Non-Uniform Policies for Multi-Robot Asymmetric Perimeter Patrol in Adversarial Domains

Yaniv Oshrat
oshblo@zahav.net.il
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel

Noa Agmon
agmon@cs.biu.ac.il
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel

Sarit Kraus
sarit@cs.biu.ac.il
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel

ABSTRACT

A patrol of robot teams, where the robots are required to repeatedly visit a target area, is a useful tool in detecting an adversary trying to penetrate. In this work we examine the *Closed Perimeter Patrol* problem, in which the robots travel along a closed perimeter and the adversary is aware of the robots' patrol policy. The goal is to maximize the probability of penetration detection. Previous work dealt with symmetric tracks, in which all parts of the track have similar properties, and suggested non-deterministic patrol schemes, characterized by a uniform policy along the entire area. We consider more realistic scenarios of *asymmetric tracks*, with various parts of the track having different properties, and suggest a patrol policy with a non-uniform policy along different points of the track. We compare the achievements of both models and show the advantage of the non-uniform model. We further explore methods to efficiently calculate the attributes needed to maximize the probability of penetration detection and compare their implementation in various scenarios.

KEYWORDS

Multi-Robot Systems; Adversarial Patrol

ACM Reference Format:

Yaniv Oshrat, Noa Agmon, and Sarit Kraus. 2020. Non-Uniform Policies for Multi-Robot Asymmetric Perimeter Patrol in Adversarial Domains. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

The problem of multi-robot patrol has been widely examined during the past few years [5, 6, 20, 22, 30, 35]. In this problem, a team of robots is required to continuously travel along a path, in our case a closed perimeter, while monitoring it for the purpose of detecting changes in state. Many works in this area have dealt with assuring optimization of frequency criteria [16, 18]. However, we follow several studies that consider the problem of **multi-robot patrol in adversarial environments**: There is an adversary that intends to penetrate this perimeter without being detected. We assume that the adversary has full knowledge of the patrolling robots (their capabilities, their locations and patrolling behavior), and would take the optimal steps, in his view, in order to penetrate successfully (we elaborate on this assumption in Section 6.4). Our goal is to set up the patrol in a way that will minimize the adversary's probability of success. Naturally, this problem is relevant to many practical domains, including security applications.

Previous studies (e.g. [3]) focused on the problem of patrolling along a symmetric perimeter, i.e. all parts of the perimeter have identical characteristics. The solution to this kind of track was a uniform probability policy which yielded satisfactory results. Unfortunately, in many realistic cases there are differences between parts of the perimeter fence and the environment. Specifically, there are certain places the adversary can penetrate more easily than others. The performance of the uniform probability policy in such cases is much worse, and the quality of patrol is low. In this work we suggest using a *non-uniform probability policy* in asymmetric tracks. Since the non-uniform model presents greater complexity and requires changes in the calculation of the parameters and in the deployment of the robots, we consider these aspects and suggest ways to efficiently calculate a patrolling policy for such cases. Finally, we show that the quality of patrol using this policy is significantly higher than in the former policy.

2 RELATED WORK

The problem of patrolling efficiently along a line [25] or in an area [5, 15, 17, 23] has been studied in several contexts. One approach is a frequency-based patrol, i.e. patrol that tries to maximize the visit frequency of the robots in the various parts of the arena [7, 18], (without regard for an adversarial presence, or disregarding its behavior). The track may be homogeneous, or there might be specific points that should be monitored more closely than the rest of the arena due to special importance concerning them [9, 16, 33]. Various robot movement models are discussed: The movement might be deterministic along preset cyclic paths, or randomized (continuously executing random steps). There are several possible criteria to evaluate the quality of the patrol – the uniformity of the patrol (minimal variance between frequencies), the average frequency and the under-bounded frequency (the minimal frequency in which any target is visited, also described as "idleness" [16]).

A second approach focuses on **adversarial patrol** schemes [2, 12]. This approach, also followed in this paper, assumes an adversary that wishes to penetrate a line, either an open polyline [1] or a closed perimeter [3] (there are significant differences in the patrol characteristics between the two cases [2]). It is assumed that the adversary knows the properties of the patrol and thus chooses its strategy and aims its penetration attempt at the patrol's weakest points. In this context, an efficient patrol would be one that makes it harder for the adversary to succeed in his penetration attempt despite full knowledge about the patrol. Implementing this

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

approach with stochastic movement may prevent the adversary from assuring the success of the penetration by choosing a smart penetration point. As mentioned before, these papers discuss only symmetric tracks, and we would like to approach the more common situation of asymmetric tracks.

In a different approach, elaborated in [8, 29] regarding both fence and perimeter patrol, both parties (the attacker and the defender) choose their strategies simultaneously, each of them without having knowledge about the choice of its opponent. There are interesting results regarding properties of the tracks and the possible consequences, but it is assumed that the tracks are symmetric (i.e. all of the parts of a track have identical characteristics), and the possibility of different parts having different properties is not discussed.

Another approach to adversarial patrolling is the game theoretic approach based on stochastic Stackelberg game models. The anchor of this method is defender-attacker Stackelberg games: In the first step the defender (leader) commits to a randomized security policy, and then the attacker (follower) uses surveillance to learn about the policy before launching its attack. The application might be either attack-timing indifferent [11] or it may have temporal preferences incorporated using exponential discounting [36, 37]. Bounded rational attackers in such settings was also studied [4, 27, 31].

The application of security-games to patrolling was examined in a series of innovative papers [10, 11, 13] that studied the problem of single and multi-robot patrols in a graph environment. They provide algorithms for determining the smallest number of robots needed to patrol a given environment, and the optimal patrolling strategies along several coordination dimensions. Our 1-D environment is a very common simplified case of their environment that allows us to consider synchronized robots' behavior yielding solutions for problems that are neither bounded by the number of robots nor by the size of the environment.

Additional approaches and comparison of different methods of patrol are described in [32]. A recent comprehensive survey of various patrol models can be found in [21].

