

# Multi-Robot Adversarial Patrolling Strategies via Lattice Paths

Jan Buermann\* and Jie Zhang

University of Southampton, UK

{J.Buermann, Jie.Zhang}@soton.ac.uk

## Abstract

In full-knowledge multi-robot adversarial patrolling, a group of robots have to detect an adversary who knows the robots' strategy. The adversary can easily take advantage of any deterministic patrolling strategy, which necessitates the employment of a randomised strategy. While the Markov decision process has been the dominant methodology in computing the penetration detection probabilities, we apply *enumerative combinatorics* to characterise the penetration detection probabilities. It allows us to provide the closed formulae of these probabilities and facilitates characterising optimal random defence strategies. Comparing to iteratively updating the Markov transition matrices, our methods significantly reduces the time and space complexity of solving the problem. We use this method to tackle four penetration configurations.

## 1 Introduction

Multi-robot adversarial patrolling is a well-established problem with numerous security applications including crime prevention [An *et al.*, 2017], stopping piracy, defending critical infrastructure [Oliva *et al.*, 2019], protecting animals, natural reserves, or the environment [Basilico *et al.*, 2017]. In these settings, robots have the benefit of providing a cheaper mobile assistance in monitoring vast open areas [An *et al.*, 2017]. In the general problem, a defender has a team of robots to patrol an area while an adversary tries to penetrate the area. The problem has many characteristics and has been considered with different approaches; we focus on the important setting of finding optimal random strategies for defending polyline graphs against full-knowledge adversaries.

Random strategies play an important role in adversarial patrolling since only these are able to guarantee that an adversary with knowledge of the defender's strategy can be caught [Sak *et al.*, 2008]. Finding an optimal strategy is a two-step process [Agmon *et al.*, 2011]: firstly, determining the probability functions for catching the adversary and, secondly, solving the functions for an optimal random parameter. The first step depends on the movements and paths the

robots can perform and walk, respectively. The current approach is to algorithmically find all paths and calculate the probability using Markov chain models [Agmon *et al.*, 2011; Sless *et al.*, 2014; Sless Lin *et al.*, 2019; Talmor and Agmon, 2017]. These black-box algorithms do not allow us to gain further understanding of the setting or the resulting probabilities, they must be repeated for every combination of the setting, and they have a polynomial space and time complexity. For example, on a closed polyline with  $d$  segments and  $k$  robots the time complexity is  $\mathcal{O}(d^3)$  [Agmon *et al.*, 2011] and the space complexity tends towards  $\mathcal{O}(d^2/k^2)$ . However, prevailing classes of graphs like polylines allow a succinct expression of the number of possible paths.

Consequently, this allows to analytically express the probabilities of random strategies. These expressions not only completely remove the required runtime but allow us to further analyse the relationship between the parameters of the setting. We present a novel approach to model all possible paths of a robot's random strategy as lattice paths. A lattice path is a path along points in an Euclidean space. Lattice path problems aim to count the number of lattice paths given start and end point of the paths, the allowed steps, coordinates restricting the paths, and further path features. Lattice paths have been studied for a long time and are, despite apparent simplicity, powerful. Moreover, they have applications to many problems including physical systems, encoding, probability and statistics [Krattenthaler, 2015].

Our main contribution is the modelling of the robots' paths as lattice paths. For two movement types, and open/closed polylines, we describe the characteristics that influence the robot's paths and how these are translated into lattice path characteristics. We use the lattice path representation to count the number of paths which immediately allow us to explicitly state the probability of catching the adversary in different segments. This removes the first step of calculating the functions, such that the remaining step of solving the system of equations can be done efficiently. Finally, we highlight connections between the parameter and the resulting probabilities.

### 1.1 Related Work

Multi-robot patrolling is an ongoing research area with substantial attention in the past decade [Huang *et al.*, 2019]. The area's four main modelling dimensions are: the environment,

\*Contact Author

the type of adversary, the objective or evaluation, and the approach or method. In terms of overall goal, the area is divided between regular patrolling and adversarial patrolling [Huang *et al.*, 2019]. In regular patrolling the aim is to optimise a frequency related goal; for example, minimising travel time or maximising the visits of important points [Elmaliach *et al.*, 2009]. In contrast, the aim of adversarial patrolling is to defend against an adversary by detecting [Basilico *et al.*, 2012] or handling [Sless Lin *et al.*, 2019] penetration events.

The adversarial model is mostly concerned about the adversary’s knowledge about the robot’s strategy. This ranges from no information over the ability to learn to full knowledge [Agmon *et al.*, 2008]. For example, the learning adversary in the work of Sak *et al.* [2008] collects information on the robot’s visits and uses this data to predict safe attack times. In comparison, we assume that the adversary has full knowledge of the defenders’ strategy which can also be interpreted as the adversary has enough time to learn the strategy. It is well established that a full-knowledge adversary can only be caught using random strategies [Agmon *et al.*, 2011].

