A Computational Model for Management of Wildlife Reserves in India

A

Project Report

submitted in partial fulfillment of the

requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWROK

by

Name	Roll No.
Saurabh Shanu	R102214006(SAP:500034227)

Under the guidance

Mr.Anil Kumar
Centre of Information Technology, UPES, Dehradun

Dr. Y.V. Jhala
Wildlife Institute of India, Dehradun

Prof. Qamar Qureshi
Wildlife Institute of India, Dehradun

Department of Computer Science & Engineering

Centre for Information Technology

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, UK

April – 2016

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled **"A Computational Model for Management of Wildlife Reserves in India"**in partial fulfilment of the requirements for the award of the Degree of MASTER OF TECHNOLOGY IN COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORKand submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my work carried out during a period from **January, 2016** to **April, 2016** under the supervision of **Dr. Y.V. Jhala and Prof. Qamar Qureshi** *Wildlife Institute of India, Dehradun* **and Mr.Anil Kumar** *Centre of Information Technology, UPES , Dehradun.*

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other University.

Saurabh Shanu
Roll No.: R102214006
Sap ID:500034227

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: April 11 2016

Anil Kumar
(Project Guide)

Dr. Amit Agarwal
Program Head – M.Tech(AI & ANN)
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 007 (Uttarakhand)

भारतीय वन्यजीव संस्थान
Wildlife Institute of India

Certificate

I hereby certify that the project work entitled **"A Computational Model for Management of Wildlife Reserves in India"**, undertaken by Mr. Saurabh Shanu, enrollment number: R102214006 (SAP: 500034227) in partial fulfillment of the requirements for the award of the Degree of MASTER OF TECHNOLOGY IN COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORK from the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun in collaboration with the Wildlife Institute of India, Dehradun is an authentic record of the work carried out during a period from **January, 2016** to **April, 2016.** The **project was** undertaken under the supervision of **Prof. Qamar Qureshi and myself from the** *Wildlife Institute of India, Dehradun* **and Mr.Anil Kumar** *Centre of Information Technology, UPES , Dehradun.*

Date: April 5, 2016

Yadvendradev Jhala, Ph.D.
Sr. Professor & Scientist "G"
Wildlife Institute of India
Dehradun, India 248001

पत्रपेटी सं0 18, चन्द्रबनी, देहरादून – 248001, उत्तराखण्ड, भारत
Post Box No. 18, Chandrabani, Dehradun – 248001, Uttarakhand, INDIA
ई.पी.ए.बी.एक्स : +91-135-2640111 से 2640115 फैक्स : 0135-2640117
EPABX : +91-135-2640111 to 2640115; Fax : 0135-2640117;
ई–मेल / E-mail: wii@wii.gov.in, वेब / website: www.wii.gov.in

ACKNOWLEDGEMENT

Iwish to express my deep gratitude to my guides **Dr. Y.V. Jhala and Prof. Qamar Qureshi** *Wildlife Institute of India, Dehradun* **and Mr.Anil Kumar** *Centre of Information Technology, UPES , Dehradun*, for all advice, encouragement and constant support they have given to methroughoutmy project work. This work would not have been possible without their support and valuable suggestions.

I am heartily thankful to my course coordinator, **Mr. Vishal Kaushik**, for the precise evaluation of the milestone activities during the project timeline and the qualitative and timely feedback towards the improvement of the project.

I sincerely thank to our respected **Dr. Amit Agarwal**, Program Head of the Department,for his great support in doing our project in **M. Tech Artificial Intelligence and Artificial Neural Networks** at **CIT**.

Iam also grateful to **Dr. Manish Prateek, Associate Dean** and **Dr. Kamal Bansal**Dean CoES, UPES for giving me the necessary facilities to carry out my project work successfully.

I would like to thank all my**friends** for their help and constructive criticism during my project work. Finally,I have no words to express my sincere gratitude to my**parents** who have shown me this world and for every support they have given me.

Name **Saurabh Shanu**

Roll No. **R102214006**

Sap ID **500034227**

ABSTRACT

The problem of adversarial multi-path guard dependent patrol has gained interest in recent years, mainly due to its immediate relevance to various security applications. In this problem, patrol guards are required to repeatedly visit a target area in a way that maximizes their chances of detecting an adversary trying to penetrate through the patrol path. When facing a strong adversary that knows the patrol strategy of the guards, if the guards use a deterministic patrol algorithm, then in many cases it is easy for the adversary to penetrate undetected (in fact, in some of those cases the adversary can guarantee penetration). Therefore, this project presents a non-deterministic patrol framework for the guards. Assuming that the strong adversary will take advantage of its knowledge and try to penetrate through the patrol's weakest spot, hence an optimal algorithm is one that maximizes the chances of detection in that point. We therefore present a polynomial-time algorithm for determining an optimal patrol under the Markovian strategy assumption for the guards, such that the probability of detecting the adversary in the patrol's weakest spot is maximized. We build upon this framework and describe an optimal patrol strategy for several patrol guards based on their movement abilities (directed or undirected) and sensing abilities (perfect or imperfect), and in different environment models - either patrol around a perimeter (closed polygon) or an open fence (open polyline).

In this work, we use game theory and graph theory to model and design a patrolling guard path web. We construct a graph using the patrol chaukis as vertices and the possible paths between these vertices as edges. A cost matrix is constructed to indicate the cost incurred by the patrol guard for passage between the habitat patches in the landscape, by modelling a Hawk and Dove game. A minimum spanning tree or a Hamiltonian path, depending on the start and end point is then obtained by employing Kruskal's algorithm or Travelling Salesman problem, which would suggest a feasible adversary detection path for the patrol guards within the landscape complex.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page No**

LIST OF TABLES

Table **Page No**

1.INTRODUCTION

Wildlife monitoring is essential for keeping track of animal movement patterns, habitat utilization, population demographics, snaring and poaching incidents and breakouts. This valuable information, which Wildlife ACT and forest patrol guards gather on wildlife projects, has numerous management applications, including the planning of successful introduction and removal strategies of wildlife. There are many reasons why natural resource managers need to monitor wildlife populations. A large array of methods has been developed and used to that purpose.

There are many diverse reasons why we need to monitor wildlife populations (Caughley 1977). For example, the population may be a valued game species (e.g. deer, bear, grouse) that is being managed on a sustained-yield basis. The population may be an actual or potential pest species (e.g. rodents, flocking birds, invasive/non-native species) capable of causing agricultural, property, or natural resource damage or of posing a human or livestock disease or safety hazard. We may need to assess the status of an endangered or threatened species or the progress of a recovery program for that species. We may need to determine the status of a purposeful introduction or reintroduction of a wildlife species to an area. We may be trying to define the biological diversity or 'ecological health' of an area and to monitor changes over time. We may desire to know the effects of our management actions or land-use practices or alternative activities on one or more "featured or indicator" species.

A large number of methods have been used to monitor terrestrial vertebrates (e.g. Caughley 1977; Davis 1982), although many methods have not been compared or validated with a more rigorous method of density estimation or a known population size (but see exceptions: Quy*et al.* 1993; Dodd and Murphy 1995). The methods include, for example, direct observation (day or night) of individuals, mark-recapture/resight, removal, and transects and variable plot surveys (see examples presented in Thompson *et al.* 1998). A large number of 'indirect' methods, often referred to as population or abundance indices (Thompson and Fleming 1994; Engeman and Alien 2000) or activity indices (Quy*et al.* 1993), have also been used. These methods do not rely on directly seeing or hearing the animals, but merely noting some form of 'sign' that tells us that the animals have been in the area: track stations, fecal counts, food removal, open or closed burrow-opening counts, burrow counts, runway counts, knockdown cards, snow tracks, or responses to audio calls (Engeman and Witmer 2000). These indices are based on the concept that a fixed amount of searching effort will locate a fixed proportion of the population. Furthermore, it is assumed that the index is proportional to the density and that the rate of proportionality is (relatively) constant (Caughley 1977). If the index doubles, we assume that the population has doubled. Some people might be more comfortable in calling this approach an 'activity index,' because we usually do not know the exact relationship of the index to the population density or how that relationship may change over time and space. For example, if three sets of tracks are found at a track station, we do not know if those were made by one, two, or three individual animals, but that there was three times as much 'activity' than at a track station, which had only one set of tracks (Alien *et al.* 1996). On the other hand, we typically find more sets of tracks (or, for example, more food removed from a bait station) where there is a larger population (Witmer, unpublished data on voles). Hence this approach provides a useful 'relative' index of the abundance of the animals using the area of interest. Technological developments have provided additional methods for monitoring populations such as the use of remote cameras (Bull *et at.* 1992; Glen and Dickman 2003), infrared thermal imaging (Boonstra*et al.* 1994), DNA analysis (Foran*et al.* 1997), and radio-isotope detection (Elbert *etal.* 1999).

The Monitoring system for Forest guards – Intensive Protection and Ecological Status (MSTrIPES) system consists of two components a) field based protocols for patrolling, law enforcement, recording wildlife crimes and ecological monitoring, and b) a customized software for storage, retrieval, analysis and reporting. Currently law enforcement and ecological monitoring are being done, but the information generated is ad hoc and rarely available in a format for informed decision making. The "MSTrIPES" addresses this void and is a tool for adaptive management. The system uses a holistic approach by integrating ecological insights obtained through the standardized forest guard, prey, and habitat assessment protocols (Phase I) to guide protection and management. It enables managers to assess intensity and spatial coverage of patrols in a GIS based tool. The system performs statistical computations of occupancy, precision, sample size, and assesses trends over desired time and spatial scales for forest guards, other carnivores, prey populations, human impacts, illegal activities, and law enforcement investments. MSTrIPES produces easily interpretable reports and maps that are useful for management and policy decisions. If implemented as designed, the system reduces the response time to detrimental events like poaching or habitat degradation and becomes a comprehensive tool to keep the pulse of a forest guard reserve.

During the last century the variety of species has decreased dramatically and numerous species are today classified as endangered or threatened. The main determining factor is human related activities such as forestry, farming, intensive hunting and fishing. Since the middle of the last century, a number of legal instruments concerning the use and conservation of natural resources have been applied, such as protection of individual species and their nests and restrictions on hunting and fishing, without hindering the eradication of species. One proposition for the failures of traditional legal instruments is the lack of a holistic approach in regarding ecosystem characteristics such as inter-species and habitats relations and biodiversity. It has been found that the legal instruments mainly are concerned with the rational use or protection of a certain species rather than dealing with inter-relations and the sustainability of ecosystems. E.g. the motive for legal protection is based on a definition of a sustainable population which is determined with respect to the conservation status of the targeted species rather than with aim of achieving sufficient diversity of species in the ecosystems and legal limits related to the use of wildlife populations fail to integrate ecological concepts such as biodiversity for determining such limits. Another proposition is that since ecosystems are dynamic and complex it is important that the legal system has the capacity to respond to ecological changes.

Patrolling is a basic activity of forest guards and squads. The purpose of a typical patrol is to gather information or to conduct appropriate operations. In the Contemporary Operating Environment (COE). The information gathered during patrols can be crucial to the success of the larger mission.

Patrolling can accomplish several specific objectives:

- Gathering information on the animal trails, on the terrain, or on the populace
- Reassuring or gaining the trust of a local population
- Preventing public disorder

- Deterring and disrupting insurgent or criminal activity
- Providing unit security
- Protecting key infrastructure or bases.

A patrol is organized to perform specific tasks. The patrol unit must be prepared to secure itself, navigate accurately, identify and cross danger areas, and reconnoiter the patrol objective. With the present techniques of wildlife monitoring, the wildlife conservation authorities use the patrols organized by the forest guards as an important source of data collection and prevention of poaching in the wildlife reserves but there are no defined protocols for undertaking a defined path for patrol movement in wildlife reserves which keeps many important points of protection and data acquisition out of touch of Patrol parties and thus the conservation authorities.