3 CONCEPTS AND NOTATIONS

We are given a closed perimeter track that should be monitored by k identical robots. The track is divided into N segments, $S = \{seg_1, \ldots, seg_N\}$, and each robot travels through one segment in one time-cycle. The segments are not required to be of the same length or the same orientation: Some of them may be longer than others, and might contain obstructions, turns or other special characteristics, according to the conditions of the target area. However, all segments are set with proper length and robot velocity such that a robot traverses each segment in one time-cycle. We presume that robots travel on a predesignated track and their movement is not interrupted by passersby.

The robots have directionality associated with their movement. In each time-cycle, a robot has to decide where it should go - to either continue in its current direction or to turn around and go in the opposite direction. Turning around might be costly, i.e. if the robot decides to change its direction, it takes τ time-units to perform the turn and start the patrol in the opposite direction. In this paper, for simplicity, we demonstrate the movement with $\tau = 0$ (a realistic scenario for rail-mounted robots, for example), but the model can be applied for any $\tau \in \mathbb{N}$. Other movement models, including ones with $\tau > 0$, are presented in [2], along with the means to convert calculations from one model to another.

The patrol algorithm of the robots is characterized by a probability, i.e., in each step a robot continues in its current direction with some probability p, and turns around with probability $1 - p$. There are several methods for assigning the probability value to a specific step and we discuss them below. **Finding the optimal value of p for each robot step is the essence of the adversarial patrol problem**.

The adversary has to decide its preferred penetration set-up before time-cycle 0, i.e. which segment is going to be penetrated based on the location of the robot(s). It may take several time-cycles for this set-up to form, and the adversary will wait for it to arrive. At time-cycle 0 the adversary would already be in place, and in time-cycle 1 the penetration begins. We assume that the time it takes the adversary to penetrate, *penetration time*, is not instantaneous: For each $seg_i \in S$, penetration of seg_i lasts $t_i > 0$ time-cycles ($t_i \in \mathbb{N}$), from time-cycle 1 to time-cycle t_i.

The robots succeed in their mission if (at least) one of them traverses a segment seg_i while the adversary is in this segment. If t_i time-cycles have passed since the adversary began its penetration and none of the robots have traversed this segment, then the penetration is classified as successful and the robots fail. The adversary does not have another chance, nor is it allowed to stop the penetration attempt once it has commenced and then retry later.

We define the Probability of Penetration Detection (PPD) concept as follows: Let seg_i be a discrete segment of a track which is patrolled by k robots ($k \geq 1$). The Probability of Penetration Detection in segment seg_i, denoted as PPD_i, is the probability that an adversary going through seg_i is detected by some robot passing through seg_i. In other words, PPD_i is the probability that a patrol path of any robot will pass through segment seg_i during the time that a penetrator is passing through that segment. Note that PPD_i is a function of the probability values of the track. MinPPD of a track is the minimal PPD_i, $i \in \{1, \ldots, N\}$.

In this work we concentrate on assigning optimal probability values to track segments in order to maximize the minimal PPD_i, $\forall seg_i \in S$, namely MaxMinPPD of the track. With these values, the adversary, although knowing all of the patrol parameters and choosing the weakest point, i.e. the segment seg_i with $\mathsf{PPD}_i \leq \mathsf{PPD}_j \forall j \in \{1, \ldots, N\}$ as its penetration attempt arena, would have the minimal probability to succeed in his penetration attempt.

4 TYPES OF TRACKS

The problem of adversarial patrol in *symmetric tracks* – tracks where all of the segments have a uniform penetration time value, i.e. $t_i = t_j\ \forall i, j \in \{1, \ldots, d\}$ – was analyzed in [1, 2]. It was shown that *synchronous movement* of all of the robots along the track would yield the best results, namely the highest MinPPD. Synchronous movement in this sense means that the distance between every two consecutive robots is uniform, $d = N/k$ (d is assumed to be an integer), and the robots move in the same direction and at the same speed (i.e. when it is time to turn around, all of the robots do so simultaneously). The track is divided into k *sectors*, each of

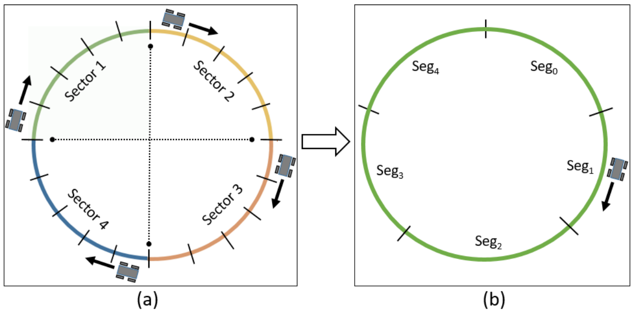

Figure 1: (a) A symmetric track with $N = 20$ segments and $k = 4$ robots. (b) A track with $N = 5$ segments and a single robot. Analysis can be done on this track and yields the optimal p value for the longer and more complicated track of (a).

which contains d consecutive segments. In symmetric tracks we note the uniform penetration time value t (without index). Fig. 1a describes a symmetric track with $N = 20, k = 4, t = 3$. This track has 4 sectors, each of them containing $d = 5$ segments. At any moment, there is exactly one robot in each sector. Furthermore, it was shown in [2] that due to symmetry between all of the sectors in the track, analysis can be done considering a closed perimeter track with one robot (Fig. 1b), i.e. in order to calculate the optimal patrol algorithm, it is sufficient to consider only one section of d segments, and not the entire perimeter of N segments. Finding the optimal p value for the segments of this single sector track would also yield the optimal p value for every closed track with sectors of length d, no matter what $k \in \mathbb{N}$ is.

It should be noted that in non-deterministic closed perimeter patrol with a given d value, the range of interesting t_i values is $\lfloor d/2 \rfloor \le t_i < d$. Lower t_i values mean that the adversary can find a segment to penetrate with a probability 0 of being detected. Higher values of t_i mean that deterministic movement along the track without turning would detect the adversary with a probability of 1.

In many real world scenarios we encounter *asymmetric tracks* – tracks with one or more segments which have a different penetration time value, $t_{irreg} < t$, compared to the other segments. This case is common where, for various reasons (lower or weaker fence, bad visibility by guards, assisting terrain conditions etc.), the adversary needs less time to penetrate these segments. Naturally, from the adversary's point of view, these segments present an opportunity to increase the probability of success in penetration, hence they require special attention from the defender's point of view.