The patrolling environments are generally assumed to be discrete [Basilico *et al.*, 2017] or the underlying continuous environment is discretised. For example, Elmaliach *et al.* [2009] divide a 2D environment into cells. Similarly, the nodes in Sea *et al.* [2018] represent coordinates in a plane. Discrete environments might be general graphs [Chevalyere, 2004; Basilico *et al.*, 2012], representing important points, or polylines [Agmon *et al.*, 2011; Elmaliach *et al.*, 2008], representing a perimeter. Furthermore, Agmon *et al.* [2011] show that any continuous fence or perimeter can be represented as a graph with nodes representing sections of equal travel time. Hence, we focus on discrete polylines which can be applied to cover continuous environments as well.

Multi-robot adversarial patrolling also overlaps with the game-theoretic area of security games [Sless Lin *et al.*, 2019; Basilico *et al.*, 2012]. This area is dominated by leader-follower or Stackelberg games [Sless Lin *et al.*, 2019] in which the leader, in this case the defender, makes a move, i.e. commits to a strategy, and then the attacker can respond with a move, i.e. a penetration attempt. The overall goal is to find equilibria that are good for the defender.

However, general graphs and game-theoretic approaches are mostly either NP-hard [Jain *et al.*, 2013] or likely to be NP-hard [Sless Lin *et al.*, 2019]. For example, Chevalyere [2004] and Basilico *et al.* [2012] prove that, in their respective cases, the optimal strategy is NP-hard. Consequently, a great number of works use experimental evaluation of heuristics [Chevalyere, 2004; Elmaliach *et al.*, 2008; Basilico *et al.*, 2009; Basilico *et al.*, 2012; Clempner, 2018; Sea *et al.*, 2018; Huang *et al.*, 2019]. These heuristic approaches are in contrast to our work of finding specific optimal solutions for specific graphs and movement patterns.

Our work is close to the work of Agmon *et al.* [2011] and subsequent papers addressing different aspects like coordinated attacks [Sless *et al.*, 2014], sequential attacks [Sless Lin *et al.*, 2019] and deception [Talmor and Agmon, 2017]. Their general aim is to find optimal polynomial-time patrol strategies for specific graphs and specific goals maximising the minimal probability of the adversary being caught. They

use algorithmic approaches, mainly the construction and evaluation of Markov transition matrices, to find the probabilities of the robots catching the adversary [Agmon *et al.*, 2011] or the number of paths [Sless Lin *et al.*, 2019]. In contrast, we give explicit terms and functions for the number of paths and the probability of catching the adversary which removes the necessity for the respective algorithms.

## 2 Preliminaries

We are considering multi-robot patrolling in a discrete environment assuming discrete time. A polyline with  $d \in \mathbb{N}$  nodes, called segments, shall be defended by  $k$  homogeneous robots against a full-knowledge adversary who tries to penetrate at a target segment  $1 \leq j \leq d$  which requires  $t$  time steps. The polyline is defended if a robot detects the adversary by reaching the targeted segment within the  $t$  time steps. We mostly focus on one robot since Agmon *et al.* [2011] established that the multi-robot case directly follows from the single robot case when synchronised robots are placed equidistant. The polyline can be open or closed to represent a perimeter or a fence [Agmon *et al.*, 2011], respectively.

1. **Perimeter / Circle:** A closed polyline.
2. **Fence / Line:** An open polyline.

The polyline restricts the robot’s movement to two directions, right and left. We differentiate two movement types based on if the robot can freely go left and right at any moment, or if it needs to turn around first [Agmon *et al.*, 2011].

1. **Omnidirectional:** The robot can freely either walk into the segment to the left or the right of the current segment in one time step.
2. **Directional:** The robot can walk into the segment ahead of the current segment in one time step but needs to turn around to change direction which takes  $\tau \geq 1$  time steps.

Since the adversary has full knowledge of the robots’ strategy the robot has to act randomly. The random walk of the robot is a Bernoulli process where in every round the robot does, for both movement types, one of the two possibilities with probability  $p$  and the other action with probability  $1 - p$ . The objective is to determine the best random walk to catch the adversary. This can be achieved by maximising the probability of the segment with minimal probability of catching the adversary, i.e. we seek  $p \in \arg \max_{p' \in [0,1]} \min_{j \in [d]} Pr(j, t, d)$  where  $Pr(j, t, d)$  is the probability that the adversary is caught in segment  $j$  within the  $t$  time steps on a graph of  $d$  segments. This is also referred to as *minmax* approach [Agmon *et al.*, 2011].

The probability depends on the paths the robots can walk. In general, a path is a sequence of consecutive segments representing the robot’s left, right, forward or turn movements.

**Definition 1** (Path / Right Step / Left Step). *A path of length  $l$  is a sequence of consecutive segments, i.e. a vector  $(s_1, s_2, \dots, s_l) \in [d]^l$  where every pair of segments  $(s_i, s_{i+1})$  for  $1 \leq i < l$  are neighbours in the graph. For a path a step to the right or a step to the left is a tuple  $(s_{i-1}, s_i)$  where the second segment is right of or left of the first segment, respectively. Accordingly, the robot moves into a segment from the*

left or clockwise if the step into the segment is a right step; the same applies for moving in from the right.