The problem of multi-guard patrol has gained interest in recent years (e.g., Ahmadi & Stone, 2006; Chevaleyre, 2004; Elmaliach, Agmon, &Kaminka, 2007; Paruchuri, Tambe, Ordonez, & Kraus, 2007; Amigoni, Gatti, &Ippedico, 2008), mainly due to its immediate relevance to various security applications. In the multi-guard patrol problem, guards are required to repeatedly visit a target area in order to monitor it. Many researchers have focused on a frequency-based approach, guaranteeing that some point-visit frequency criteria are met by the patrol algorithm, for example maximizing the minimal frequency or guaranteeing uniform frequency (e.g., refer to Elmaliach et al., 2007; Chevaleyre, 2004; Almeida, Ramalho, Santana, Tedesco, Menezes, Corruble, &Chevaleyr, 2004). In contrast, we advocate an approach in which the forest guards patrol in adversarial settings, where their goal is to patrol in a way that maximizes their chances of detecting an adversary trying to penetrate through the patrol path. Thus the decisions of the adversary must be taken into account. Our objective is, therefore, to develop patrol paths for the guards, such that following these paths the patrolling guards will maximize the chance of adversarial detection. The problem of adversarial planning and specifically adversarial patrolling is a wide problem, where generally no computational tractable results exist. This report presents the problem in a restrictive environment of perimeter patrol by a set of homogenous patrol guards, providing a computational tractable optimal result. As opposed to frequency-driven approaches, in adversarial settings the point-visit frequency criteria become less relevant. Consider the following scenario. We are given a cyclic fence of a length of

100 meters and 4 guards must patrol around the fence while moving at a velocity of 1m/sec. Clearly, the optimal possible frequency at each point around the fence, in terms of maximizing the minimal frequency, is 1/25, i.e., each location is visited once every 25 seconds. This optimal frequency is achieved if the guards are placed uniformly along the fence (facing the same direction) and move forward without turning around. Assume that it takes an adversary 20 seconds to penetrate the area through the fence. As the guards move in a deterministic path, an adversary knowing the patrol algorithm can guarantee penetration if they simply enter through a position that was recently visited by a patrolling guard. On the other hand, if the guards move non-deterministically, i.e., they turn around from time to time with some probability greater than 0, then the choice of penetration position becomes less trivial. Moreover, if we assume that an adversary may penetrate at any time, it motivates the use of nondeterministic patrol behavior indefinitely. We first consider the problem of patrolling around a closed polygon, i.e., a perimeter. We introduce a non-deterministic framework for patrol under a first order Markovian assumption for the patrol guards strategy, in which the guards choose their next move at random with some probability p. This probability value p characterizes the patrol algorithm. We model the system as a Markov chain (e.g., Stewart, 1994), and using this model we calculate in polynomial time the probability of penetration detection at each point along the perimeter as a function of p, i.e., it depends on the choice of patrol algorithm. Based on the functions defining the probability of penetration detection, we find an optimal patrol algorithm for the guards in the presence of a strong adversary, i.e., an adversary having full knowledge on the patrolling guards—their algorithm and current placement. In this case, the adversary uses this knowledge in order to maximize its chances of penetrating without being detected. It is therefore assumed that the adversary will penetrate through the weakest spot of the path, hence the goal of the guards is to maximize the probability of penetration detection in that weakest spot. We provide a polynomial time algorithm (polynomial in the input size, depending on the number of guards and the characteristics of the environment) for finding an optimal patrol for the guards facing this full knowledge adversary. We show that a non-deterministic patrol algorithm is advantageous, and guarantees some lower bound criteria on the performance of the guards, i.e., on their ability to block the adversary. We then use the patrol framework to consider additional environment and patrolling models. Specifically, we consider the case in which the guards are required to patrol along an open polyline (fence). We show that although this case is inherently

different from patrolling along a perimeter, the basic framework can still be used (with some changes) in order to find an optimal patrol algorithm for the guards. We investigate also different movement models of patrol guards, namely the guards can have directionality associated with their movement (and turning around could cost the system time), or they can be omnidirectional. In addition, we model various types of sensing capabilities of the guards, specifically, their sensing capabilities can be perfect or imperfect, local or long-range. In all these cases we show how the basic framework can be extended to include the various models.

This work concentrated on defining a route for patrolling to be undertaken by the patrol party in a wildlife reserve depending on various parameters which are essential to be considered by the patrol party to fulfil the aims and objectives of their movement in a reserve.

2. RELATED WORK

Systems comprising multiple patrol guardss that cooperate to patrol in some designated area have been studied in various contexts (e.g., Chevaleyre, 2004; Elmaliach, Agmon, &Kaminka, 2009). Theoretical (e.g., Chevaleyre, 2004; Elmaliach et al., 2009; Amigoni et al., 2008) and empirical (e.g., see Sak, Wainer, &Goldenstein, 2008; Almeida et al., 2004) solutions have been proposed in order to assure quality patrol. The definition of quality depends on the context. Most studies concentrate on the frequency-based patrolling, which optimizes frequency of visits throughout the designated area (e.g. refer to Elmaliach et al., 2009; Almeida et al., 2004; Chevaleyre, 2004). Efficient patrol, in this case, is a patrol guaranteeing a high frequency of visits in all parts of the area. In contrast, adversarial patrolling (addressed in this work) deals with the detection of moving adversaries who are attempting to avoid detection. Here, an efficient patrol is one that deals efficiently with intruders (e.g., see Sak et al., 2008; Basilico, Gatti, &Amigoni, 2009b; Amigoni et al., 2008). The first theoretical analysis of the frequency-based multi-patrol guards patrol problem that concentrated on frequency optimization was presented by Chevaleyre (2004). He introduced the notion of idleness, which is the duration each point in the patrolled area is not visited. In his work, he analyzed two types of multi-patrol guards patrol schemes on graphs with respect to the idleness criteria: partitioning the area into subsections, where each section is visited continuously by one patrol guard; and the cyclic scheme in which a patrol path is provided along the entire area and all patrol guards visit all parts of the area, consecutively. He proved that in the latter approach, the frequency of visiting points in the area is considerably higher. Almeida et al. (2004) offered an empirical comparison between different approaches towards patrolling with regards to the idleness criteria, and shows great advantage of the cycle based approach. Elmaliach et al. (2007, 2009) offered new frequency optimization criteria for evaluating patrol algorithms. They provide an algorithm for multi-patrol guards patrol in continuous areas that is proven to have maximal minimal frequency as well as uniform frequency, i.e., each point in the area is visited with the same highest-possible frequency. Their work is based oncreating one patrol cycle that visits all points in the area in minimal time, and the patrol guards simply travel equidistantly along this patrol path. Sak et al. (2008) considered the case of multi-agent adversarial patrol in general graphs (rather than perimeters, as in our work). They concentrated on an empirical evaluation (using a simulation) of several non-deterministic patrol algorithms that can be roughly divided into two: Those that divide the graph between the patrolling agents, and those that allow all agents to visit all parts of the graph. They considered three types of adversaries: random adversary, an adversary that always chooses to penetrate through a recently-visited node and an adversary that uses statistical methods to predict the chances that a node will be visited soon. They concluded that there is no patrol method that outperformed the others in all the domains they have checked, but the optimality depends on the graph structure. In contrast to this investigation, we provide theoretical proofs of optimality for different settings. The work of adversarial multi-patrol guards patrol was examined also by using game-theoretic approaches (e.g., see Basilico et al., 2009b; Basilico, Gatti, &Amigoni, 2009a; Pita, Jain, Ordonez, Tambe, Kraus, &Magorii-Cohen, 2009; Paruchuri, Tambe et al., 2007). Note that the work described herein can be modeled as a game theoretic problem: Given two players, the patrol guards and the adversary, with a possible set of actions by each side, determine the optimal policy of the patrol guards that will maximize their utility gained from adversarial detection. This is a zero-sum game. Since we assume a strong (full knowledge) adversarial model, we adopt the minmax approach, namely, minimizing the maximal utility of the opponent (or in this case: equivalent to maximizing the minimal probability of detection of the patrol guards). However, in our work we do not use game theoretic tools for finding the equilibrium strategy, but use tailored ad-hoc solution that finds the optimal policy for the patrol guards in polynomial time, taking into account the patrol guard's possible sensing and movement capabilities. The most closely related work by Amigoni et al. (2008) and Basilico et al. (2009b, 2009a) used a game-theoretic approach using leader-follower games for determining the optimal strategy for a single patrolling agent. They considered an environment in which a patrolling patrol guards can move between any two nodes in a graph, as opposed to the perimeter model we use. Their solution is suitable for one patrol guards in heterogeneous environments, i.e., the utility of the agent and the adversary changes along the vertices of the graph. They formulate the problem as a mathematical programming problem (either multilinear programming or mixed integer linear programming). Consequently, the computation of the optimal strategy is exponential, yet using optimization tools they manage to get good approximation to the optimal solution. Paruchuri, Tambe et al. (2007) considered the problem of placing stationary security checkpoints in adversarial environments. Similar to our assumptions, they assume that their agents work in an adversarial environment in which the adversary can exploit any predictable behavior of the agents, and that the adversary has full knowledge of the patrolling agents. They model their system using Stackelberg games, which uses policy randomization in the agents' behavior in order to maximize their rewards. The problem is formulated as a linear program for a single agent, yielding an optimal solution for that case. Using this single agent policy, they present a heuristic solution for multiple agents, in which the optimal solution is intractable. Paruchuri, Pearce et al. (2007) further study this problemin cases where the adversarial model is unknown to the agents, although the adversary still has full knowledge of the patrol scheme. They again provide heuristic algorithms for optimal strategy selection by the agents. Pita et al. (2009) continued this research to consider the case in which the adversaries make their choices based on their bounded rationality or uncertainty, rather than make the optimal game-theoretic choice. They considered three different types of uncertainty over the adversary's choices, and provided new mixed-integer linear programs for Stackelberg games that deal with these types of uncertainties. As opposed to all these works that are based on using game-theoretic approaches and provide approximate or heuristic solutions to intractable optimal solutions, in our work we focus on specific characteristics of the patrol guards and the environment, and provide optimal polynomial-time algorithms for finding an optimal patrol strategy for the multi-patrol guards team using the minmax approach. Theoretical work based on stochastic processes that is related to our work is the cat and mouse problem (Coppersmith, Doyle, Raghavan, &Snir, 1993), also known as the predatorprey (Haynes & Sen, 1995) or pursuit evasion problem (Vidal, Shakernia, Kim, Shim, &Sastry, 2002). In this problem, a cat attempts to catch a mouse in a graph where both are mobile. The cat has no knowledge about the mouse's movement, therefore as far as the cat is concerned, the mouse travels similarly to a simple random walk on the graph. We, on the other hand, have worst case assumptions about the adversary. We consider a patrol guard model, in which the movement of the cat is correlated with the movement of a patrol guards, with possible directionality of movement, possible cost of changing directions and possible sensorial abilities. Moreover, in our model the patrol guards travel around a perimeter or a fence, rather than in a general graph. Thus in a sense, our research is concerned with pursuit-evasion on a polyline - open or closed. Other theoretical work by Shieh and Calvert (1992), based on computational geometry solutions, attempts to find optimal viewpoints for patrolling patrol guards. They try to maximize the view of the patrol guards in the area, show that the problem is NP-Hard, and find approximation algorithms for the problem.

3.PATROL GUARDS AND ENVIRONMENT MODEL

In the following section, we provide a description of the patrol guardsmodel, environment model and the adversarial model. We describe the basic model of patrolling around a perimeter (closed polygon). Further environments and patrol guard's models are discussed later.

3.1 The Environment

We consider a patrol in a circular path around a closed polygon P. The path around P is divided into N segments of a length of uniform time distance, i.e., each patrol guards travels through one segment per cycle while sensing it (its velocity is 1 segment / 1-time cycle). This division into segments makes it possible to consider patrols in heterogeneous paths. In such areas, the difficulty of passing through terrains varies from one terrain to another, for example driving in muddy tracks vs. driving on a road. Figure 1 demonstrates a transition from a given area to a discrete cycle. The area, on the left, is given along with its velocity constraints. The path is then divided into segments such that a patrol guards travels through one segment per time cyclewhile monitoring it, i.e., the length of each segment is determined by both the velocity of the patrol guards (corresponding to the time it takes it to travel through the specific segment) and the sensorial capabilities of the patrol guards. After the path is divided into segments with uniform travel time, it is equivalent to considering a simple cycle as appears in the right of Figure 1. Note that the distance between the patrol guards is calculated with respect to the number of segments between them, i.e., the distance is in travel time. For example, if we say that the distance between R_1 and R_2 is 7, then there are 7 segments between them, and if R_1 had remained still, then it would have taken R_2 7 time cycles to reach R_1 (assuming R_2 is headed towards the right direction).

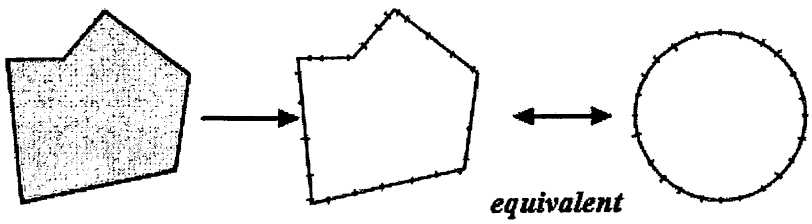


Figure 3.1: An example for creating discrete segments from a circular path with the property that the patrol guards travel through one segment per cycle. The different line structures along the perimeter on the left

correlate to different velocity constraints, which are converted (in the middle figure) to *N* segments in which the patrol guards travel during one-time cycle. This figure is equivalent to the figure in the right, which is a simple cycle divided into *N*unit-time segments.