5 DISADVANTAGE OF UNIFORM MODEL IN ASYMMETRIC TRACK

The use of a uniform probability policy in an asymmetric track is possible, but is bound to yield weak results, i.e. significantly lower MinPPD values compared to a symmetric track. We hereby prove this fact.

Penetration configuration in a single robot track is defined as the tuple $< seg_{adv}, seg_{rob}, dir_{rob} >$, where seg_{adv} indicates the segment through which the adversary is going to penetrate, seg_{rob} is the segment in which the robot resides at time-cycle 0 and dir_{rob} is the direction of the robot at time-cycle 0. For example, the penetration configuration $< seg_1, seg_3, clockwise >$ describes the situation in which the adversary tries to penetrate segment 1, commencing as the robot is in segment 3 moving clockwise.

In the uniform p model, every penetration configuration $conf$ has a PPD function, $\mathrm{PPD}_{conf} : [0, 1] \rightarrow [0, 1]$, where $\mathrm{PPD}_{conf}(p)$ is the probability that an adversary attempting to penetrate in penetration configuration $conf$, when the robot uses the uniform policy p, will succeed in the attempt. Note that the PPD_i of seg_i is the minimal PPD_{conf} of all of the penetration configurations whose $seg_{adv} = seg_i$.

Proposition 5.1. *If the penetration time of segment seg_i changes, then it may change the PPD functions of penetration configurations in which $seg_{adv} = seg_i$, but no other PPD functions in the track change.*

Intuitively, the proposition is correct because $\mathrm{PPD}(conf)$ is the sum of the probabilities of all of the paths of length $t_{seg_{adv}}$ or less that start at seg_{rob} as the robot is heading to dir_{rob} and reach seg_{adv}. These probabilities are defined by the p value of the track and by $t_{seg_{adv}}$ (the penetration time of the penetrated segment), but not by any other penetration time value of other segments in the track.

Theorem 5.2. *The uniform optimal p policy for an asymmetric track, with one segment (or more) having a penetration time t_{irreg}, is the same as the uniform optimal policy for a similar environment where **all** of the segments in the track have a penetration time t_{irreg}.*

Proof. Given a symmetric track T1 with N segments and a penetration time of t in all of the segments, let p be the optimal probability value for all segments as calculated by the uniform model, and the MinPPD of the track using this p value is M. Let T2 be an asymmetric track with N segments, where a single segment seg_x has a penetration time value t and the rest of the segments have penetration time values $t' > t$. Assume, towards contradiction, that the uniform model finds $p' \neq p$ that yields a MinPPD $M' > M$ in track T2 in seg_x. We now change the penetration time of all of the segments in T2 to t (making T2 identical to T1) but use p' as the probability value for all segments in T2. Note that PPD_{seg_x} has not changed and is still M' (according to Proposition 5.1). Since T2 is now symmetric, all of the segments have the same sets of penetration configurations, thus the MinPPD of the track is $M' > M$, in contradiction to the optimality of p.

The same argument also holds for the cases where there are more-than-one segments with penetration time t in T2 (that is, segments $seg_{x_1}, seg_{x_2}, \ldots, seg_{x_m}$ have penetration time values of t, and the rest of the segments have penetration time values of $t' > t$). □

As a result, using a uniform policy for an asymmetric track results in a much lower MinPPD in the track than in the symmetric track, making it significantly easier for the adversary to penetrate. For example, experimental calculations (described in Section 7 below) show that a regular (symmetric) closed perimeter with $d = 8$ and $t = 6$ would have a MinPPD = 0.477, but if a single segment in the perimeter has $t = 4$ then applying a uniform policy would result in MinPPD = 0.148.

6 NON-UNIFORM PROBABILITY POLICY

We present a non-uniform p policy for a closed perimeter. First we consider basic configurations of asymmetric tracks, and later we advance to more complicated configurations.

6.1 Single irregular segment

The basic configuration of an asymmetric track is a track with one *irregular segment* - a segment with a different (lower) penetration time. We denote the irregular segment as seg_1, and the segments to its right are segments $seg_2, \ldots, seg_d$ respectively. The d segments, from segment seg_1 to segment seg_d, form a sector we denote as *the irregular sector*. The track has $k-1$ *regular sectors* (sectors in which all of the segments have a regular penetration time t) and one *irregular sector* in which a single segment has t_{irreg} penetration time and the rest of the segments have penetration time t. Fig. 2a describes a track with $N = 20, k = 4, t = 3$ and a single segment with $t_{irreg} = 2$. There are four sectors of length $d = N/k = 5$: One of them (sector 1) is irregular and the other three sectors are regular.

As in the symmetric track case, we analyze the track as if it were built from k identical sectors, all of them irregular sectors like the one original irregular sector. In this way we preserve the synchronous movement model and, due to the symmetry of sectors, we can reduce the track to a single irregular d-segments-long sector with a single robot (Fig. 2b).

The various PPD functions are calculated with regards to an irregular sector, and our objective is to find values for the set $P = \{p_1, \ldots, p_{2d}\}$ (two values per segment: one for clockwise movement and one for counterclockwise movement) in order to maximize MinPPD. We hereby explain why it is sufficient to consider only an irregular sector.

THEOREM 6.1. *In a closed perimeter track using synchronous motion,* $\forall i \in \{1, \ldots, d\}$, $PPD^{irreg}_{seg_i} \leq PPD^{reg}_{seg_i}$.