Since the random walk is a Bernoulli process, describing the probability of a path is straightforward. The outcome set  $\Omega$  consists of all paths from a start segment with a length at most  $t$  and the probability measure  $P_W$  is the obvious Bernoulli process measure.

**Definition 2** (Probability of a Path). *The probability of a path is  $P_W(\omega) = p^Y \cdot (1 - p)^X$  where  $X$  and  $Y$  are the number of actions depending on the movement type.*

Catching the adversary means that within the penetration time  $t$  there is a time step in which the robot is in the segment where the adversary penetrates. This means that if the adversary penetrates segment  $j \in [d]$  and the robot walks a path that contains  $j$ , the robot will detect the adversary. Hence, we can define the probability measure of catching the adversary  $Pr(j, t, d)$  as the probability of reaching the adversary  $Pr(j, t, d) := P(\{\omega | \omega \in \Omega \wedge j \in \omega\}) = \sum_{\omega \in \Omega: j \in \omega} P_W(\omega)$ .

### 3 Omnidirectional Movement in the Circle

We use the case of omnidirectional movement in the circle to introduce the general modelling. This encompasses the different types of paths, devising, based on the different types of paths, the lattice path setting and the combination to get the probability functions. For the rest of this section we consider one robot and a segment  $j$  as the specified target. Without loss of generality the robot is placed in segment 1 since the segment numbering is arbitrary. Moreover, for several robots we can shift the segments according to their position.

The probability of catching the adversary depends on the possible paths and the probability of one path  $P_W = p^Y \cdot (1 - p)^X$  where  $X$  is the number of right or clockwise steps and  $Y$  is the number of left or anticlockwise steps. We determine the valid paths and their number in five steps. Firstly, for short penetration times  $t$  the adversary can always penetrate successfully while for large  $t$  the robots can always catch the adversary. Secondly, the symmetry of the circle means that we only have to consider reaching a segment in one direction and the other directions follow similarly. Thirdly, by the definition of catching we only need to consider paths that end in  $j$  and are at most of length  $t$ . Fourthly, we determine the freedom that leads to the different possible paths. Finally, the major part is modelling the different paths as lattice paths and counting them. Altogether, this gives us the probabilities of the robot reaching the different segments.

#### 3.1 Penetration Time Range

Firstly, only a certain range of  $t$  makes sense. If the adversary requires a long penetration time, the robot can always detect the penetration; if the adversary requires only a short penetration time, the adversary can always avoid the robot. We omit the proof since the result is analogous to Agmon et al. [2011].

**Lemma 1.** *The penetration time for the omnidirectional robot defending a circle is  $\lfloor d/2 \rfloor \leq t \leq d - 2$ .*

#### 3.2 Path Symmetry

Secondly, by virtue of a circle every clockwise path has a similar anticlockwise path, denoted as mirrored path.

**Definition 3** (Mirrored Path). *For a path of length  $l$  let  $(s_1, \dots, s_{l-1})$  be a sequence of left and right steps, i.e.  $s_i \in \{\text{left}, \text{right}\}$  for  $i \in [l - 1]$ . The mirrored path is the path whose left and right step sequence is  $(s'_1, \dots, s'_{l-1})$  with  $s'_i = \text{right}$  if and only if  $s_i = \text{left}$  for all  $i \in [l - 1]$ .*

From this definition it is obvious that the number of left and right steps is the other way around in the mirrored path.

**Observation 1.** *For a path with  $X$  right steps and  $Y$  left steps, the mirrored path has  $Y$  right steps and  $X$  left steps.*

This symmetry means that a set of paths and the set of their mirrored paths has to have the same cardinality. Hence, we can focus on one direction and the other direction follows.

**Lemma 2.** *Moving from left into segment  $j \in \{2, \dots, \lfloor d/2 \rfloor\}$  is mirrored by moving from the right into  $d - (j - 2) \in \{\lfloor d/2 \rfloor + 1, \dots, d\}$  and vice versa.*

*Proof Sketch.* Since the graph is cyclic, considering the mirrored path of any path directly implies the result.  $\square$

#### 3.3 Restriction of Necessary Paths

Thirdly, we restrict the number of paths we have to consider since any path passing  $j$  is covered by a path that ends in  $j$ . In more detail, any path passing through  $j$  can catch the adversary (see Section 2). Moreover, we can divide all paths into sets of paths which share a common prefix, i.e. the path from 1 to the first occurrence of  $j$ . By a simple probability argument, summing up all path probabilities of a set is equivalent to the probability of the prefix since the rest sums up to 1. Hence, we can conclude that we only need to consider the prefixes (paths ending in  $j$ ).