3.2 Patrolling Patrol Guards Model

We consider a system of $k > 1$ homogenous mobile patrol guards $R_1 \ldots, R_k$, that are required to patrol around a closed polygon. The patrol guards operate in cycles, where each cycle consists of two stages.

1. Compute: Execute the given algorithm, resulting in a goal point, denoted by p_G, to which the patrol guards should travel.
2. Move: Move towards point p_G.

This model is synchronous, i.e. all patrol guards execute each cycle simultaneously. We concentrate our attention to the Compute stage, i.e., how to compute the next goal point. We assume the patrol guards' movement model is directed such that if p_G is behind the patrol guards, it has to physically turn around. Turning around is a costly operation, and we model this cost in time, i.e., if the patrol guards turns around it resides in its segment for τ time units. The case in which the movement model is not directed is discussed in later Section. Throughout the work we assume for simplicity that $\tau = 1$, unless stated otherwise. A key result of this research is that optimal patrolling necessitates patrol guards to be placed at a uniform distance

$$d = N/k \qquad (1)$$

from one another along the perimeter. Consequently, we require the patrol guards to be coordinated in the sense that all patrol guards move in the same direction, and if decided to turn around they do it simultaneously. This requirement guarantees that the uniform distance of d is maintained throughout the execution of thepatrol algorithm. Note that this tightly-coordinated behavior is achievable in centralized systems, or in systems where communication exists between all team members. Other practical implementations may exist (for example uniformly seeding a pseudorandom number generator for all the patrol guards), but they all require coordination inside the team. Distributed systems that cannot assume reliable communication are left for future work.

3.3 Adversarial Model

Our basic assumption is that the system consists of an adversary that tries to penetrate once through the patrolling patrol guards path without being detected. The adversary decides, at any unknown time, through which segment to penetrate. Its penetration time is not instantaneous, and lasts t time units, during which it stays at the same segment.

Definition 1. Let s_i be a discrete segment of a perimeter P which is patrolled by one patrol guard or more. Then the Probability of Penetration Detection in s_i, ppd_i, is the probability that a penetrator going through s_i during t time units will be detected by some patrol guards going through s_i during that period of time.

In other words, ppd_i is the probability that a patrol path of some patrol guards will pass through segment s_i during the time that a penetration is attempted through that segment, hence it is calculated for each segment with respect to the current location of the patrol guards at a given time (since the patrol guards maintain uniform distance between them throughout the execution, this relative location remains the same at all times). We use the general acronym *ppd* when referring to the general term of probability of penetration detection (without reference to a certain segment). Recall that the time distance between every two consecutive patrol guards around the perimeter is $d = N/k$. Therefore, we consider t values between the boundaries

$$(d+\tau)/2 \leq t < d. \qquad (2)$$

The reason for this is that if it takes the patrol guards τ time units to turn, then the patrol guard adjacent to s_0 will have probability > 0 of arriving at every segment $s_i, 0 \leq i \leq t$, while the patrol guard adjacent to s_d has probability > 0 of arriving at segments s_i, $d-(t-\tau) \leq$ *ileqd.* Hence segment s_t+1 has probability > 0 of being visited only if

$$d-(t-\tau) \leq t+1 \qquad (3)$$

$$\Rightarrow (d+\tau+1)/2 \leq t,$$

otherwise there is at least one segment, s_t+1, that has probability 0 of being visited during t time units. Therefore, an adversary having full knowledge on the patrol will always manage to successfully penetrate regardless of the actions taken by the patrolling patrol guards. Note that τ appears in this equation since it influences the number of segments reachable by the patrol guards located in segment s_d+1 if turning around (s_d, $sd_{-1}, \ldots, s_{d/2+\tau}$). On the other hand, if $t \geq d$

then all segments s_i can have $ppd_i = 1$ simply by using a deterministic algorithm. We define the patrol scheme of the patrol guards as the

1. Number of patrol guards, the distance between them and their current position.
2. The movement model of the patrol guards and any characterization of their movement.
3. The patrol guard's patrol algorithm.

The patrol scheme reflects the knowledge obtained by the adversary on the patrolling patrol guards at any given time (hence is not necessarily time dependent).

We consider a strong adversarial model in which the adversary has full knowledge of the patrolling patrol guards. Therefore, the full knowledge adversary knows the patrol scheme, and will take advantage of this knowledge in order to choose its penetration spot as the weakest spot of the patrol, i.e., the segment with minimal *ppd*. The solution concept adopted here (as stated) is similar to the game-theoretic minimax strategy, yielding a strategy that is in equilibrium (none of the players—patrol guards or adversary—has any initiative to diverge from their strategy). The adversary can learn the patrol scheme by observing the behavior of the patrol guards for a sufficient amount of time. Note that in security applications, such strong adversaries exist. In other applications, the adversary models the behavior of the system in the "worst case scenario" from the patrolling patrol guards point of view (similar to the classical Byzantine fault model in distributed systems, see Lynch, 1996). In our environment, the patrol guards are responsible only for detecting penetrations and not handling the penetration (which requires task-allocation methods). Therefore, the case in which the adversary issues multiple penetrations is similar to handling a single penetration, as the patrol guards detect, report and continue to monitor the rest of the path, according to their algorithm.

4.A FRAMEWORK FOR ADVERSARIAL PATROLLING OF PERIMETERS

The environment we consider is a linear environment, in which at each step the patrol guards can decide to either go straight or turn around. The framework we suggest is nondeterministic in the sense that at each time step the decision is done independently, at random, with some probability p. Formally,

$$\textbf{Probability of next move} = f(x) = \begin{cases} p, & \textbf{\textit{Go straight}} \\ 1-p, & \textbf{\textit{turn around}} \end{cases}$$

Since the different patrol algorithms we consider vary in the probability p of the next step, we assert that the probability p characterizes the patrol algorithm. Assume a patrol guard is currently located in segment s_i. Therefore, if the patrol guard is facing segment s_{i+1}, then with a probability of p it will go straight to it and with a probability of $1-p$ it will turn around and face segment s_{i-1}. Similarly, if it is facing segment s_{i-1}, then with a probability of p it will reach segment s_{i-1} and with a probability of $1-p$ it will face segment s_{i+1}. Note that the probability of penetration detection in each segment s_i, $1 \leq i \leq d$, is determined by probability p characterizing the patrol algorithm, therefore $ppd_i$is a function of p, i.e., $ppd_i(p)$. However, whenever possible we will use the abbreviation ppd_i. By the definition of ppd_i, we need to find the probability that s_i will be visited during t time units by some patrol guards. Assuming perfect detection capabilities of the patrol guards, ppd_i is determined only by the first visit of some patrol guards to s_i, since once the intruder is detected the detection mission is successful (specifically, once the segment is visited, the "game" is over). Note that ppd_i is calculated regardless of the actions of the adversary. As stated previously, in order to guarantee optimality of the patrol algorithm, the patrol guards should be uniformly distributed along the perimeter with a distance of $d = N/k$ between every two consecutive patrol guards, and that they are coordinated in the sense that if they are supposed to turn around, they do so simultaneously. In the following theorem and supporting lemmas we prove optimality of these assumptions in a full-knowledge adversarial environment. Lemma 1 follows directly from the fact that the movement of the patrol guards is continuous, thus a patrol guardR_l cannot move from segment si to segment s_{i+j}, $j > 0$, without visiting segments $s_{i+1},\ldots,$ sj_{-1}in between. Note that since $k > 1$ it follows that the number of segments unvisited by R_l is greater than *2t* (otherwise a simple deterministic algorithm would suffice to detect the adversary with probability 1). Therefore, during *t* time units R_l residing initially in segment s_0 cannot visit segment s_i, $i < t$, arriving from the other direction of the perimeter without visiting the segments closer to its current location (s_0) first (this argument holds for segments to the left and to the right of s_0).

Lemma 1. For a given *p*, the functionppd^t_i: $N \Rightarrow [0,1]$ for constant *t* and R_l residing in segment s_0 is a monotonic decreasing function, i.e., as the distance between a patrol guard and a segment increases, the probability of reaching it during *t* time units decreases.

Lemma 2. As the distance between two consecutive patrol guards along a cyclic patrol path is smaller, the *ppd* in each segment is higher and vice versa.

Proof. Consider a sequence S_1 of segments $s_1,\ldots,s_w$ between two adjacent patrol guards, R_l and R_r, where s_1 is adjacent to the current location of R_l and s_w is adjacent to the current location of R_r. Let S_2 be a similar sequence, but with w −1 segments, i.e., the distance between R_l and R_r decreases by one segment. Assume that other patrol guards are at a distance greater than or equal to w −1 from R_l and R_r, and that $w - 1 < t$. Since a patrol guards may influence the ppd in segments that are up to a distance t from it (as it has a probability of 0 of arriving at any segment at a greater distance within t time units), the probability of penetration detection, ppd, in these sequences is influenced only by the possible visits of R_l and R_r. Denote the probability of penetration detection in segment $s_i \in S_j$ by $ppd_{i(j)}$, $1 \le i \le w$, $j \in \{1,2\}$, and the probability that the penetrator will be detected by patrol guards R_x by $ppdx_{i\,(j)}$, $x \in \{l,r\}$. Therefore, for any segment $s_i \in S_j$,

$$ppd_{i(j)} = ppd_{li(j)} + ppdr_{\ i(j)} - ppd_{li(j)} ppd_{r\ i(j)} \qquad (4)$$

(either Rl or Rr will detect the adversary, not both).

Note that either $ppd_{li(j)}$, $ppd_{r\ i(j)}$ or both can be equal to 0. We need to show that

$$ppd_i(2) \ge ppd_i(1), \text{ for all } 1 \le i \le w, \qquad (5)$$

and for at least one segment s_m, $ppd_m(2) > ppd_m(1)$. Specifically, it is sufficient to show that

$$ppd_{li}(2) + ppd_{r\ i}(2) - ppd_{li}(2) ppd_{r\ i}(2) - \{ppd_{li}(1) + ppd_{r\ i}(1) - ppd_{li}(2) ppd_{r\ i}(2)\} \ge 0, \qquad (6)$$

and for some i this inequality is strict. For every segment s_i, $ppd_{li}(1) = ppd_{li}(2)$ (there is no change in its relative location), hence we need to prove that

$$ppd_{r\,i}(2)-ppd_{r\,i}(1) \geq ppd_{li}(2)\{ppd_{r\,i}(2)-ppd_{r\,i}(1)\} \qquad (7)$$

Since $0 \leq ppd_{li}(2) \leq 1$, in order for the inequality to hold, it is left to show that $ppd_{r\,i}(2)-ppd_{r\,i}(1) \geq 0$. From Lemma 1 we know that $ppd_{r\,i(j)}$ is monotonically decreasing, therefore for each i, $ppd_{r\,i}(2) \geq ppd_{r\,i}(1)$, which completes the proof of this inequality. It is left to show that for some i = m, $ppd_{rm}(2)-ppd_{rm}(1) > ppd_{lm}(2)\{ppd_{rm}(2)-ppd_{rm}(1)\}$ (8)

i.e., for some m in which $ppd_{lm}(2) \neq 1$, $ppd_{rm}(2) > ppd_{rm}(1)$. Patrol guard Rr may influence the ppd on both of his sides - segments located to the left and to the right of its current position. Denote the number of influenced segments to its right by y (y may be equal to 0). If y > 0, then $ppd_r\, w-y+1(2) > ppd_r\, w-y(1)$. In other words, Rr has a probability of 0 of reaching the segment with a distance of t + 1 from it in S_1, but in S_2 it is y segments away from it, therefore R_r has a probability greater than 0 to reach it. If y = 0, then $ppd_r w(2) = 1 > ppd_r\, w(1)$, as R_r lies exactly in segment $s_w$of S_2, and $ppd_r\, w(1) = 0$.

Theorem 3. A team of k mobile patrol guards engaged in a patrol mission maximizes minimal ppd if the following conditions are satisfied.

a) The time distance between every two consecutive patrol guards is equal.
b) The patrol guards move in the same direction and speed.

Note that condition b means that all patrol guards move together in the same direction, i.e., if they change direction, then all k patrol guards change their direction simultaneously.