PROOF. In synchronous motion, all of the robots visit segment seg_i in each robot's sector simultaneously. Therefore, $\text{PPD}^{reg}_i = \text{PPD}^{irreg}_i$ for all i values in which seg_i (in the irregular sector and in the regular sectors) have exactly the same properties, including penetration time t, and every robot path that detects the adversary in the former would also detect it in the latter **and vice versa**. However, this is not the case for i values in which some of the segments seg_i have penetration time values of $t_{irreg} < t$: Indeed every path that intercepts the adversary in the irregular segment $seg^{irreg}_{seg_i}$ also intercepts it in the regular segment $seg^{reg}_{seg_i}$, but there might be paths that intercept the adversary in the regular segment $seg^{reg}_{seg_i}$ yet would **not** intercept it in the irregular segment $seg^{irreg}_{seg_i}$ due to the shorter penetration time (t_{irreg}) in the irregular segment $seg^{irreg}_{seg_i}$, hence $\text{PPD}^{irreg}_{seg_i} \leq \text{PPD}^{reg}_{seg_i}$. Combining the two results, $\forall i \in \{1, \ldots, d\}, \text{PPD}^{irreg}_{seg_i} \leq \text{PPD}^{reg}_{seg_i}$. □

Since we are looking for the minimal PPD in the track, it will suffice to calculate the PPDs of the irregular sector and find their minimum. As Theorem 6.1 shows, PPDs of other segments would never be lower than this minimum.

6.2 Multiple irregular segments

The same method of analysis that was applied to a track with a single irregular segment is also adequate for the following two more complicated scenarios: (1) A track with two or more irregular sectors, but all of them are identical, and the rest of the sectors in the track are regular. The identicality is both with respect to the location of the irregular segment(s) in the sector, and with respect to the penetration time values of all of the segments (Fig. 3). (2) A track with several irregular segments, all of them in one sector, while the rest of the segments in the track are regular, i.e. there is one irregular sector in the track and the rest of the sectors are regular (Fig. 4). Theorem 6.1 can be applied in both cases. Therefore the track can be reduced to a single-sector track, and the calculation of the P values is done for this reduced track as described above.

A more complex scenario is a track with multiple irregular segments with various penetration time values, scattered along

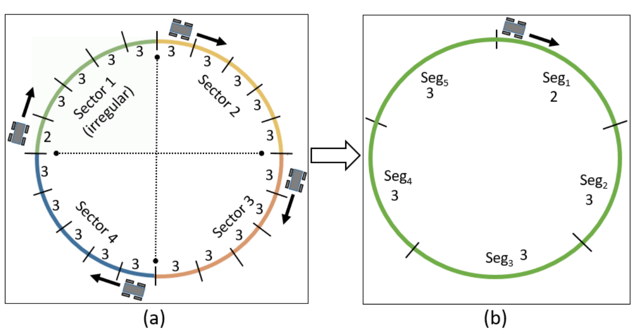

Figure 2: (a) A track with $N = 20$ segments and $k = 4$ robots. For all segments $t = 3$, except for a single irregular segment with $t_{irreg} = 2$ in the irregular sector. (b) A single-sector asymmetric track with $N = 5$ segments and a single robot. There is one irregular segment in the track – Seg_1.

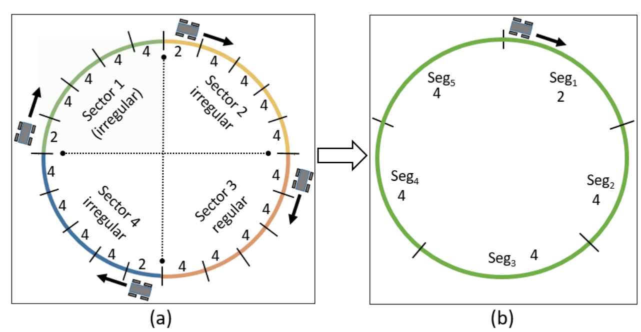

Figure 3: (a) An asymmetric track with $N = 20$ segments and $k = 4$ robots. There are 3 identical irregular sectors in the track (sector 1, sector 2 and sector 4) and one regular sector (sector 3). (b) The accordant single-sector asymmetric track with $N = 5$ segments and a single robot.

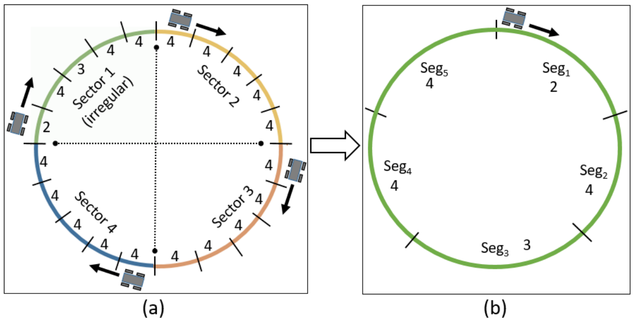

Figure 4: (a) An asymmetric track with $N = 20$ segments and $k = 4$ robots. There is one irregular sector in the track (sector 1) which contains two irregular segments (Seg_1 and Seg_3). (b) The accordant single-sector asymmetric track with $N = 5$ segments and a single robot.

the track without symmetry. In this case, for each segment $i \in \{1, \ldots, d\}$ in the sector we need to find the minimal value of penetration time of all of the sectors. Denote $t_{j,i}$ as the penetration time of segment i in sector j. We build the reduced sector and assign it penetration time values $\{\tau_1, \tau_2, \ldots, \tau_d\}$ calculated as described in Algorithm 1.

Algorithm 1 Build reduced sector

1: Divide track into k equal sectors $\{sector_1, \ldots, sector_k\}$
2: For $i \in \{1, \ldots, d\}$:
  $\tau_i = min(t_{j,i})\ \forall sector_j, j \in \{1, \ldots, k\}$

The use of the minimal t value of all segments seg_i in all of the sectors is due to the fact that if a robot detects the adversary in t time-cycles, it will also detect it in $t+1$ time-cycles. Fig. 5 presents an example of a general form of an asymmetric track without any lenient assumptions (a), and the reduced sector built upon this track by the algorithm (b). In this case, the uniform p model treats the track as if all of the segments had the minimal penetration time value

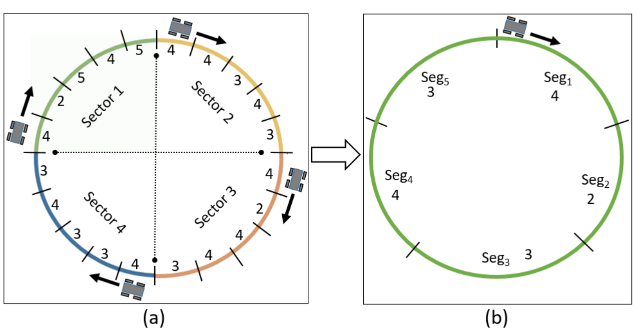

Figure 5: (a) The general case of an asymmetric track with $N = 20$ segments and $k = 4$ robots. All of the sectors are irregular and they are neither identical nor symmetric. (b) The accordant single-sector asymmetric track with $N = 5$ segments and a single robot.