**Observation 2.** *It suffices to consider the subset of paths which end in segment  $j$ , without having passed it before, and are at most of length  $t$ .*

#### 3.4 Freedom of Movement

Fourthly, we consider the robot's freedom and possible movements towards segment  $j$  and away from it. In any case the robot can walk up to  $t$  steps. Within these  $t$  steps the robot needs to reach  $j$  at a distance of  $dist := dist(1, j)$ . This distance is  $j - 1$  for the clockwise direction and  $d - j + 1$  for the anticlockwise direction. By Lemma 1 the robot can reach every segment within  $t$  so we assume that  $t \geq dist$ . Hence, since the robot needs  $dist$  moves to reach  $j$  we have  $t - dist$  additional moves. Considering the direct walk as the starting point, we observe that any step away from  $j$  has to be countered by a step towards  $j$  at some point.

**Observation 3.** *The robot may walk up to  $\lfloor (t - dist)/2 \rfloor$  away from  $j$  which has to be countered by the same number of moves towards  $j$ .*

Hence, by Observation 3 and 2, for every length  $t - dist + 2i$  with  $i \in \{0, 1, \dots, \lfloor dist/2 \rfloor\}$  we have a different amount of steps and their placement determines the robot's path.

#### 3.5 Modelling as Lattice Paths

In order to determine the number of paths we model the setting as lattice paths. The overall result, the number of paths for fixed  $d, j$  and  $i$ , can be expressed as the following entry of Catalan's triangle (see Bailey [1996] and Stanley [2015]).

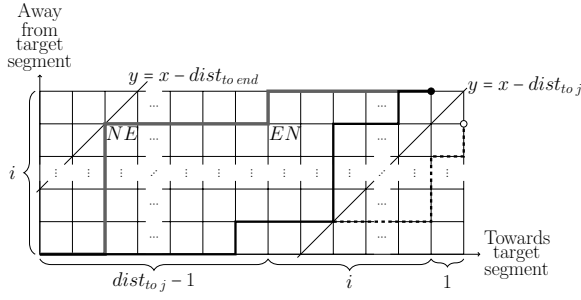


Figure 1: The lattice that represents the movement pattern. The  $x$ -axis are steps in anticlockwise direction and the  $y$ -axis are steps in clockwise direction. The start is the bottom left corner and the finish is the top right corner. The disk (filled circle) and the circle are the end points of the valid and the invalid paths, respectively. The thicker grey line is a valid path whereas the solid black path is an illegal path that touches the line  $y = x - dist_{to j}$ . The dashed path shows how reflecting every move after the first touch of the path and the line leads to the circle. Additionally, a NE-turn and a EN-turn are illustrated on the thicker grey line indicated by NE and EN, respectively. Finally, the line  $y = x - dist_{to end}$  represents the end of fence boundary in the case of the line.

**Theorem 1.** *The number of paths of exactly length  $dist + 2i$  is  $C(dist - 1 + i, i)$ .*

The lattice path setting (see Figure 1) is as follows. The robot’s initial position corresponds to  $(0, 0)$ , the bottom left corner. Starting there, we allow the following lattice paths.

**Definition 4 (Lattice Path).** *A lattice path is a sequence of horizontal and vertical unit steps beginning at  $(0, 0)$  and ending in a specified point.*

The horizontal direction represents movement towards the target and the vertical direction represents movement away from the target. Hence, point  $(x, y)$  corresponds to the robot being in segment  $y - x + 1$  if  $x \leq y$  or else segment  $y - x + d + 1$ . The fact that we consider only paths that end in  $j$  is incorporated by considering only paths that do not cross the line  $y = x - dist$ .

Finally, while paths that end in  $j$  after  $dist + 2i$  steps correspond to paths that end in  $(dist + i, i)$ , for technical reasons, we will consider paths to coordinate  $(dist - 1 + i, i)$ . The number of paths is not affected by this. Logically, every path that ends in  $j$  after  $dist + 2i$  steps must end in  $j - 1$  after  $dist - 1 + i$  steps. Graphically, considering Figure 1, the only way to reach  $(dist + i, i)$  is by going through  $(dist - 1 + i, i)$ .

### 3.6 Counting the Lattice Paths

The described lattice path setting allows us to show that the number of lattice paths from  $(0, 0)$  to  $(dist - 1 + i, i)$  amounts to the number of paths stated in Theorem 1. We do this by counting all lattice paths from the origin to the end point and subtract all paths that touch or cross the line  $y = x - dist$ .

**Lemma 3.** *The number of lattice paths from  $(0, 0)$  to  $(dist - 1 + i, i)$  not crossing line  $y = x - dist$  is  $C(dist - 1 + i, i)$ .*

*Proof.* The number of unrestricted lattice paths, not restricted by a line, from  $(0, 0)$  to  $(a, b)$  is  $\binom{a+b}{b}$  (see Theorem 10.3.1 in Lattice Path Enumeration [Krattenthaler, 2015]).

For the restricted paths, we can use André’s reflection principle [Krattenthaler, 2015] to count them. According to the reflection principle, we can change any paths that touches or crosses the line so that horizontal steps become vertical steps and vice versa (see Figure 1) which reflects the endpoint of these paths to  $(dist + i, i - 1)$ . Hence, the total number of paths, using above binomial coefficient for the legal and illegal paths, is  $\binom{dist-1+2i}{i} - \binom{dist-1+2i}{i-1} = \frac{dist}{i} \binom{dist-1+2i}{i-1}$  which is  $C(dist - 1 + i, i)$ .  $\square$

The number of the robot’s paths follows immediately.