Proof. Following Lemma 2, it is sufficient to show that the combination of conditions a and b yield the minimal distance between two consecutive patrol guards along the cyclic path. Since we have N segments and k patrol guards, there are$({}^{N}C_{k})$possibilities of initial placement of patrol guards along the cycle (patrol guards are homogenous, so this is regardless of their order). If the patrol guards are positioned uniformly along the cycle, then the time distance between each pair of consecutive patrol guards is N/k. This is the minimal value that can be reached. Therefore, clearly, condition a guarantees this minimality. If the patrol guards are not coordinated, then it is possible that two consecutive patrol guards along the cycle, R_i and R_{i+1}, will move in opposite directions. Therefore, the distance between them will increase from N/k to N/k + 2, and by Lemma 2 the ppd in the segments between them will be smaller. If R_i and R_{i+1}

move towards one another, then the distance between them will be N/k −2 and the ppd in the segments between them will become higher. On the other hand, some pair R_j and R_{j+1} exists such that the distance between them increases, as the total sum of distances between consecutive patrol guards remains N, hence the minimal ppd around the cycle will become smaller. Therefore, the only way of achieving the minimal distance (maximal ppd) is by assuring that condition a is satisfied, and maintaining it is achieved by satisfying condition b.

Since when facing a full-knowledge adversary, the goal of the patrol guards is to maximize the minimal ppd along the perimeter, the following corollary follows.

Corollary 4. In the full-knowledge adversarial model, an optimal patrol algorithm must guarantee that the patrol guards are positioned uniformly along the perimeter throughout the execution of the patrol.

4.1 The Penetration Detection Problem

The general definition of the problem is as follows.

Penetration detection (PD) problem: Given a circular fence (perimeter) that is divided into N segments, k patrol guards uniformly distributed around this perimeter with a distance of d = N/k (in time) between every two consecutive patrol guards, assume that it takes t time units for the adversary to penetrate, and the adversary is known to have full-knowledge of the patrol scheme. Let p be the probability characterizing the patrol algorithm of the patrol guards, and let $ppd_i(p)$, $1 \leq i \leq d$ be a description of ppd_i as a function of p. Find the optimal value p, p_{opt}, such that the minimal ppd throughout the perimeter is maximized. Formally,

$$p_{opt} = argmax_{0 \leq p \leq 1}\{ min_{1 \leq i \leq d} ppd_i(p)\} \qquad (9)$$

To summarize the model and the Theorems presented above, an optimal algorithm for multi-patrol guards perimeter patrol under the Markovian strategy assumption for the patrol guards has the following characteristics.

- The patrol guards are placed uniformly around the perimeter with d segments between every two consecutive patrol guards.
- The patrol guards are coordinated in the sense that if they decide to turn around, then they do it simultaneously.

- At each time step, the patrol guards continue straight with a probability of p or turn around with a probability of 1−p, and if they turn around they stay in the same segment for τ time units.

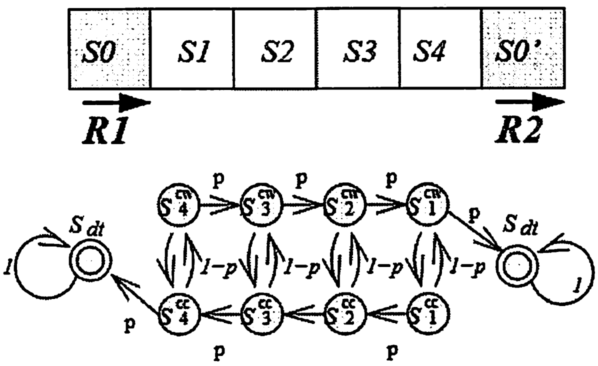
Note that under the above framework (i.e. the framework for homogenous patrol guards), the division of the perimeter into sections of d segments creates an equivalent symmetric environment in the sense that in order to calculate the optimal patrol algorithm it is sufficient to consider only one section of d segments, and not the entire perimeter of N segments. This is due to the fact that each section is completely equivalent to the other, and remains so throughout the execution. We divide the goal of solving the PD problem, i.e., finding an optimal patrol algorithm into two stages.

1. Calculating the d ppd_i functions for each $1 \leq i \leq d$. This is determined according to the patrol guard's movement model (directed or undirected), environment model (perimeter/fence) and sensorial model (perfect/imperfect, local/extended).
2. Given the d ppd_i functions, find the solution to the PD problem, i.e., maximize the ppd in the segment(s) with minimal ppd.

These two steps are independent in the sense that incorporating various different patrol guards models will not change the process of determining the solution to the PD problem, as long as the result of the procedure are d functions representing the ppd values in each segment. On the other hand, if we would like to consider different goal functions other than maximizing the minimal ppd (for example maximizing the expected ppd), it can be done without any change in the first stage, i.e., determining the ppd functions. The important result is that this framework can be applied to both different environment and patrol guard's models (for example fence patrol), and different goal functions (corresponding to different adversarial settings). The first stage for the basic model (perimeter patrol, directed movement model of the patrol guards, patrol guards with perfect local sensing) is described in next Section, and the second stage is described in later Section. Extensions of the first stage to different patrol guards motion models and sensing models are described later.

4.2 Determining the Probability of Penetration Detection

In order to find an optimal patrol algorithm, it is necessary to first determine the probability of penetration detection at each segment s_i (ppd_i), which is a function of p (the probability characterizing the patrol algorithm, as discussed). In this section we present a polynomial time algorithm that determines this probability. As stated previously, based on the symmetric nature of the system, we need to consider only one section of d segments that lie between two consecutive patrol guards, without loss of generality, R_1 and R_2. We use a Markov chain in order to model the possible states and transition between states in the system. In order to calculate the probability of detection in each segment along t time cycles, we use the graphic model G illustrated in Figure 2. For each segment s_i in the original path, $1 \leq i \leq d$, we create two states in G: One for moving in a clockwise direction (scw_i), and the other for moving in a counterclockwise direction (scc i). If R_1 or R_2 reach one of the s_i segments within t time units, then the adversary is discovered, i.e., it does not matter if the segment is visited more than once during these t time units. Therefore, we would like to calculate only the probability of the first arrival to each segment, and this is done by defining the state s_{dt} (corresponding to s_0 and s'_0) as absorbing states, i.e., once a patrol guards passes through s_i once, its additional visits to this segment in this path will not be considered. The edges of G are as follows. One outgoing edge from scw_i to scc_i exists with a probability of $1-p$ for turning around, and one outgoing edge to scw_{i-1} exists with a probability of p for continuing straightforward. Similarly, one outgoing edge from scc_i to scw_i exists with a probability $1-p$ for turning around, and one outgoing edge to scc_{i+1} exists with a probability of p for continuing straightforward.



	s_1^{cc}	s_1^{cw}	s_2^{cc}	s_2^{cw}	s_3^{cc}	s_3^{cw}	s_4^{cc}	s_4^{cw}	s_{dt}
s_1^{cc}	0	1-p	p	0	0	0	0	0	0
s_1^{cw}	1-p	0	0	0	0	0	0	0	p
s_2^{cc}	0	0	0	1-p	p	0	0	0	0
s_2^{cw}	0	p	1-p	0	0	0	0	0	0
s_3^{cc}	0	0	0	0	0	1-p	p	0	0
s_3^{cw}	0	0	0	p	1-p	0	0	0	0
s_4^{cc}	0	0	0	0	0	0	0	1-p	p
s_4^{cw}	0	0	0	0	0	p	1-p	0	0
s_{dt}	0	0	0	0	0	0	0	0	1

Figure 4.1: Conversion of the initial segments and guard locations into a graphical model, and the respective stochastic matrix M. Each segment corresponds to two states: one going clockwise and one going counterclockwise. ppd_i are all paths starting from scw

In the following theorem, we prove that the probability of detecting the adversary by some patrol guards in segment si (i.e., the probability of arriving at a segment during t time units) is equivalent to finding all paths of size at most t to the absorbing state starting at state scc $_i$ .Therefore, it is possible to use the Markov chain representation for determining ppd_i, as shown in Algorithm FindFunc.

Theorem 5. Determining the probability of penetration detection at segment s_i, ppd_i, is equivalent to finding all paths of length at most t that start at scw $_i$ and end in s_{dt} in the Markov chain described above. Proof. For simplicity reasons, in this proof we distinguish between s_{ldt} and s_{rdt}, which are the absorbing state to the left and to the right of the Markov chain (respectively), although practically they are represented by the same state s_{dt}. Clearly, due to the d and t values considered, ppd_i is determined only by the visits of the two patrol guards surrounding the section of d segments $s_1,\ldots,s_d$, denoted by R_l and R_r. Recall that the probability of penetration detection in segment s_i is defined as

$$ppd_i = ppdl_i + ppdr_i - ppdl_i ppdr_i, \qquad (10)$$

where $ppdr_i$ ($ppdl_i$) is the probability that the adversary, penetrating through s_i, is detected by R_r (R_l). We claim that $ppdl_i$ is equivalent to computing the paths starting from scw_i and ending at the absorbing state sr_{dt} (similarly $ppdr_i$ by state sl_{dt}). Clearly, under this claim, since a path of length at most t cannot reach both sr_{dt} and sl_{dt}, it follows that $ppdl_i ppdr_i = 0$, and the theorem will follow. We prove the claim for $ppdl_i$, where $ppdr_i$ follows directly. $ppdl_i$ is the probability that R_l

will reach s_i at least once during t time units. Therefore, we must construct all paths starting from the current location of R_l that passes through s_i, but take into account only the first visit to the segment (everything beyond the first visit results anyway in probability of detection = 1). At each step R_l continues straight with probability p or turns around with probability 1−p. This is equivalent to keeping R_l in place, and moving the segments towards R_l with probability p and switch the segments' direction with probability 1−p. Hence, every path starting at state scw_i (without loss of generality; computing paths starting at scc_i is equivalent, but requires switching the locations of R_l and R_r in the representation) reaching $sr_{dt}$is equivalent to a path started by R_l and passing through s_i. Since sr_{dt} is set to be an absorbing state, every path passing through it will not be considered again, i.e., only the first visit of R_l to s_i is considered, as required.

Using the Markov chain, we can define the stochastic matrix M which describes the state transitions of the system. Figure 2 illustrates the Markov chain and its corresponding stochastic matrix M used for computing the ppd functions. The probability of arrival at segment s_i during t time units, hence the probability penetration detection in that segment, is the scc_{2d+1} + scw_{2d+1} entry of the result of $V_i \times M^t$, where V_i is a vector of 0's, except for a 1 on the 2_{i-1}'th location. The formal description of the algorithm is given by Algorithm 1. Note that the algorithm makes a symbolic calculation, hence the result is a set of d functions of p. The time complexity of Algorithm FindFunc depends on the calculation time of M^t, which is generally $t\times(2d)^3$. However, since M is sparse, methods for multiplying such matrices efficiently exist (e.g., see Gustavson, 1978), reducing the time complexity to $t(2d)^2$, i.e. $O(td^2)$. Since t is bounded by d−1, the time complexity is $O(d^3)$.

Algorithm 1 Algorithm FindFunc(d,t)

1: Create matrix M of size (2d + 1)(2d + 1), initialized with 0s

2: Fill out all entries in M as follows:

3: M [2d + 1,2d + 1] = 1

4: for i ← 1 to 2d do

5: M [i,max{i + 1,2d + 1}] = p

6: M[i,min{1,i−2}] = 1−p

7: Compute MT = M^t

8: Res = vector of size d initialized with 0s

9: for $1 \leq loc \leq d$ do

10: V = vector of size 2d + 1 initialized with 0s.

11: V [2loc] ← 1

12: Res[loc] = V ×MT[2d + 1]

13: Return Res

4.2.1 Handling Higher Values of τ

Algorithm FindFunc and Figure 2 demonstrate the case in which $\tau = 1$, i.e., if the patrol guard turns around (with probability $1-p$) it remains in its current position for one-time step. In the general case, when the patrol guard turns around, the cost of turning—modeled in time— can be higher. In such cases, the Markov chain is modified to represent the value of τ. Specifically, for each segment s_i, instead of having two corresponding states (scw_i and $sccw_i$), we have $2(\tau)$ states: scw_i and $sccw_i$, and one set of $\tau - 1$ states for turning around to each direction (from cw to ccw and vice versa). The probabilities assigned to each of the edges is $1-p$ for the first outgoing edge from scc_i to the first intermediate state towards $sccw_i$ and 1 for each edge on that direction, and similarly on the path from $sccw_i$ to scc_i . See Figure 3 for an illustration. The matrix M is filled out according to the new chain, and the time complexity of creating this matrix grows in a factor of τ—from $(2d + 1)^t$ to $(2\tau d + 1)^t$. However, as long as τ is a constant, the total time complexity does not change.