(2 time-units in this example), thus finds an optimal probability $p = 0.5$ that yields MaxMinPPD of 0.25. The non-uniform p model builds a reduced sector with penetration time values $\{4, 2, 3, 4, 3\}$ (using the aforementioned Algorithm 1), thus finds an optimal $P = \{0.689, 0.518, 0.604, 0.597, 0.707, 0.484, 0.545, 0.868, 0.527, 0.587\}$ that produces MaxMinPPD of 0.349 (these results were calculated using the function-optimization method, which is elaborated on in Section 7.1).

In general, the non-uniform model achieved a higher MinPPD in all track configurations that were investigated. Fig. 6 shows the MinPPD achieved by both models in several examples of track with a single irregular segment. Fig. 7 presents examples of multiple irregular segment scenarios. The results presented in these figures and in all of the figures throughout the paper were obtained in computational experiments performed using the methods described in Section 7.

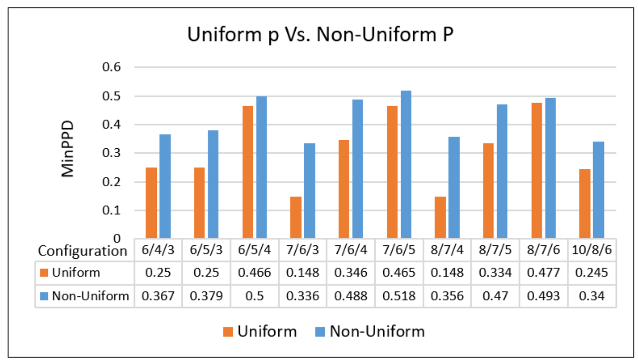

Figure 6: Comparison between the uniform p model and the variant P model in different perimeter track configurations with a single irregular segment. A configuration is denoted by its d, t and t_{irreg} values (e.g., configuration 5/4/3 means a perimeter with sectors of 5 segments each, and a penetration time of 4 time-cycles for all segments except one which has $t_{irreg} = 3$).

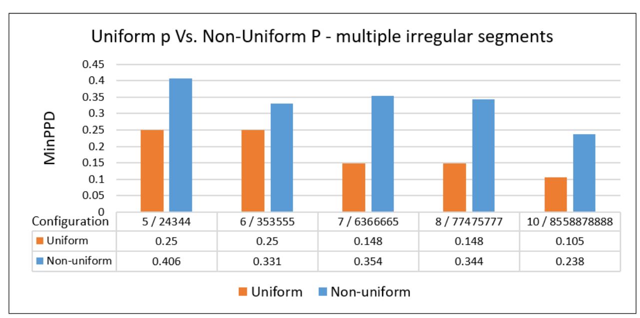

Figure 7: Comparison between the uniform p model and the variant P model in different perimeter track configurations with multiple irregular segments. Each configuration is characterized by its d and $\{t_1, \ldots, t_d\}$ values (e.g., configuration $5/\{4, 3, 4, 4, 3\}$ means a perimeter having sectors of 5 segments each, and the t values of the segments are 4,3,4,4,3).

6.3 A note on the use of the non-uniform policy

There is a distinction between the uniform p model and the non-uniform p model regarding the assignment of specific p values to segments: In the uniform model all of the segments are equal, each and every one of them is assigned the same p value, and there is no actual division into sectors in practise (only virtual division as a means of analysis). This is not the case in the non-uniform model: We have to define the sectors and number the segments for the calculation, and after obtaining the results (a set of $P = \{p_1, \ldots, p_{2d}\}$) we must assign each segment seg_i with its specific p_i, p_{i+d} values (one for clockwise movement and one for counterclockwise movement). Even when having found the optimal P values, assigning them to the segments in a different order will result in different PPD values and a lower MinPPD. All of the sectors in the track, whether regular or irregular, are assigned the same P values that were calculated using the reduced sector.

6.4 The omniscient adversary assumption

In Section 1 we assumed that our adversary has full knowledge of the patrolling robots. This assumption might look too strong, but actually it is rather realistic. The adversary does not have to monitor the exact locations of all of the patrolling robots at every moment. In order to decide which segment is optimal to penetrate, it is sufficient to know the track data (penetration times of the segments) and to calculate the patrol scheme (p values of the segments) as described above, thus finding the segment(s) with the lowest PPD in the track. Having decided the specific segment it is going to penetrate, the adversary only needs to know where the two adjacent robots (one to the left of the penetrated segment and one to its right) are located in order to initiate the penetration attempt at the right moment. Furthermore, when approaching a track patrolled by robots deployed in a synchronous movement, monitoring a single arbitrary robot of the track indicates the exact locations of all of the robots. Therefore, the assumption that the adversary knows all of the locations of the robots at any time-cycle is very realistic.

7 CALCULATING P VALUES

There are major differences between the uniform probability policy and the non-uniform probability policy regarding the optimization problems that need to be solved. In the uniform model, due to symmetry of the track (as shown in Theorem 5.2), there are d possible penetration configurations, hence we have a set of d constraints with one variable (that is p). In the non-uniform model there are $2d^2$ possible penetration configurations, since there are d possible locations for the adversary to penetrate, and d possible starting points for the robot with 2 possible starting directions each. This sums up to $2d^2$ constraints with $2d$ variables ($p_1, \ldots, p_{2d}$). As a result, methods that are practical for the uniform policy are not applicable in the non-uniform one. We first describe the methods, then discuss their adequacy in specific scenarios.

7.1 Function optimization methods

In the first group of methods, *function optimization methods*, we compute the PPD probability functions of the track, then use numeric methods to find an optimal solution for them. In order to compute the PPD functions, we deploy the method described in [2]: We build a Markov Chain G in which, for each segment seg_i in the original track, two states of G are created – one for moving clockwise and the other for moving counterclockwise. For each $1 \leq i \leq d$, if the robot reaches segment seg_i through which the adversary tries to penetrate within t_i time-cycles, then the adversary is caught. Fig. 8 demonstrates a Markov Chain of a closed perimeter track with $d = 5$ segments in the case where the adversary tries to penetrate segment seg_2.