*Proof of Theorem 1.* By the correspondence between the robot’s paths and the lattice paths (see Section 3.5), Lemma 3 directly implies the claim.  $\square$

### 3.7 Probability of Catching the Adversary

Finally, having the number of paths, we can immediately state the probability that a robot in segment 1 reaches segment  $j$  in at most  $t$  steps given Bernoulli parameter  $p$ .

**Theorem 2.** *The probability of an omnidirectional robot catching an adversary in segment  $j$  of a perimeter of size  $d$  is*

$$Pr(1, j, t) = \sum_{i=0}^{\lfloor \frac{t-d+j-1}{2} \rfloor} C(d-j+i, i) \cdot p^{d+i-j+1} \cdot (1-p)^i + \sum_{i=0}^{\lfloor \frac{t-j+1}{2} \rfloor} C(j-2+i, i) \cdot p^i \cdot (1-p)^{i+j-1}.$$

*Proof.* The probability function follows by combining the statements. Firstly, Observation 1 and Lemma 2 imply that the results hold for both directions. As stated in Section 3.4, by the Observations 2 and 3 we can sum over all prefixes to get all paths. Finally, Theorem 1 gives the number of paths and Definition 2 gives the probability of a single path.  $\square$

## 4 Remaining Settings

In the previous section we presented the general modelling technique and how to count the number of paths. Similarly, the other three cases can be determined by counting lattice paths of different features. Due to the space limitations, we focus on the important differences. We present them for the cases of the directional movement in the circle and the omnidirectional movement on the line while omitting the directional movement on the line which requires no further ideas and approaches on top of the other three cases.

### 4.1 Directional Movement in the Circle

For the directional movement we have a path probability of  $P_W = p^X \cdot (1-p)^Y$  where, in this case,  $X$  is the number of steps and  $Y$  is the number of turns. Like before, the robot needs  $dist$  moves to reach  $j$ . However, in comparison to the previous case,  $X$  and  $Y$  do not directly correspond to the left and right steps which means we need a different approach of counting the paths. Moreover, we separate the additional moves into additional steps, similar to the previous case, steps besides those required to reach the target segment, and two types of changes in direction called turn and spin.

**Definition 5 (Turn / Spin).** *A turn is a change of direction followed by a step into the new direction and a spin are two consecutive changes of direction.*

Overall, we have the following result for the number of paths for one combination which we prove in this section.

**Lemma 4.** *The number of paths from  $(0, 0)$  to  $(dist - 1 + k, k)$  with  $2i$  turns and spins, from which  $2m$  are turns,  $2k$  additional steps, and not crossing line  $y = x - dist$  is*

$$\binom{dist+2k+i-m-1}{i-m} \cdot \left( \binom{dist-1+k}{m} - \binom{dist+k}{m} \binom{k-1}{m} \right).$$

In more detail, similar to the omnidirectional case we have  $t - dist$  time for the robot to perform movements other than reaching the target. The time is divided between additional steps, turns requiring  $\tau + 1$  (the change of direction and the following step) and spins requiring just  $\tau$ . Like before, all of them come in pairs, since moving or turning away from the target has to be reversed at some point. The division into turns and spins is for technical reasons. In the lattice, as it is presented in Section 3.5, we can represent turns since they require an additional step after it. In comparison, a spin does not change the segment of the robot and cannot be integrated into the path. Nevertheless, spins can be performed at any time during a path. This allows us to split the counting into two steps. The first step is to count all paths considering only additional steps and turns and the second step is to multiply this by the number of possible spins. Turns can be represented as a feature of the lattice paths since they appear as either a *north east turn* or a *east north turn* (see Figure 1).

**Definition 6** (NE-turn / EN-turn). *A point is a north east turn (NE-turn) if it is the end point of a vertical step and the starting point of a horizontal step. Similarly, a point is a east north turn (EN-turn) if it is the end point of a horizontal step and the starting point of a vertical step.*

We count the number of turn pairs  $m$  by the number of NE-turns since those represent the total number of turns.

**Lemma 5.** *A path with  $2m$  turns is a path with  $m$  NE-turns.*

*Proof.* With two exceptions every turn of the robot shows up as a NE-turn or an EN-turn (see the thick grey path in Figure 1 with two turns of either type). The exceptions are walking vertically from  $(0, 0)$  which is a turn of direction (equivalent to EN-turn) and walking vertically into the target segment. The latter of walking into the target segment is not possible in our setting (compare Observation 2). Consequentially, the paths with  $m$  NE-turns represent all path with  $2m$  turns.  $\square$

This lemma and further results from combinatorics gives us Lemma 4; only sketched here due to space limitations.