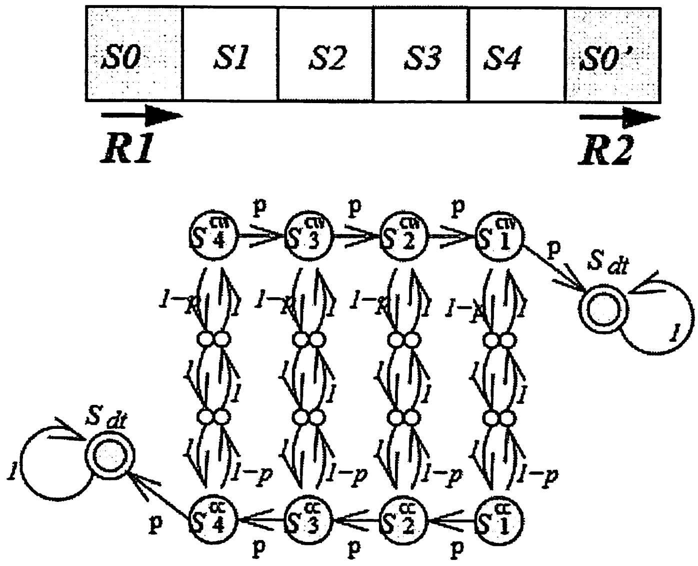


Figure 4.2: Illustration of the Markov chain when $\tau > 1$, and specifically, here $\tau = 3$

4.3 An Optimal Adversarial Patrol Algorithm for Full-Knowledge Adversaries

In cases in which the patrol guards face a full knowledge adversary, it is assumed that the adversary will take advantage of this knowledge to find the weakest spot of the patrol, i.e., the segment with minimal probability of penetration detection. Therefore, an optimal patrol algorithm to handle such an adversary is the one that maximizes the minimal ppd throughout the perimeter. Hence we need to find an optimal p, p_{opt}, such that the minimal ppd throughout the perimeter is maximized. Also here, since our environment is symmetric, we do not need to consider the entire patrol path, but only a section of d segments between two consecutive patrol guards. The input in this procedure is the set of d $ppd_{i(p)}$ functions that were calculated in the previous section (Section 4.2). After establishing d equations representing the probability of detection in each segment, we must find the p value that maximizes the minimal possible value in each segment, where p is continuous in the range $p \in [0,1]$. Denote these equations by $ppd_{i(p)}$, $1 \leq i \leq d$. The maximal minimal value that we are looking is the p value yielding the maximal value inside the intersection of all integrals of $ppd_{i(p)}$. The intersection of all integrals is also known as the lower envelope of the functions (Sharir& Agarwal, 1996). Observing the problem geometrically, consider a vertical sweep line that sweeps the section [0,1] and intersects with all d curves. It seeks the point p in which the minimal intersection point between the sweep line and the curves, denoted by ppd∗(p), is maximal. This p is the maximin point. Since the segment [0,1] and the functions $ppd_1,\ldots,ppd_d$ are continuous, this sweep line solution cannot be implemented. We prove in the following lemma that this point is either an intersection point of two curves, or a local maximum of one curve (see Figure 4). See Algorithm 2 for the formal description of Algorithm FindP.

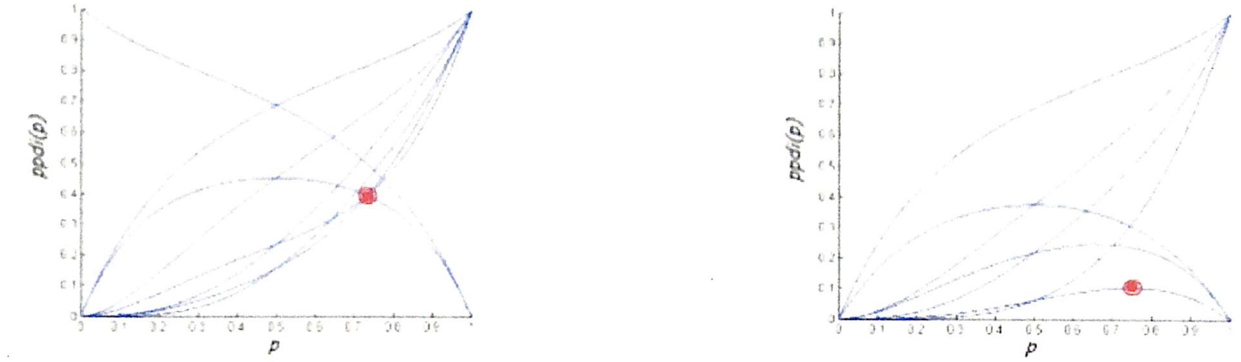


Figure 4.3: An illustration of two possible maximin points (marked by a full circle). The curves represent d $ppd_i(p)$ functions in $p \in [0,1]$. On the left, the maximin point is created by the intersection of two curves. On in the right, the maximin point it is the local maxima of the lowest curve.

In the following, we prove that Algorithm FindP finds point p such that the maximin property is satisfied.

Lemma 6. A point p yields a maximin value ppd∗(p) if the following two properties are satisfied.

a. *$ppd*(p) \leq ppd_i(p)\ \forall 1 \leq i \leq d$.*
b. *One of the two following conditions holds: ppd∗(p) is an intersection of two curves (or more), $ppd_i(p)$ and $ppd_j(p)$ or a local maxima of curve $ppd_k(p)$.*

Proof. Property a. is derived from the definition of a maximin point. Therefore, we are looking for the maximal point that satisfies property a. We must still show that this point, ppd∗(p), is obtained by either an intersection of two or more curves or is a local maximum. Clearly, a maximal point of an integral is found on the border of the integral (the curve itself). The area which is in the intersections of all curves lies beneath parts of curves, $ppd_{i1},\ldots,ppd_{im}$, such that ppd_{ij} is the minimal curve in the section between two points $[l^j,r^j]$ and $\bigcup_{j=1}^{m}[l^j,r^j] = [0,1]$.

By finding the maximal point in each section $ppd^j_{max} = \max\{f(x), x \in [l^j, r^j]\}$, and choosing the maximal between them, i.e., $\max\{ppd^j_{max}, 1 \leq j \leq m\}$, we obtain ppd∗(p). In each section $[l^j, r^j]$ the maximal point can be either inside the section or on the borders of the section. The former case is precisely a local maximum of ppd_{ij}. The latter is the intersection point of two curves $ppd_{ij}-1, ppd_{ij}$ or $ppd_{ij}, ppd_{ij}+1$.

Lemma 7. *A point p exists yielding a maximin value ppd∗(p) > 0.*

Proof. In order to prove the lemma, we need to show that the intersection of all integrals $ppd_1, \ldots, ppd_d$ in the x section [0,1], and the y section (0,1] is not empty. It suffices to show that for every ppd_i, $ppd_i(x) > 0, 0 < x < 1$. Each function $ppd_i, 1 \leq i \leq d$ represents the ppd in a segment si between two patrol guards. From our requirement that $t \geq d/2 + 1$ (for $\tau = 1$), it follows that in all models we consider, for $0 < p < 1$ the ppd $\neq 0$. Note that if p = 0 or p = 1, then ppd is either 0 or 1, but this does not contradict the fact that we have a point guaranteeing ppd∗(p) > 0.

Algorithm FindP finds this point by scanning all possible points satisfying the conditions given in Lemma 6, and reporting the x-value (corresponding to the p value) with a y-value dominated by all ppd_i. The input to the algorithm is a vector of functions ppd_i, $1 \leq i \leq d$ and the value t. Computing the intersections between every pair of functions costs d^2t2: d^2 for all pair computation, t^2 for finding the root of the polynomial using, for example, the Lindsey-Fox method presented by Sitton, Burrus, Fox, and Treitel (2003). Computing the dominance of the resulting points with respect to all other curves is d^2t as well. Therefore the time complexity of Algorithm FindP is the complexity of Algorithm FindFunc, $O((N/k)^3)$, with additional cost of $O(t^2d^2) = O((N/k)^4)$ (the algorithm itself), i.e., jointly $O((N/k)^4)$.

Theorem 8. *Algorithm FindP(F,t) finds point p yielding the maximin value of ppd.*

Proof. Algorithm FindP checks all intersection points between the pair of curves, and the points of local maxima of the curves. It then checks the dominance of these points, i.e., whether in the location these points have a lower value compared to all other curves, and picks the maximal of them. Therefore, if such a point is found, by Lemma 6, this point is precisely the maximin point. Moreover, by Lemma 7 this point exists.

4.4 Examples

We have fully implemented Algorithm FindP in order to find the optimal maximin p for pairs of d's and t's. We use the following examples to illustrate how the relation between t and d is reflected in the ppd values. Recall that when running a deterministic patrol algorithm in all scenarios we handle, the minimal ppd is 0. We assume the patrol guards are initially heading to the clockwise direction.

Algorithm 2 Algorithm FindP(d,t)

1: F ← Algorithm FindFunc(d,t).

2: Set $p_{opt} \leftarrow 0$.

3: for $F_{pivot} \leftarrow F_{1,\ldots,d}$ *do*

4: Compute local maxima $(p_{max}, F_{pivot}(p_{max}))$ *of* F_{pivot} *in the range (0,1).*

5: for each F_i, $1 \leq i \leq d$ *do*

6: Compute intersection point p_i *of* F_i *and* F_{pivot} *in the range (0,1).*

7: if $F_{pivot}(p_i) > F_{pivot}(p_{max})$ *and* $F_{pivot}(p_i) \leq F_k(p_i)\ \forall k$ *then*

8: $p_{opt} \leftarrow p_i$.

9: if $F_{pivot}(p_{max}) > F_{pivot}(p_i)$ *and* $F_{pivot}(p_i) \leq F_k(p_i)\ \forall k$ *then*

10: $p_{opt} \leftarrow p_{max}$.

11: Return $(p_{max}, F_{pivot}(p_{max}))$.

First of all, we have seen that the minimal ppd achieved after running FindP was always more than 0. As t/d → 1, i.e., t increases, then the value of the maximin ppd increases, and vice versa, i.e., as t/d → 1/2, then the value of the maximin ppd decreases. This can be seen clearly in Figure 5. In this case, we have fixed the value of t to 8 and checked the maximin ppd for $9 \leq d \leq 15$. When t/d is close to 1 (d = 9,t = 8) the maximin ppd = 0.423, and the value decreases to 0.05 when t/d is close to 1/2 (d = 15,t = 8). Similar results are seen if we fix the value of d and check for different values of t.

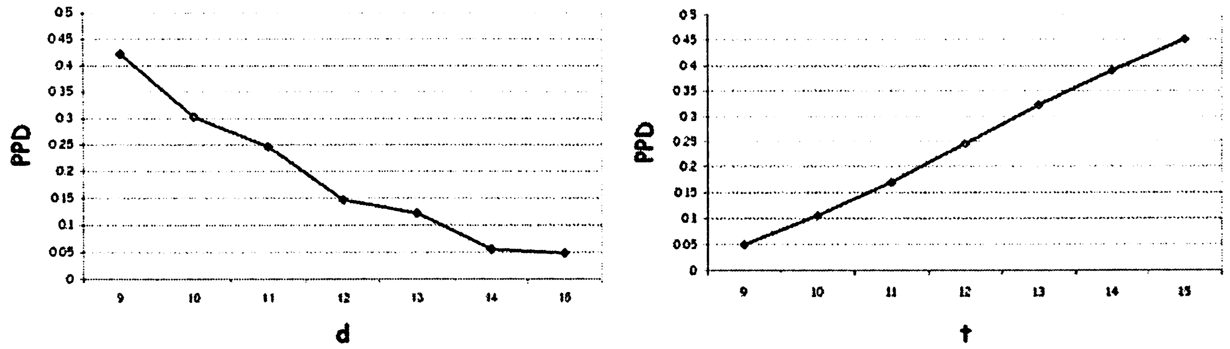


Figure 4.4: On the left, results of maximin ppd for fixed t = 8 and different values of d: the possible maximin ppd decreases as d increases. On the right, results of maximin ppd for fixed d = 16 and different values of t: the possible maximin ppd increases as t increases.

In Figure 6, we present the values of the ppd in all 16 segments, for all different possible values of t ($9 \leq d \leq 15$). It is seen clearly, that the value of ppd usually decreases as the distance from the left patrol guard increases, until it reaches the segment with maximin ppd, then the value rises again until reaching the current location of the patrol guard to the right. The reason lies in the fact that the segments to the left of the segment with the maximin ppd are influenced mostly by the patrol guards on the left, and the segments to the right of that point are mostly influenced by the patrol guards to the right. Since the p's yielding the maximin point in this example have value of greater than 0.8 for all t's, the segment having the maximin value is to the right of the midpoint.