From Markov Chain G we derive the respective stochastic matrix. As explained above, the application of the algorithm as described in [2] for the use in the non-uniform policy requires adjustments: We use the algorithm $2d$ times, one for every segment through which the adversary may penetrate in both directions, and each time we use the relevant penetration time value t_i. We then combine the sets of functions we get as a result to form a $2d \times d$ matrix of non-linear PPD functions, i.e. there are $2d^2$ non-linear functions with $2d$ variables $p_1, \ldots, p_2d$.

After building the PPD functions we need to find values of P to maximize MinPPD. The algorithms described in former works [2, 3] to calculate the value of p for MaxMinPPD were designed for the uniform p policy (one-variable functions) and are not applicable here ($2d$-variable functions). Numerical methods may be used to find an approximation to the MaxMinPPD, e.g. downhill simplex (Nelder-Mead) method [26] or sequential quadratic programming (SQP) method [14]. In our implementation we used the Nelder-Mead and the SLSQP methods of the function optimize.minimize() in the Python SciPy library for scientific computing [19, 24].

7.2 Simulation-based search methods

In the second group of methods, *simulation-based search* methods, we "guess" a possible set of P values and use simulations to estimate

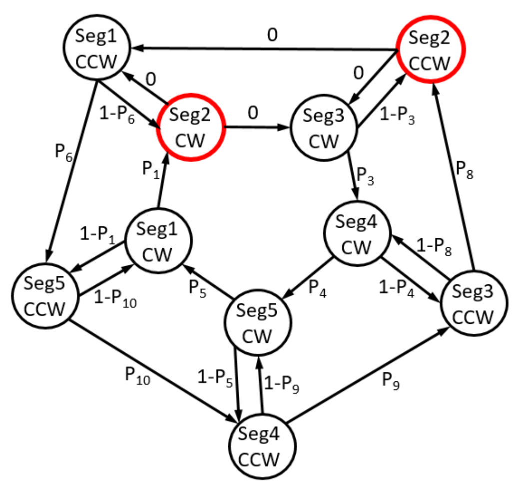

Figure 8: A Markov Chain for a perimeter track with $d = 5$, for the case of the adversary trying to penetrate segment 2. The clockwise (cw) movement states are in the inner circle and the counterclockwise (ccw) ones are in the outer circle. Note that both states of segment 2 (red) are absorbing states.

the MinPPD that these values yield. The simulation is carried out by going over all of the possible penetration configurations, running the track using random "coins" with the "guessed" P probability values, and counting the catch/fail ratio.

There are several possible ways to search the p values space and "guess" a set of values for the simulations. For a small search space, exhaustive search is practical and yields optimal results. If the space is too big to be exhausted (and in many realistic scenarios this is the case, as we elaborate in the next section), then random search algorithms are more appropriate, because in this situation finding an approximate global optimum is more important than finding a precise local optimum. We implemented a random-restart hill-climbing search, which combines iterative hill-climbing search with recurring runs using arbitrary start conditions, in order to avoid extensive search around local maxima instead of finding the global maximum [34].

7.3 Calculation methods analysis

Addressing a specific task of assigning probability values to a given track, the calculation method should be fitted to the properties of the task: (a) the number of segments in the track; (b) symmetry level of the penetration time values of the segments; and (c) the time-frame given to yield the results. For a theoretical analysis without critical time bounds, the function optimization methods are liable to produce the most accurate results. For small values of d ($d \le 8$), calculation times are acceptable (up to 24H), but for higher values the time needed to get a result soars with the complexity of the PPD functions (recall that for an asymmetric sector with d segments, $2d^2$ functions with $2d$ variables should be evaluated). Another disadvantage of these methods is the inability to adjust the calculation to a given time-frame.

The simulation-based search methods are more adequate for practical use: The complexity of a higher d grows linearly, the resolution of the search space can be adjusted, and the search can be bounded to the needed time-frame. Moreover, these methods enable a trade-off between time-performance and result-optimality: A preliminary result might be provided in a relatively short time for immediate use in the patrol, and a better result (achieved by a longer or higher-resolution search) would be supplied later to improve performance. Exhaustive search is relevant to small search spaces (e.g. search of all possible p values with a resolution of 0.001 in a uniform p model), but it is impractical to exhaust larger search spaces in this way (for example, even if we compromise on a resolution of 0.01, a non-uniform p model for an asymmetric track of 9 segments has $100^{2\cdot 9} = 10^{36}$ possible P values to consider).

We performed multiple comparative tests of more than 30 scenarios in order to establish the above characteristics of the methods. In order to demonstrate these tests, we herewith present two of the tested scenarios (Figures 9 and 10). In each of the scenarios the following five methods have been deployed:

(1) Uniform p analysis using the function optimization method (SQP).
(2) Uniform p simulation-based exhaustive search of the range $[0.4, 1.0]$ with resolution 0.001.
(3) Non-uniform p analysis using the function optimization method (SQP).
(4) Non-uniform simulation-based search using random-restart hill-climbing, stopping after 100K steps.
(5) Non-uniform simulation-based search using random-restart hill-climbing, stopping after a pre-set time.

All simulation-based methods (2, 4, 5) used 200K simulation rounds to check a single result.

Scenario 1 (Fig. 9a) is a triple-robot perimeter with $N = 21$ and a regular penetration time value $t = 6$, but there are two irregular segments along the track with t_{irreg} values of 3 (in sector 1) and 4 (in sector 2). Using the reduction scheme presented in section 4, we need to analyze a track with $d = 21/3 = 7$ segments (Fig. 9b). Fig. 9c presents the results and the time it took to calculate them (method 5 was set to stop after 3 hours of search): For this relatively short sector configuration, the function analysis methods (1 and 3) are comparatively faster and yield better results (that is, a higher MinPPD) than the simulation-based search methods. As we expected according to Theorem 5.2 above, the non-uniform model (methods 3, 4 and 5, colored in blue) produces much higher results compared to the uniform model (methods 1 and 2, colored in orange).