*Proof Sketch of Lemma 4.* The term is comprised of two factors from the two steps. Firstly, counting all paths with  $m$  NE-turns can be done using known lattice path results (see Equation 10.120 in Section 10.14 of Krattenthaler [2015]). Similar to the previous case, illegal paths can be deducted using the reflection principal. Together, by Lemma 5, this gives the number of legal paths given the number of steps and turns.

The spins can happen on any part of the path. Hence, distributing the  $i - m$  spins over the paths of length  $dist - 1 + 2k + 2m + 1$  is like distributing indistinguishable balls (spins) into distinguishable boxes (the points on the lattice path). This number is an elementary result in enumerative combinatorics (see e.g. Stanley [2011]).  $\square$

Finally, the probabilities are simply a combination of  $i, m$  and  $k$  similar to Theorem 2 (omitted due to space constraints).

## 4.2 Omnidirectional Movement On the Line

In comparison to the circle, the line’s defined ends break the previous symmetry. This affects the paths’ probabilities since the robot turns around with probability 1. In general, the movement is now split into  $X$  right steps with probability  $(1 - p)$ ,  $Y$  left steps with probability  $p$  and  $Z$  steps turning around in the end segments; implying the probability measure  $P_W(\omega) = p^Y \cdot (1 - p)^X \cdot 1^Z$ . This is not reflected in the lattice described above since in the circle the robot had no restriction, except the available steps, for walking away. Hence, the crucial part of this setting is to reflect the paths’ restrictions by restrictions of the lattice paths. This can be done by introducing a second line  $y = x + dist(j', 1)$  which represents the end of the line (see Figure 1). However, since reaching the end of the line affects the probability, we have to consider not only paths of a certain freedom of movement but how often a path reaches the end of the line. While this number depends on the number of available steps, in general, we have to determine the number of paths given that the end is reached  $k$  times, i.e. the number of lattice path that touch the introduced line  $k$  times. We enable this by adapting a bijection used by Spivey [2012]. We describe a bijection between the lattice paths that touch the line  $k$  times and the lattice paths that touches the line once.

**Lemma 6.** *There is an explicit bijection between paths from  $(0, 0)$  to  $(a, b)$  touching the line  $y = x + s$  exactly  $k$  times and paths from  $(0, 0)$  to  $(a, b - k)$  touching the line exactly once.*

*Proof.* We can transform a path from the first group to a path in the second group by taking any path from the first group and removing, beginning with the last touch, any vertical step that ends on the line until only one touch is left.

For the reverse, we take any path from the second group. We start from the last point where the path touches or intersects  $y = x + s - 1$  and add into the path one vertical step. We repeat this until the path ends in  $(a, b)$ . This procedure will always produce a path in the first group since at least one touch is guaranteed, by the one touch of line  $y = x + s$ , and shifting the path up by one vertical step results in an intersection of the path and  $y = x + s - 1$ . By always choosing the last touch or intersection this procedure reverses the mapping.  $\square$

We can directly use this result to count the relevant paths.

**Lemma 7.** *The number of paths from  $(0, 0)$  to  $(a, b)$  touching the line  $y = x + s$  exactly  $k$  times is*

$$\sum_{m=0}^{b-k-s} | |(0, 0) \rightarrow (m, m + s)| \cdot |(m, m + s) \rightarrow (a, b - k)| |,$$

where  $| (u, v) \rightarrow (x, y) |$  denotes the number of paths from  $(u, v)$  to  $(x, y)$ .

*Proof Sketch.* By Lemma 6 any relevant path can be counted with one of the sets of paths from the origin to one point on the line and ending in  $(a, b - k)$ . We can count one of these sets by multiplying the numbers of paths to the point on the line and the number of paths from the point on the line to the endpoint. Finally, summing up over all possible points on the line gives the statement.  $\square$

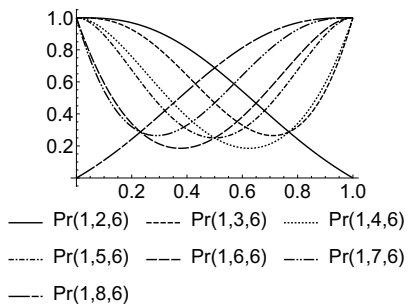


Figure 2: The probabilities of the robot reaching the segments for  $p$  ranging from 0 to 1 in a circle with 8 segments ( $d = 8$ ) and a penetration time of 6 ( $t = 6$ ).

The explicit number of paths can be obtained by counting the number of paths between two lines. This can be obtained using standard lattice path results (e.g. Theorem 10.3.4 in *Lattice Path Enumeration* [Krattenthaler, 2015]). Finally, again, combining these results yields the probability functions; omitted due to space limitations.

## 5 Examples

In this section we illustrate that our approach allows to gain further insight in the relationship between the optimal  $p$  and different values of  $d$  and  $t$ . Moreover, we illustrate the runtime as well as the removed runtime.