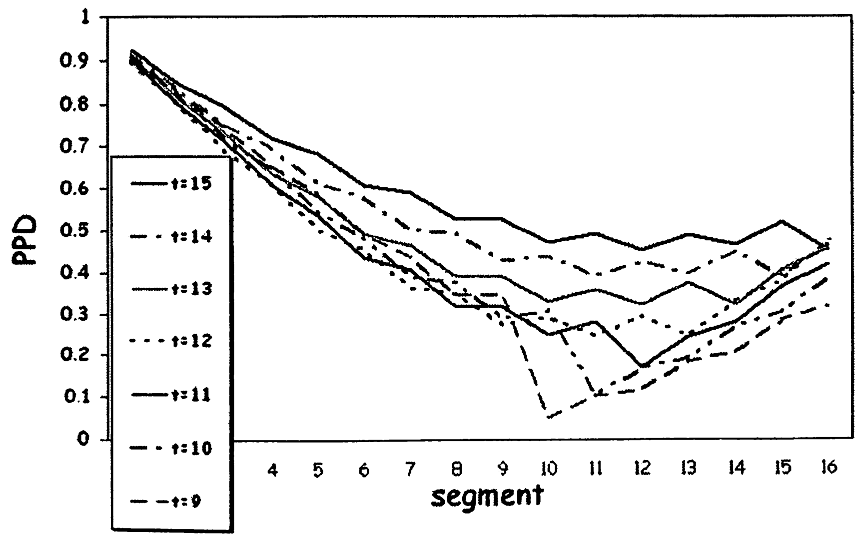


Figure 4.5: ppd values in all 16 segments for all t values (9 to 15)

5. ACCOUNTING FOR MOVEMENT CONSTRAINTS AND SENSING UNCERTAINTY

In this section we describe various ways in which the basic framework of multi-patrol guard patrol can be used to solve the problem of finding an optimal patrol algorithm in various other settings. First, we describe the case in which the movement model of the patrol guards is not necessarily directed. We then discuss various sensing capabilities of the patrol guards in perimeter patrol: imperfect local sensing, perfect long-range sensing and imperfect long-range sensing. Finally, we describe the case in which the patrol guards should travel along an open polyline (fence) rather than a perimeter.

5.1 Different Movement Models

A basic assumption of the patrol guard framework is that the patrol guard's movement model is directed in the sense that if a patrol guard has to go back to visit a point behind him, he has to physically turn around. This directed movement model is suitable for various patrol guards type, like foot patrolling, vehicle patrolling, etc. However, in some cases the patrol guard's movement is undirected, for example if the patrol guard travels along train tracks.We will demonstrate in this section how the basic framework can be used also in the latter case, i.e., if the patrol guard's movement is undirected. We examine the difference in the Markov chain and the resulting ppd in three different cases:

1. Bidirectional Movement model, denoted by BMP. Here, the patrol guard's movement pattern is similar to movement on tracks or a camera going back and forth along a fixed course (omnidirectional patrol guards). In this model, the patrol guards have no movement directionality in the sense that switching directions—right to left and vice versa—does not require physically changing the direction of the patrol guards (turning around).
2. Directional Costly-Turn model, denoted by DCP, the basic framework discussed so far for $\tau \geq 1$. The patrol guard's movement is directed, and turning around is a special operation that has an attached cost in time. Specifically, we show the results here for $\tau = 1$.

3. Directional Zero-Cost model, denoted by DNCP, which is a special case of the DCP model with $\tau = 0$. The patrol guard's movement is directed, yet turning around does not take extra time. This is coherently different from BMP, as in each step the patrol guard does not go either right or left, but straight or back (where each could be either to the right or to the left, depending on the current heading of the patrol guards).

The basic framework can be used for handling all three models simply by adapting the Markov chain to the current model. This changes only lines 5−6 in Algorithm FindFunc. A description of the Markov chains is described in Figure 7. In the BMP model, it moves one step to the right (segment $i + 1$) with a probability of p and one step to the left (segment $i-1$) with a probability of $1-p$. This model is similar to a random walk. The corresponding Markov chain is simple: edges exist from s_i to s_{i+1} with a probability of p and from s_i to s_{i-1} with a probability of $1-p$ (with no related direction). In both the DNCP and DCP models, we assume directionality of movement, hence the patrol guard continues his movement in its current direction with a probability of p, and turns around (rewinds) with a probability of $1-p$. In DCP, if the patrol guard turns around it will remain in segment i (as described in Figure 2). In the DNCP model, the chain is similar to the one above, however edges will exist from scw_i to scc_{i+1} and from scc_i to scw_{i-1} with a probability of $1-p$. See Figure 7 for an illustration of DNCP, DCP and BMP as a Markov chain.

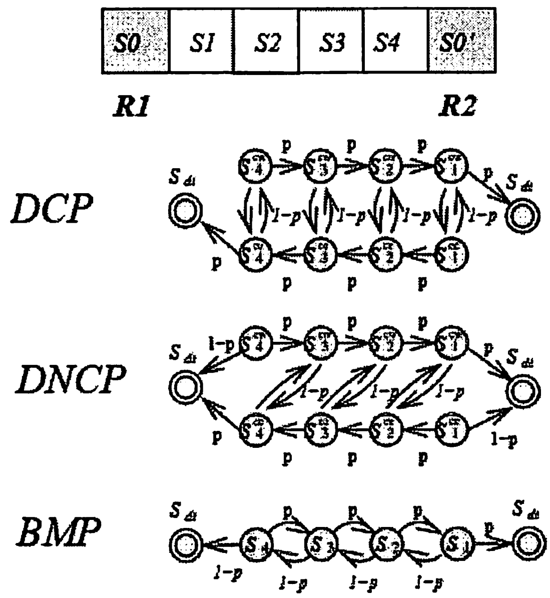


Figure 5.1: Conversion of the initial segments and guard locations into a graphical model in all three movement models

We examined the difference between the resulting ppd values in the three models in a case where d = 16,t = 12 (Figure 8). It is clearly noticeable that the DCP model yields less or equal values of ppd compared to DNCP model throughout the segments. The reason is because when turning around, in the DCP model, the operation costs an extra cycle, therefore the probability of arriving at a segment decreases, compared to the case in which turning around is not costly. Another interesting phenomena is that the ppd values of the BMP are considerably higher (and close to 1) than the values obtained by other models for segments closer to the location of the righthand side patrol guards. The value then decreases dramatically around the value of t and then increases back again. Recall that here there is no directionality of movement, therefore the probability of going right is 0.707 and going left is 1−0.707 = 0.293, which explains this phenomenon. One might have expected to have p = 0.5 in the random walk model (BMP), however by choosing an equal probability for going right and left, the patrol guards will necessarily neglect the segments further away from them (the mid segments between two consecutive patrol guards), resulting in a lower minimal ppd.

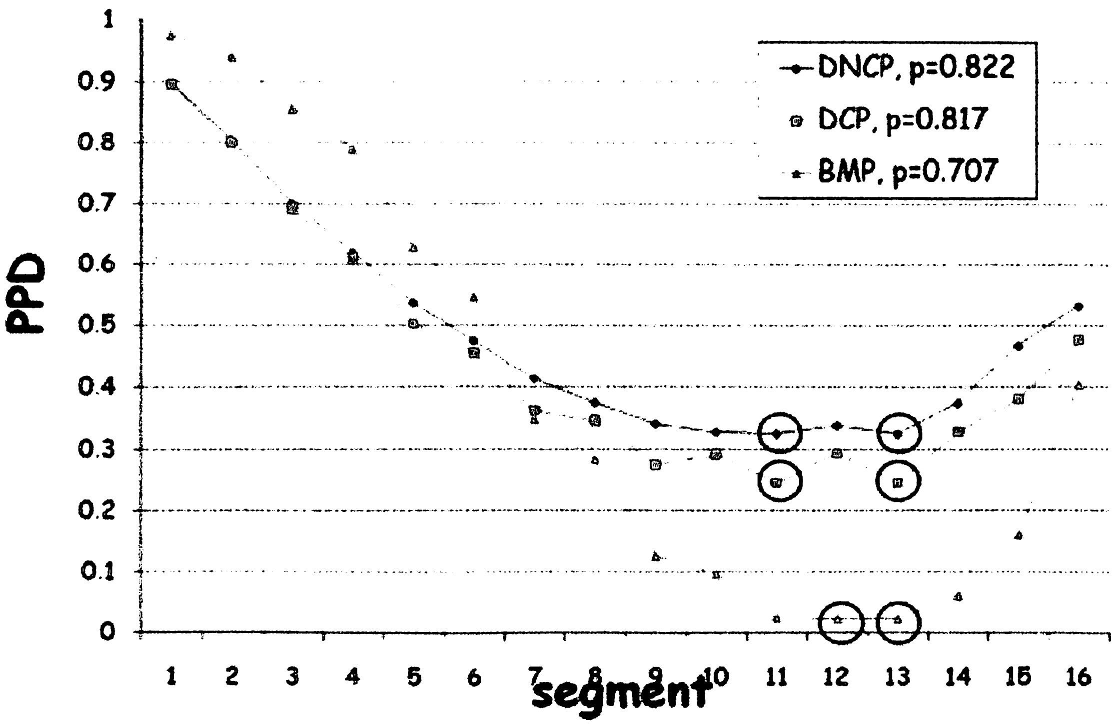


Figure 5.2: Results of maximin ppd values for d = 16 and t = 12 for all three models: DNCP, DCP and BMP. The maximin ppd values are circled

5.2 Perimeter Patrol with Imperfect Penetration Detection

Uncertainty in the perception of the patrol guards should be taken into consideration in practical multi-patrol guards problems. Therefore, we consider the realistic case in which the patrol guards have imperfect sensorial capabilities. In other words, even if the adversary passes through the sensorial range of the patrol guards, it still does not necessarily detect it.

We introduce the ImpDetect model, in which a patrol guard travels through one segment per time cycle along the perimeter while monitoring it, and has imperfect sensing. Denote theprobability that an adversary penetrating through a segment s_i while it is monitored by some patrol guard R and R will actually detect it by $p_d \leq 1$. Note that if $p_d < 1$, revisiting a segment by a patrol guards could be worthwhile—it could increase the probability of detecting the adversary. Therefore the probability of detection in a segment s_i (ppd_i) is not equivalent to the probability of first arriving at s_i (as illustrated in Section 4.2), but the probability of detecting the adversary during some visit y to s_i, $0 \leq y \leq t$. Denote the probability of the y'th visit of some patrol guards to segment s_i by w^y_i .Therefore,ppdi is defined as follows.

$$ppd_i = w^1_i p_d + w^1_i (1-p_d) \times \{w^2_i p_d + w^2_i (1-p_d) \times \{\ldots\{w^t_i \times p_d\}\}\} \quad (11)$$

In other words, the probability of detecting the penetration is the probability that it will be detected in the first visit ($w^1_i \times p_d$) plus the probability that it will not be detected then, but during later stages. This again is the probability that it will be detected during the second visit ($w^2_i \times p_d$) or at later stages, and so on. Note that after t time units, $w^t_i = 0$ for all currently unoccupied segments s_i, and if a patrol guards resides in s_i, then w^t_i is precisely $(1-p_d)^t$. One of the building blocks upon which the optimal patrol algorithms are based, is the assumption that the probability of detection decreases or remains the same as the distance from a patrol guard increases, i.e., it is a monotonic decreasing function. This fact was used in Section 4 in proving that in order to maintain an optimal ppd, the patrol guards must be placed uniformly around the perimeter (with a uniform time distance), and maintain this distance by being coordinated. In order to show this here as well, we first prove that the probability of detection monotonically decreases with the distance from the location of the patrol guards.

