3) and (x_2, x_4) would need to be traversed with certainty at some point. This is impossible in a time-homogeneous first-order Markov patrol. In this small example, it can be verified quite easily that the probability of visiting all vertices during the patrol can never exceed 0.5 when using a first-order Markov patrol. However, by using a time-inhomogeneous Markov chain, it would again be possible to find a patrol that visits all vertices with certainty. Do note that this is a very specific example. As such, it is unknown at this time if this effect is as significant in more realistic settings and, if so, if using a time-inhomogeneous Markov patrol could still resolve this limitation.

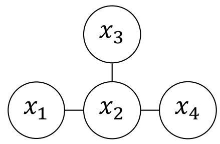


Figure 5.1: Example graph

Lastly, the optimization method using gradient descent/ascent as implemented in Chapter 4 does not guarantee convergence to a global optimum. For example, as mentioned in Section 4.2, the NH patrol strategy had a higher minimum visit probability than the MMV patrol strategy, while the latter is specifically optimized to maximize this probability. If, in fact, it had converged to a global maximum, this could not have been the case. Future research could potentially focus on the implementation of improved optimization methods or the provision of theoretical convergence guarantees.

In addition, future studies could explore the impact of heterogeneous environments where nodes have different values or significance, further complicating the patrolling strategy but potentially leading to more realistic models. Extending the model to include multiple patrollers and attackers could also provide deeper insights into more complex security scenarios, reflecting the collaborative efforts often required in real-world security operations.

Bibliography

[1] K. Hunt and J. Zhuang, "A review of attacker-defender games: Current state and paths forward," *European Journal of Operational Research*, vol. 313, no. 2, pp. 401–417, Mar. 2024. DOI: 10.1016/j.ejor.2023.04.009.

[2] S. Alpern, A. Morton, and K. Papadaki, "Patrolling Games," *Operations Research*, vol. 59, no. 5, pp. 1246–1257, 2011. DOI: 10.1287/opre.1110.0983.

[3] N. Basilico, N. Gatti, and F. Amigoni, "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder," *Artificial Intelligence*, vol. 184-185, pp. 78–123, Jun. 2012. DOI: 10.1016/j.artint.2012.03.003.

[4] S. Alpern, T. Bui, T. Lidbetter, and K. Papadaki, "Continuous Patrolling Games," *Operations Research*, vol. 70, no. 6, pp. 3076–3089, 2022. DOI: 10.1287/opre.2022.2346.

[5] K. Y. Lin, M. P. Atkinson, T. H. Chung, and K. D. Glazebrook, "A graph patrol problem with random attack times," *Operations Research*, vol. 61, no. 3, pp. 694–710, May 2013. DOI: 10.1287/opre.1120.1149.

[6] A. Yolmeh and M. Baykal-Gürsoy, "Patrolling Games on General Graphs with Time-Dependent Node Values," *Military Operations Research*, vol. 24, no. 2, pp. 17–30, 2019. DOI: 10.5711/1082598324217.

[7] R. A. Horn and C. R. Johnson, "The Hadamard Product," in *Topics in Matrix Analysis*, Cambridge University Press, 1991, ch. 5, ISBN: 9780521305877. DOI: 10.1017/CBO9780511840371.

[8] S. Axler, *Linear Algebra Done Right*, 4th ed. Springer, 2024, ch. 8, ISBN: 978-3031410253. DOI: 10.1007/978-3-031-41026-0.

[9] J. van Neerven, "Spectral Theory," in *Functional Analysis*, Cambridge University Press, 2022, ch. 6, pp. 209–226. DOI: 10.1017/9781009232487.008.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, Maryland: The Johns Hopkins University Press, 2013, ISBN: 978-1-4214-0794-4.

[11] J. Nash, "Non-Cooperative Games," *The Annals of Mathematics*, vol. 54, no. 2, p. 286, Sep. 1951. DOI: 10.2307/1969529.

[12] T. Lin, C. Jin, and M. I. Jordan, "On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems," *Proceedings of Machine Learning Research*, vol. 119, pp. 6083–6093, 2020. DOI: 10.48550/arXiv.1906.00331.