Firstly, the analytic probability functions allow us to plot the curves. Figure 2 illustrates the 7 probabilities of reaching the segments 2–8 from segment 1 in a circle with 8 segments and a penetration time of 6. Two different groups of segments are apparent from the figure. The first group are segments 2 and 8 where the penetration time does not allow the robot to reach the segment from both directions. Hence, the curves decrease or increase for reaching segment 2 clockwise or segment 8 anticlockwise, respectively. The other segments are reachable from both directions and make the second group. The graph shows three local maxima on the lower envelope, one in the centre at  $p = 0.5$  and two equidistant to 0.5 near 0.225 and 0.775. The general observations of this example appear to hold for all values of  $d$  and  $t$ . We conjecture, based on numerous combinations, that the grouping and the monotonic behaviour is the same for the different values.

For our analysis, we implemented an algorithm similar to FINDP of Agmon et al. [2011] in Python using Mathematica for efficient and accurate root-finding. The algorithm observes their results that the optimal value is a local maximum or an intersection of two functions on the lower envelope of the probabilities and that robots need to be synchronised and placed equidistant for optimality. Finally, the mathematically undefined edge cases of the functions above are covered by taking the limit of the functions which is always defined.

We observe that the highest possible probability of catching the adversary is 0.5 and the probability decreases exponentially for increasing  $d$  and increases for increasing  $t$ . This is in line with previous research which is why, due to space limitations, we have omitted further illustrating figures. As

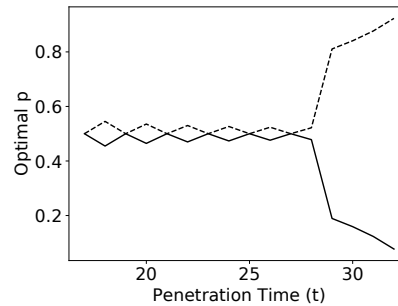


Figure 3: The optimal  $p$  for a circle of size  $d = 34$  for all possible penetration times  $t$  ( $17 \leq t \leq 32$ ). The solid and the dashed line show the one or two optimal  $p$  values.

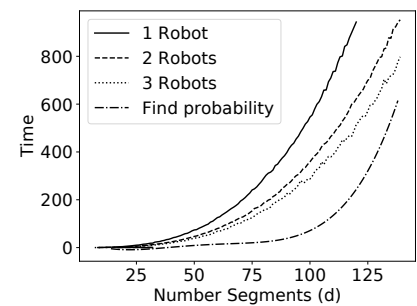


Figure 4: The runtime for finding an optimal  $p$  for 1, 2 and 3 robots (2nd step) and the additional Markov matrix approach runtime (1st step, finding the probability functions).

expected, the calculation time increases polynomially with  $d$ . However, we save the equally polynomial time of finding the probability functions which has to be added on top of finding the optimal  $p$  for previous approaches (see Figure 4).

For further insights, Figure 3 visualises a general pattern the optimal  $p$  exhibits for all  $d$  and increasing  $t$ . An analysis of a larger range of  $d$  values shows that for every  $d$  the optimal  $p$  tends towards 0.5 until  $d - 2 - \lfloor \frac{d}{11} \rfloor$  for increasing  $t$ . For smaller  $t$  values, every other optimal  $p$ , beginning with  $\lfloor \frac{d}{2} \rfloor$ , is 0.5. This suggests that for smaller penetration times the randomness is important, whereas for large  $t$  the best value tends towards the robot mostly walking in one direction.

## 6 Conclusions

We use lattice path techniques to determine the number of possible paths and the probability of catching the adversary for optimal random multi-robot adversarial patrolling strategies on a perimeter/fence with two different movement patterns. The probabilities were previously determined using Markov chain based polynomial-time black-box algorithms [Agmon et al., 2011]. Moreover, we illustrate an underlying structure of the probability functions and the change of the optimal parameter depending on the penetration time.

The techniques and results invite future work by showing how calculations can be simplified and further insight can be gained. The techniques have the potential to simplify the calculation in all similar scenarios [Sless et al., 2014; Sless Lin et al., 2019]. Furthermore, it would be interesting to apply the techniques to other structures and longer histories. Finally, more specific to the setting in this work, while high degree polynomials allow no general algebraic solution [Neri, 2016], it would be interesting to analyse further patterns like monotonicity, as well as trying to reduce the system of equations via the apparent structure of the optimal values.

## Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) doctoral training grant EP/M508147/1. Moreover, we acknowledge the use of the IRIDIS High Performance Computing Facility and associated support services at the University of Southampton.