Lemma 9. *Let $S = \{s_{-t+\tau},\ldots,s_{-1},s_0,s_1,\ldots,s_t\}$ be a sequence of 2t segments, where patrol guards R_a resides in s_0 at time 0. Then $\forall i \geq 0$, $ppd_i \geq ppd_{i+1}$, and $\forall i \leq 0$, $ppd_i \geq ppd_{i-1}$.*

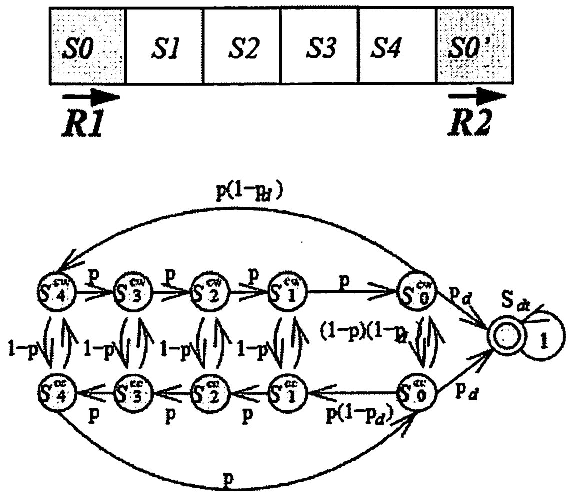
Proof. First, assume that $i > 0$ (positive indexes). By previous Equation, we need to compare between $w^1_i p_d + w^1_i (1-p_d)\times\{w^2_i p_d + w^2_i (1-p_d)\times\{\ldots\{w^t_i \times p_d\}\}\}$ and $w^1_{i+1} p_d + w^1_{i+1}(1-p_d)\times\{w^2_{i+1} p_d + w^2_{i+1}(1-p_d)\times\{\ldots\{w^t_{i+1} \times p_d\}\}\}$. It is therefore sufficient to show that $w^m_i \geq w^m_{i+1}$, for all $1 \leq m \leq t$. We prove this by induction on m. As the base case, consider $m = 1$, i.e., we need to show that $w^1_i \geq w^1_{i+1}$. This is accurately proven in Lemma 1, based on the fact that the movement of the patrol guards is continuous, therefore in order to get to a segment you must visit the segments in between (the formal proof also uses the conditional probability law). We now assume correctness for $m' < m$, and prove that $w^m_i \geq w^m_{i+1}$. Denote the probability that a patrol guard placed at segment s_i will return to s_i within r time units by $x_i(r)$. In our symmetric environment, for every i and j, $x_i(r) = x_j(r)$. Moreover, $\forall r, x_i(r) \geq x_i(r-1)$. Therefore, w^m_i can be described as $\sum_{r+u\leq t} w^{m-1}_i (u)\times x_i(r)$, and similarly $w^m_{i+1} = \sum_{r+u\leq t} w^{m-1}_{i+1} (u)\times x_{i+1}(r)$. By the induction assumption, $w^{m-1}_i \geq w^{m-1}_{i+1}$, and since $x_i(r) = x_{i+1}(r)$, it follows that $w^m_i \geq w^m_{i+1}$, proving the lemma for positive indexes. The negative indexes are a reflective image of the positive indexes, but with $t-\tau$ time units. Since the induction was proven for all t values, the proof for the negative indexes directly follows. The following Theorem follows directly from Lemma 9. The idea behind this is that since the probability of penetration detection decreases as the distance from the patrol guardsgrow, both minimal ppd and average ppd are maximized if the distance between the patrol guards is as small as possible. Since the patrol path is cyclic, this is achieved only if the distance between every two consecutive patrol guards is uniform, and remains uniform. Note that Theorem 10 below is a generalization of Theorem 3 for imperfect sensing (based on the fact that that the general structure of the ppd function remains the same even if the patrol guards might benefit from revisiting a segment, and by that increasing the ppd in that segment).

Theorem 10. *In the full knowledge adversarial model, a patrol algorithm in the ImpDetect model is optimal only if it satisfies two conditions:*

a. The patrol guards are placed uniformly around the perimeter.

b. The patrol guards are coordinated in the sense that if they turn around, they do it simultaneously. By assuring these two conditions, the patrol guards preserve a uniform distance between themselves throughout the execution.

Algorithm for finding ppd_i with imperfect sensorial detection: Find the probability of penetration detection with $p_d \leq 1$ results in a different Markov chain, hence a different stochastic matrix M. Figure 9 demonstrates the new graphical model and the new resulting stochastic matrix M (compared to Figure 2, in which $p_d = 1$). The difference in the algorithm is in the division of s_0 to two states, scw_0 and scc_0, the addition of the absorbing state s_{dt} that represents the detected state and the transitions between these states. The ppd_i is therefore obtained after t+1 steps (compared to t steps) in the s_{dt}'s location in the result vector. The time complexity of the algorithm remains $O(d^4)$.



	s_1^{cc}	s_1^{cw}	s_2^{cc}	s_2^{cw}	s_3^{cc}	s_3^{cw}	s_4^{cc}	s_4^{cw}	s_0^{cc}	s_0^{cw}	s_{dt}
s_1^{cc}	0	1-p	p	0	0	0	0	0	0	0	0
s_1^{cw}	1-p	0	p	0	0	0	0	0	0	0	0
s_2^{cc}	0	0	0	1-p	p	0	0	0	0	0	0
s_2^{cw}	0	p	1-p	0	0	0	0	0	0	0	0
s_3^{cc}	0	0	0	0	0	1-p	p	0	0	0	0
s_3^{cw}	0	0	0	p	1-p	0	0	0	0	0	0
s_4^{cc}	0	0	0	0	0	0	0	1-p	p	0	0
s_4^{cw}	0	0	0	0	0	p	1-p	0	0	0	0
s_0^{cc}	$p(1-p_d)$	0	0	0	0	0	0	0	0	$(1-p)(1-p_d)$	p_d
s_0^{cw}	0	0	0	0	0	0	0	$p(1-p_d)$	$(1-p)(1-p_d)$	0	p_d
s_{dt}	0	0	0	0	0	0	0	0	0	0	1

Figure 5.3: Conversion of the initial segments and patrol guard's locations into a graphical model, and the respective stochastic matrix M for the imperfect sensing model

5.3 Improving Sensing Capabilities in Perimeter Patrol

In this section we present further enhancements by considering various sensing capabilities of the patrol guards. Specifically, we first consider the case in which a patrol guard can sense beyond its currently visited segment. We then offer a solution to the case in which the patrol guard can sense beyond its current position, yet its sensing capabilities are not perfect, and change as a function of the distance from its current position.

5.3.1 Extending (Perfect) Sensing Range

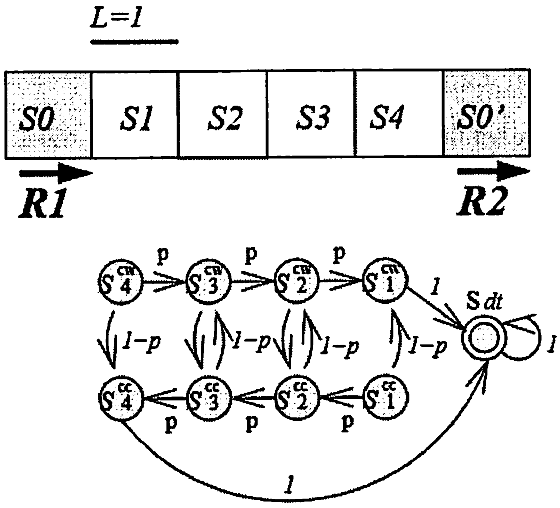
In this section we consider the LRange model, in which the sensorial range of a patrol guards exceeds the section which it currently resides in. Use L to denote the number of segments the patrol guard senses beyond the segment it currently occupies. If $L > 0$, we refer to the L segments as shaded segments. Note that the location of the shaded segments depends on the direction of the patrol guard shading them, and they are always in the direction the patrol guard is facing. A trivial solution to dealing with such a situation is to enlarge the size of the segment, and thus enlarge the length of the time unit used as base for the system, such that it will force L to be 0. However, in this case we lose accuracy of the analysis of the system, as the length of the time cycle should be as small as possible to also suit the velocity of the patrol guards and the value of t. In general, the values of t that can be handled by the system are bounded by its relation to d (the distance between every two patrol guards along the path) - see Section 4. If $L > 0$, this changes. Specifically, if $L = 0$, then the possible values of t considered are $(d+\tau)/2 \leq t \leq d-1$. However, if $L > 0$, then it is possible to handle even smaller values of t, i.e., even if the penetration time of the adversary is short. Formally, the possible values of t are given in the following equation.

$$(d + \tau)/2 - L \leq t \leq d - L - 1 \qquad (12)$$

If t is smaller than $(d+\tau)/2 - L$, then an adversary with full knowledge will manage to penetrate with a probability of 1, i.e., there is a segment (s_{L+1}) which is unreachable within t time units. On the other hand, if t is greater than $d-L-1$, then a simple deterministic patrol algorithm will detect all penetrations with a probability of 1. We assume that during the τ time units the patrol guards turns around, it can sense only its current segment. This change in the sensing model of the patrol guard is reflected in the Markov chain, as seen in Figure 10. The change is that we add 2L arrows to the absorbing state s_{dt}, from scw_1 ,...,scw_L and scc_d ,...,scc_{d-L+1}. The stochastic matrix M changes accordingly, and the probability of penetration detection in segment s_i becomes the result of the vector multiplication $M^{t+1}V_i$, where V_i is a vector of size $2d + 1$ with all entries 0 except for entry corresponding to the location of scw_i, which holds a value of 1, similar to the process described in Algorithm FindFunc (1).

5.3.2 Extending the Sensorial Range Along with Imperfect Detection

In many cases, the actual sensorial capabilities of the patrol guard are composed of the two characteristics described in the previous sections, i.e., the patrol guard can sense beyond his current segment, however the sensing ability is imperfect. Therefore, in this section we introduce

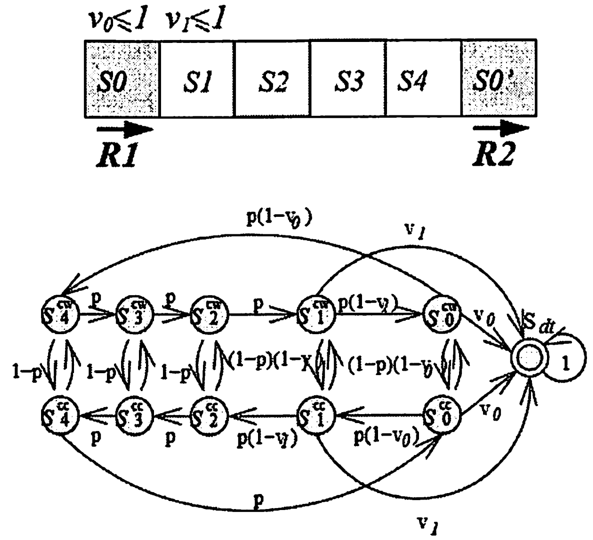


	S_1^{cc}	S_1^{cw}	S_2^{cc}	S_2^{cw}	S_3^{cc}	S_3^{cw}	S_4^{cc}	S_4^{cw}	S_{dt}
S_1^{cc}	0	1-p	p	0	0	0	0	0	0
S_1^{cw}	0	0	0	0	0	0	0	0	1
S_2^{cc}	0	0	0	1-p	p	0	0	0	0
S_2^{cw}	0	p	1-p	0	0	0	0	0	0
S_3^{cc}	0	0	0	0	0	1-p	p	0	0
S_3^{cw}	0	0	0	p	1-p	0	0	0	0
S_4^{cc}	0	0	0	0	0	0	1	0	0
S_4^{cw}	0	0	0	0	0	p	1-p	0	0
S_{dt}	0	0	0	0	0	0	0	0	1

Figure 5.4: An illustration of L segments shaded by patrol guard R. In this case R is facing right, therefore the shaded segments are to its right. The Markov chain changes accordingly, therefore also the stochastic matrix M.

the ImpDetLRange sensorial model, which is a combination of the LRange and the ImpDetect models. Here the patrol guards can sense L segments beyond its current segment, yet the p_d in each segment varies and is not necessarily 1. We therefore describe how to compute ppd_i in this case, which deals with the most realistic form of sensorial capabilities (Duarte & Hu, 2004): imperfect, long range sensing. The information regarding the sensorial capabilities of the patrol guards includes two parameters. The first describes the quantity of the sensing ability, i.e., the number of segments that exceeds the current segment in which patrol guards resides, for which it has some sensing abilities, denoted by L. The second parameter describes the quality of sensing in all segments the patrol guards can sense. This is given in the form of a vector V_S =

$\{v_0, v_1, \ldots, v_L\}$, where v_i is the probability that the patrol guards residing in s_0 will detect a penetration that occurs in segment s_i. We assume that the values in V_S decrease monotonically, i.e., as i increases, v_i decreases or remains the same. The Markov chain in this model, as illustrated in Figure 11, changes in order to reflect the imperfect sensing along with the long range sensing. The absorbing state s_{dt} exist in addition to the states scw_0 and s_0cc. The transition probabilities are added from 2L segments: $\forall i,j\ 0 \le i \le L; d-L+1 \le j \le d$, a transition from scw_i to s_0 with probability v_i and from scc_j to s_0 with probability v_{d-j+1}. In addition, the transition from scw_i to s_icc is with probability $(1-p)(1-v_i)$, from scc_j to scw_j with probability $(1-p)(1-v_{d-j+1})$, hence the transition probability from scw_i to scw_{i-1} is $p(1-v_i)$ and from scc_j to scc_{j+1} is $p(1-v_{d-j+1})$. The probability of penetration detection in segment s_i is the result of M^{t+1} multiplied by V_i in location s_0 of the result vector. Note that also here, similar to the solution described in Section 5.2, since we added a new absorbing state (which takes an extra step to reach), ppd_i is the result in the product of the stochastic matrix and V_i in location s_0 after t + 1 time steps (not t).