Scenario 2 is a dual-robot perimeter with $N = 24$ and a regular penetration time value $t = 8$, but there are 3 irregular segments along the track as demonstrated in Fig. 10a. The reduced track (Fig. 10b) has 12 segments. Fig 10c presents the results and their calculation times in the aforementioned methods (method 5 was set to stop after 24 hours of search): Both function optimization methods (methods 1, 3) failed to return a result even after 4 days of calculation due to the size of the track and the consequent complexity of analysis. The simulation-based search methods produced results: Again, as we expected, the non-uniform model (methods 4, 5, blue)

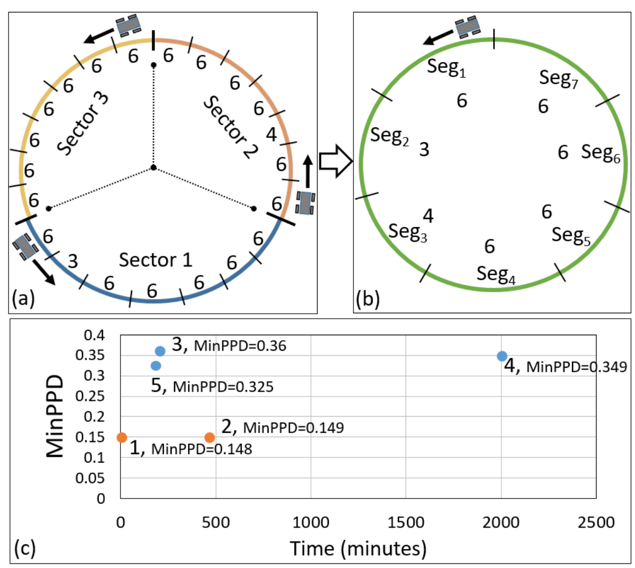

Figure 9: Scenario 1: (a) Triple-robot track with $N = 21, t = 6$ and two irregular segments. (b) The accordant single-sector reduced track. (c) MinPPD values and calculation time in the various methods: Orange – uniform p model, blue – non-uniform p model. Method 5 was set to 3H of search.

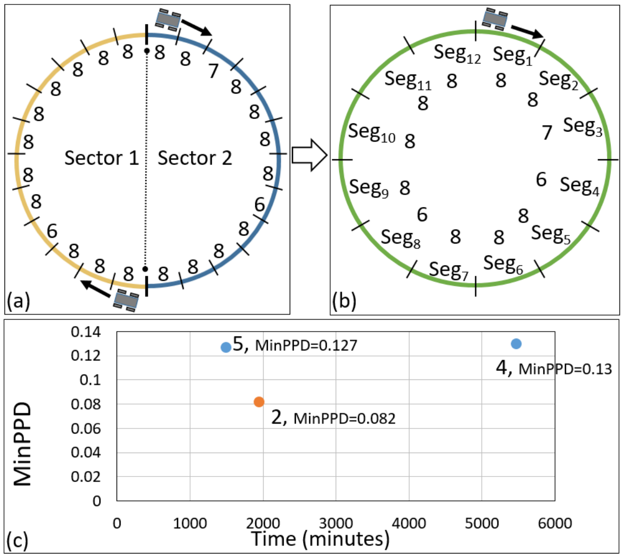

Figure 10: Scenario 2: (a) Dual-robot track with $N = 24, t = 8$ and 3 irregular segments. (b) The accordant single-sector reduced track. (c) MinPPD values and calculation time in the various methods: Orange – uniform p model, blue – non-uniform p model. Method 5 was set to 24H of search.

yielded a higher MinPPD compared to the uniform model (method 2, orange). Adjusting search parameters to the properties of the specific task allows for obtaining results in relevant time (basic results in 24H by method 5, and slightly better results , if needed, 48H later using method 4).

7.4 Practical implementation

The model presented in this paper can be implemented in real-world systems of closed perimeter fence patrol. The various properties of the task (as detailed in Section 7.3) should be considered, and rule-based methods or heuristics may be applied to fit the optimal patrol scheme to the situation (the ATAPS heuristic described in [28] is such a tool). In reality, given specific track properties and available patrol resources, the number of scenarios that should be analyzed is smaller than in theoretical analysis, due to practical limitations and cost considerations. For example, the number of robots assigned to the mission is a crucial factor, and it is determined mainly by the costs of the robots and of their deployment. We may consider and calculate p values for all the theoretically-possible numbers of robots in the track. But in practice, the available resources (i.e., the cost limitation for the patrol) will reduce the set of options to a few feasible scenarios, and only these scenarios should be analyzed and compared. Thus finding the preferred scenario can be done quickly and cheaply. Therefore, implementing a patrol using the presented non-uniform stochastic model may improve patrol quality in existing and in future applications.

8 CONCLUSION

In this paper we have presented and discussed the non-uniform p model as an improved model for adversarial patrol of asymmetric perimeter tracks. This model produces higher MinPPD values compared to the uniform p model. Though more complex to calculate by function optimization methods, the p probability values can be efficiently calculated using simulation-based methods, in adjustment to the specific circumstances of each track. The results may be rapidly applied to real-world patrol tasks, significantly improving the performance of the patrolling robots.

ACKNOWLEDGMENTS

This work was partially supported by Ministry of Science and Technology, Israel and the Japan Science and Technology Agency (JST), Japan.

REFERENCES

[1] Noa Agmon, GA Kaminka, and S Kraus. 2008. Multi-robot fence patrol in adversarial domains. In *Proceedings of the Tenth Conference on Intelligent Autonomous Systems (IAS-10)*. 193–201.

[2] Noa Agmon, Gal A Kaminka, and Sarit Kraus. 2011. Multi-robot adversarial patrolling: facing a full-knowledge opponent. *Journal of Artificial Intelligence Research* 42 (2011), 887–916.

[3] Noa Agmon, Sarit Kraus, and Gal A Kaminka. 2008. Multi-robot perimeter patrol in adversarial settings. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2339–2345.