## References

- [Agmon *et al.*, 2008] Noa Agmon, Vladimir Sadov, Gal A. Kaminka, and Sarit Kraus. The Impact of Adversarial Knowledge on Adversarial Planning in Perimeter Patrol. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 55–62, 2008.
- [Agmon *et al.*, 2011] Noa Agmon, Gal A. Kaminka, and Sarit Kraus. Multi-Robot Adversarial Patrolling: Facing a Full-Knowledge Opponent. *Journal Of Artificial Intelligence Research*, 42:887–916, 2011.
- [An *et al.*, 2017] Bo An, Milind Tambe, and Arunesh Sinha. Stackelberg Security Games (SSG) Basics and Application Overview. In Ali E. Abbas, Milind Tambe, and Detlof von Winterfeldt, editors, *Improving Homeland Security Decisions*, pages 485–507. Cambridge University Press, Cambridge, 2017.
- [Bailey, 1996] D. F. Bailey. Counting Arrangements of 1’s and -1’s. *Mathematics Magazine*, 69:128–131, 1996.
- [Basilico *et al.*, 2009] Nicola Basilico, Nicola Gatti, Thomas Rossi, Sofia Ceppi, and Francesco Amigoni. Extending Algorithms for Mobile Robot Patrolling in the Presence of Adversaries to More Realistic Settings. In *Proceedings - 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2009*, volume 2, pages 557–564, 2009.
- [Basilico *et al.*, 2012] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184-185:78–123, 2012.
- [Basilico *et al.*, 2017] Nicola Basilico, Giuseppe De Nittis, and Nicola Gatti. Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence*, 246:220–257, 2017.
- [Chevaleyre, 2004] Yann Chevaleyre. Theoretical Analysis of the Multi-agent Patrolling Problem. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004*, pages 302–308, 2004.
- [Clempner, 2018] Julio B. Clempner. A continuous-time Markov Stackelberg security game approach for reasoning about real patrol strategies. *International Journal of Control*, 91(11):2494–2510, nov 2018.
- [Elmaliach *et al.*, 2008] Yehuda Elmaliach, Asaf Shiloni, and Gal A. Kaminka. A Realistic Model of Frequency-Based Multi-Robot Polyline Patrolling. *AAMAS 2008*, 1:63–70, 2008.
- [Elmaliach *et al.*, 2009] Yehuda Elmaliach, Noa Agmon, and Gal A. Kaminka. Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence*, 57(3):293–320, 2009.
- [Huang *et al.*, 2019] Li Huang, MengChu Zhou, Kuangrong Hao, and Edwin Hou. A Survey of Multi-robot Regular and Adversarial Patrolling. *IEEE/CAA Journal of Automatica Sinica*, 6(4):894–903, jul 2019.
- [Jain *et al.*, 2013] Manish Jain, Bo An, and Milind Tambe. Security Games Applied to Real-World: Research Contributions and Challenges. In Sushil Jajodia, Anup K. Ghosh, V.S. Subrahmanian, Vipin Swarup, Cliff Wang, and X. Sean Wang, editors, *Moving Target Defense II*, pages 15–39. Springer, New York, NY, New York, NY, USA, 2013.
- [Krattenthaler, 2015] Christian Krattenthaler. Lattice Path Enumeration. In Miklós Bóna, editor, *Handbook of Enumerative Combinatorics*, Discrete Mathematics and Its Applications, chapter 10, pages 589–678. CRC Press, Boca Raton-London-New York, 1 edition, 2015.
- [Neri, 2016] Ferrante Neri. *Linear Algebra for Computational Sciences and Engineering*. Springer International Publishing, Cham, 2016.
- [Oliva *et al.*, 2019] Gabriele Oliva, Roberto Setola, and Marco Tesei. A Stackelberg Game-Theoretical Approach to Maritime Counter-Piracy. *IEEE Systems Journal*, 13(1):982–993, mar 2019.
- [Sak *et al.*, 2008] Tiago Sak, Jacques Wainer, and Siome Klein Goldenstein. Probabilistic Multiagent Patrolling. In Gerson Zaverucha and Augusto Loureiro da Costa, editors, *Advances in Artificial Intelligence - SBIA 2008*, pages 124–133, oct 2008.
- [Sea *et al.*, 2018] Vourchteang Sea, Ayumi Sugiyama, and Toshiharu Sugawara. Frequency-Based Multi-agent Patrolling Model and Its Area Partitioning Solution Method for Balanced Workload. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 530–545, Cham, jun 2018.
- [Sless *et al.*, 2014] Efrat Sless, Noa Agmon, and Sarit Kraus. Multi-Robot Adversarial Patrolling: Facing Coordinated Attacks. In *AAMAS 2014*, pages 1093–1100, 2014.
- [Sless Lin *et al.*, 2019] Efrat Sless Lin, Noa Agmon, and Sarit Kraus. Multi-Robot Adversarial Patrolling: Handling Sequential Attacks. *Artificial Intelligence*, 274:1–25, 2019.
- [Spivey, 2012] Michael Z. Spivey. Enumerating Lattice Paths Touching or Crossing the Diagonal at a Given Number of Lattice Points. *Electronic Journal of Combinatorics*, 19(3):1–6, 2012.
- [Stanley, 2011] Richard P. Stanley. *Enumerative Combinatorics, Volume 1*. Cambridge University Press, New York, NY, USA, 2 edition, 2011.
- [Stanley, 2015] Richard P. Stanley. *Catalan Numbers*. Cambridge University Press, Cambridge, 2015.
- [Talmor and Agmon, 2017] Noga Talmor and Noa Agmon. On the Power and Limitations of Deception in Multi-Robot Adversarial Patrolling. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 430–436, 2017.