[4] Noa Agmon, Vladimir Sadov, Gal A Kaminka, and Sarit Kraus. 2008. 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*. International Foundation for Autonomous Agents and Multiagent Systems, 55–62.

[5] Tauhidul Alam, Md Mahbubur Rahman, Leonardo Bobadilla, and Brian Rapp. 2017. Multi-vehicle patrolling with limited visibility and communication constraints. In *Military Communications Conference (MILCOM), MILCOM 2017-2017 IEEE*. IEEE, 465–470.

[6] Tauhidul Alam, Md Mahbubur Rahman, Pedro Carrillo, Leonardo Bobadilla, and Brian Rapp. 2019. Stochastic Multi-Robot Patrolling with Limited Visibility. *Journal of Intelligent & Robotic Systems* (2019), 1–19.

[7] Alessandro Almeida, Geber Ramalho, Hugo Santana, Patricia Tedesco, Talita Menezes, Vincent Corruble, and Yann Chevaleyre. 2004. Recent advances on multi-agent patrolling. In *Brazilian Symposium on Artificial Intelligence*. Springer, 474–483.

[8] Steve Alpern, Alec Morton, and Katerina Papadaki. 2011. Patrolling games. *Operations research* 59, 5 (2011), 1246–1257.

[9] Nicola Basilico and Stefano Carpin. 2018. Balancing Unpredictability and Coverage in Adversarial Patrolling Settings. In *Workshop on Algorithmic Foundations or Robotics (WAFR)*.

[10] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 57–64.

[11] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. 2012. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence* 184 (2012), 78–123.

[12] Nicola Basilico, Nicola Gatti, Thomas Rossi, Sofia Ceppi, and Francesco Amigoni. 2009. Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, Vol. 2. IEEE, 557–564.

[13] Nicola Basilico, Nicola Gatti, and Federico Villa. 2010. Asynchronous Multi-Robot Patrolling against Intrusions in Arbitrary Topologies.. In *Proc. of AAAI*.

[14] Joseph-Frédéric Bonnans, Jean Charles Gilbert, Claude Lemaréchal, and Claudia A Sagastizábal. 2006. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media.

[15] Yann Chevaleyre. 2004. Theoretical analysis of the multi-agent patrolling problem. In *Intelligent Agent Technology, 2004.(IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on*. IEEE, 302–308.

[16] Andrew Collins, Jurek Czyzowicz, Leszek Gasieniec, Adrian Kosowski, Evangelos Kranakis, Danny Krizanc, Russell Martin, and Oscar Morales Ponce. 2013. Optimal patrolling of fragmented boundaries. In *Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures*. ACM, 241–250.

[17] Francesco Maria Delle Fave, Albert Xin Jiang, Zhengyu Yin, Chao Zhang, Milind Tambe, Sarit Kraus, and John P Sullivan. 2014. Game-theoretic patrolling with

dynamic execution uncertainty and a case study on a real transit system. *Journal of Artificial Intelligence Research* 50 (2014), 321–367.
[18] Yehuda Elmaliach, Noa Agmon, and Gal A Kaminka. 2009. Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence* 57, 3-4 (2009), 293–320.
[19] Fuchang Gao and Lixing Han. 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications* 51, 1 (2012), 259–277.
[20] Benjamin Hefferan, Oliver M Cliff, and Robert Fitch. 2016. Adversarial patrolling with reactive point processes. In *Proceedings of the ARAA Australasian Conference on Robotics and Automation (ARAA, 2016)*. 39–46.
[21] Li Huang, MengChu Zhou, Kuangrong Hao, and Edwin Hou. 2019. A survey of multi-robot regular and adversarial patrolling. *IEEE/CAA Journal of Automatica Sinica* (2019).
[22] Marika Ivanová and Pavel Surynek. 2017. Area Protection in Adversarial Path-Finding Scenarios with Multiple Mobile Agents on Graphs: a theoretical and experimental study of target-allocation strategies for defense coordination. *arXiv preprint arXiv:1708.07285* (2017).
[23] Marika Ivanová, Pavel Surynek, and Katsutoshi Hirayama. 2018. Area Protection in Adversarial Path-finding Scenarios with Multiple Mobile Agents on Graphs. (2018).
[24] Dieter Kraft. 1988. *A software package for sequential quadratic programming*. Technical Report DFVLR-FB 88-28. DLR German Aerospace Center - Institute for Flight Mechanics, Cologne, Germany.
[25] Kyle Y Lin. 2019. Optimal Patrol on a Perimeter. *arXiv preprint arXiv:1905.03600* (2019).
[26] John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *The computer journal* 7, 4 (1965), 308–313.
[27] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. 2013. Analyzing the effectiveness of adversary modeling in security games. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.
[28] Yaniv Oshrat, Noa Agmon, and Sarit Kraus. 2020. Adversarial Fence Patrolling: Non-Uniform Policies for Asymmetric Environments. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence.*
[29] Katerina Papadaki, Steve Alpern, Thomas Lidbetter, and Alec Morton. 2016. Patrolling a border. *Operations Research* 64, 6 (2016), 1256–1269.
[30] Praveen Paruchuri, Jonathan P Pearce, Milind Tambe, Fernando Ordonez, and Sarit Kraus. 2007. An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 181.
[31] James Pita, Manish Jain, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. 2010. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence* 174, 15 (2010), 1142–1171.
[32] David Portugal and Rui Rocha. 2011. A survey on multi-robot patrolling algorithms. In *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 139–146.
[33] Ariel Rosenfeld and Sarit Kraus. 2017. When Security Games Hit Traffic: Optimal Traffic Enforcement Under One Sided Uncertainty.. In *IJCAI*. 3814–3822.
[34] Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.
[35] Efrat Sless, Noa Agmon, and Sarit Kraus. 2014. Multi-robot adversarial patrolling: facing coordinated attacks. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1093–1100.
[36] Yevgeniy Vorobeychik, Bo An, and Milind Tambe. 2012. Adversarial patrolling games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 1307–1308.
[37] Yevgeniy Vorobeychik, Bo An, Milind Tambe, and Satinder P Singh. 2014. Computing Solutions in Infinite-Horizon Discounted Adversarial Patrolling Games.. In *ICAPS*.