	s_1^{cc}	s_1^{cw}	s_2^{cc}	s_2^{cw}	s_3^{cc}	s_3^{cw}	s_4^{cc}	s_4^{cw}	s_0^{cc}	s_0^{cw}	s_{dt}
s_1^{cc}	0	$(1-p)(1-v_1)$	$p(1-v_1)$	0	0	0	0	0	0	0	v_1
s_1^{cw}	$(1-p)(1-v_1)$	0	0	0	0	0	0	0	0	$p(1-v_1)$	v_1
s_2^{cc}	0	0	0	$1-p$	p	0	0	0	0	0	0
s_2^{cw}	0	p	$1-p$	0	0	0	0	0	0	0	0
s_3^{cc}	0	0	0	0	0	$1-p$	p	0	0	0	0
s_3^{cw}	0	0	0	p	$1-p$	0	0	0	0	0	0
s_4^{cc}	0	0	0	0	0	0	0	$1-p$	p	0	0
s_4^{cw}	0	0	0	0	0	p	$1-p$	0	0	0	0
s_0^{cc}	$p(1-v_0)$	0	0	0	0	0	0	0	0	$(1-p)(1-v_0)$	v_0
s_0^{cw}	0	0	0	0	0	0	0	$p(1-v_0)$	$(1-p)(1-v_0)$	0	v_0
s_{dt}	0	0	0	0	0	0	0	0	0	0	1

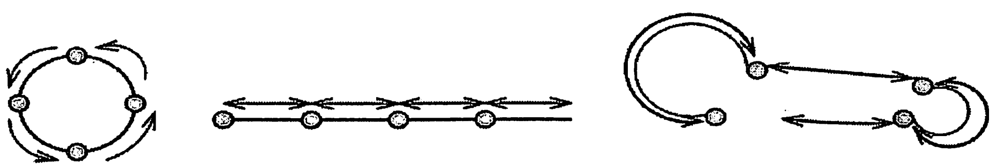
Figure 5.5: An illustration of L segments shaded by guard R, where the probability of detection is not necessarily 1. In this case R is facing right, therefore the shaded segments are to its right. The Markov chain and the stochastic matrix M changes accordingly.

5.4 Multi-Patrol Guards Adversarial Patrolling Along Fences

In our general work, and specifically in previous sections, we assumed the patrol guards travel around a closed, circular, area. In this section we discuss patrolling along an open polyline, also known as fence patrol. First, we will discuss how this patrol is different from perimeter patrol. We will then describe an algorithm for determining ppd_i in fence patrol assuming the patrol guardshave perfect sensing capabilities, and finally we will provide an algorithm for patrol guards with imperfect sensing.

5.4.1 Patrolling Along a Closed Polyline vs. an Open Polyline

In the following, we describe why patrolling along an open polyline is more challenging than patrolling in cyclic environments (closed polyline). The first reason lies in the fact that the patrol guards are required to go back and forth along a part (or parts) of the open polyline. As a result, the elapsed time between two visits of a patrol guard at each point along this line can be almost twice as long as the elapsed time in a circular setting. In Figure 12, we are given two environments: a closed polyline (circle) (a) and an open polyline (b). Note that open polylines b. and c. are equivalent in the sense that each patrol guard travels through one segment per time step, regardless of the shape of the section. Both lines a. and b. are of the same total length l and with the same number of patrol guards (4). In the circular environment, if it takes an adversary more than l/4 time units to penetrate - it will never be able to penetrate even if the patrol guards simply continuously travel with uniform distance between them. However, if the patrol guards travel along an open polyline (b), the maximal time duration between two visits of the patrol guard—even in the best case, is 2l/4−2 (Elmaliach, Shiloni, &Kaminka, 2008). Therefore, a weaker adversary that has a penetration time which is almost twice as long as in the circular fence might still be able to penetrate.



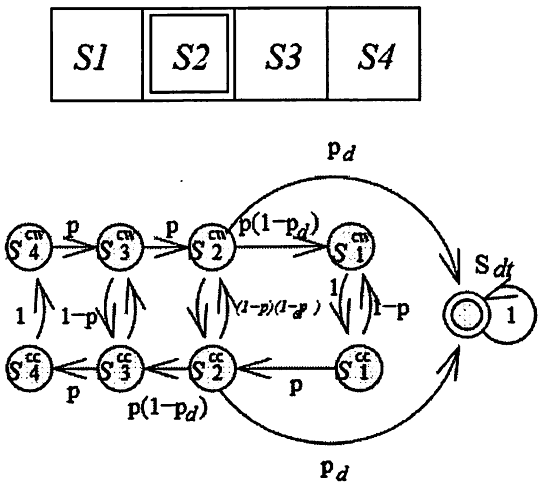
a. *b.* *c.*

Figure 5.6: Illustration of the difference between patrolling along a line and patrolling along a circle, for different polylines

Another reason for the added complication in analyzing the probability of penetration detection in open polyline environments lies in the asymmetric nature of traveling in the segments along time. In a circular environment, if the patrol guards are coordinated and switch directions in unison, then the placement of the patrol guards is symmetric in each time unit. Therefore, all segments in the same distance from some patrol guard (with respect to its direction) have the same probability of penetration detection. Hence in order to calculate an optimal way of movement (in our case the probability p of turning around), it is sufficient to consider only one section of d segments, and the resulted p is equivalent throughout the execution. In an open polyline environment this is not the case. The probability of penetration detection differs with respect to the current location and direction of the patrol guard. Therefore, the algorithm that finds the ppd for each segment, needs to calculate the ppd as a function of p for each segment s_i for each possible initial location of the patrol guard inside the section. Therefore, this results in a matrix of size d×d of the ppd functions (as opposed to a vector of d functions in the circular fence).

5.4.2 Determining ppd_i in an Open Polyline (Fence)

Following the framework for multi-patrol guard patrol along an open line proposed by Elmaliach et al. (2008), we assume each patrol guard is assigned to patrol back and forth along one section of d segments. Given this framework, we would like to compute the optimal patrol algorithm for the patrol guards along the section. Similar to the perimeter patrol case (Section 4.2), we describe the system as a Markov chain (see Figure 13), with its relative stochastic matrix M. Since the patrol guards have directionality associated with their movement, we create two states for each segment: the first for traveling in a segment in the clockwise direction, and the second for traveling in the counterclockwise direction. The probability of turning around at the end of each section is 1, otherwise the patrol guard will continue straight with probability of p, and will turn around with probability of 1−p. Note that the main difference from the perimeter patrol calculation of ppd_i lies in the number of resulting ppd_i functions. In perimeter patrol, due to its symmetric nature, there is one ppd_i function for each segment between the current location of each patrol guards, representing the probability of a patrol guards arriving there during t time units. Here, however, ppd_i depends on the current location of the patrol guards, hence for each location of the patrol guards we have d functions of probability of penetration detection, therefore a total of d^2 such functions (compared to d in perimeter patrol) Denote the probability of penetration detection in segment s_i given that the patrol guard is currently at segment s_j by ppd^j_i. In order to calculate the d ppd^j_i function for all $1 \leq i,j \leq d$, we create d different matrices: $M_1,...,M_d$. Each matrix M_i corresponds to calculating ppd^j_i, i.e., the probability of penetration detection in segment s_i, and from that we calculate ppd^j_i from every current location s_j of the patrol guard (similar to what is done in perimeter patrol). Figure 13 demonstrates the matrix M_2 with which ppd^2_i is calculated. The figure describes the general case of $p_d \leq 1$, i.e., the patrol guards might have imperfect sensing.



	s_1^{cc}	s_1^{cw}	s_2^{cc}	s_2^{cw}	s_3^{cc}	s_3^{cw}	s_4^{cc}	s_4^{cw}	s_{dt}
s_1^{cc}	0	$1-p$	p	0	0	0	0	0	0
s_1^{cw}	1	0	0	0	0	0	0	0	0
s_2^{cc}	0	0	0	$(1-p)(1-p_d)$	$p(1-p_d)$	0	0	0	p_d
s_2^{cw}	0	$p(1-p_d)$	$(1-p)(1-p_d)$	0	0	0	0	0	p_d
s_3^{cc}	0	0	0	0	0	$1-p$	p	0	0
s_3^{cw}	0	0	0	p	$1-p$	0	0	0	0
s_4^{cc}	0	0	0	0	0	0	0	$1-p$	p
s_4^{cw}	0	0	0	0	0	p	$1-p$	0	0
s_{dt}	p	0	0	0	0	0	0	0	1

Figure 5.7: Description of the system as a Markov chain, along with its stochastic matrix M for calculating the ppd in segment s_2.

5.4.3 Optimal Algorithm for Fence Patrol

In the case of fence patrolling, the ppd value depends on the current location of the patrol guard. Consequently, the optimal p value characterizing the patrol of the patrol guards is different for each segment s_i, where $1 \leq i \leq d$. Therefore, there could be different optimal p values with respect to both location and orientation of the patrol guard (2d values). However, it is sufficient to calculate the ppd values only d times (and not 2d times)—only for one direction, as the other direction is a reflective image of the first. In order to find the maximin point for the fence patrolling case, we use algorithm MaximinFence, which finds the value p such that the minimal ppd is maximized, using Algorithm FindP that computes this point by finding the maximal point in the integral intersection of all curves (ppdi). The complete description of the algorithm is shown in Algorithm 3.

Algorithm 3 Procedure MaximinFence(d,t)

1: M ← FindFencePPD(d,t)

2: for i ← 1 to d do

3: OpP[i] ← FindP(d,t) with additional given input M[i] as a vector of ppd functions.

4: Return OpP

6. EVOLUTIONARY GAME THEORY

Game theory, studies and models situations of competition and conflict – of cooperation and defection – between several interacting agents, for shared resources (Webb 2007). We use game theory in this work to model the interactions between possible adversaries within the landscape, and the different vulnerability factors for the animals in a wildlife reserve which would facilitate or restrict the adversaries for penetration.

Let $G(\Theta, \Sigma, \Pi)$ be a normal form, strategic game where $\forall i \in I = \{1,...,n\} \subset \aleph, n \geq 2$,

(i) $\Theta = \{\Theta_i\}$ is the set of interacting agents or players;

(ii) $\Sigma_i \neq \{\}$ is the set of strategies for the player $\Theta_i . \Sigma = \Sigma_1 \times ... \times \Sigma_n$ is the space of strategies, with $\sigma = (\sigma_1,...,\sigma_n) \in \Sigma$ being a strategy profile of the game G;

(iii) $\Pi_i : \Sigma \rightarrow \Re$ is the payoff function, which assigns to each strategy profile σ a real number $\Pi_i(\sigma)$, the payoff earned by the player Θ_i when σ is played in G, $\Pi = \Pi_1 \times ... \times \Pi_n$ is the space of payoff functions in the game.

Let the game G be repeated in periods of discrete time $t \in \aleph$. Assume that the players are 'hardwired' to play only pure strategies in G. Thus each strategy set Σ_i is a member of the standard basis for the strategy space Σ where the i^{th} coordinate is 1 and the rest are zeroes, and thus would correspond to a corner point of the simplex $\Lambda = \left\{ \hat{p} = (p_1, p_2,..., p_n)^T \in \Re : p_i \geq 0, i \in N, \sum_{i=1}^{n} p_i = 1 \right\}$, which is the simplex corresponding to Σ.

In this section,we explore the notion of evolutionary game theory, which shows that the basic ideas of game theory can be applied even to situations in which no individual is overtly reasoning, or even making explicit decisions. Rather, game-theoretic analysis will be applied to settings in which individuals can exhibit different forms of behavior (including those that may not be the result of conscious choices), and we will consider which forms of behavior have the ability to persist in the population, and which forms of behavior have a tendency to be driven out by others. As its name suggests, this approach has been applied most widely in the area of evolutionary biology, the domain in which the idea was first articulated by John Maynard Smith and G. R. Price.

The key insight of evolutionary game theory is that many behaviors involve the interaction of multiple players in a population, and the success of any one of these players depends on how its behavior interacts with that of others. So the fitness of an individual player can't be measured in isolation; rather it has to be evaluated in the context of the full population in which it lives. This opens the door to a natural game-theoretic analogy:anplayer's genetically-determined characteristics and behaviors are like its strategy in a game, its fitness is like its payoff, and this payoff depends on the strategies (characteristics) of the players with which it interacts. Written this way, it is hard to tell in advance whether this will turn out to be a superficial analogy or a deep one, but in fact the connections turn out to run very deeply: game-theoretic ideas like equilibrium will prove to be a useful way to make predictions about the results of evolution on a population.

6.1 Fitness as a Result of Interaction

To make this concrete, we now describe a first simple example of how game-theoretic ideas can be applied in evolutionary settings. This example will be designed for ease of explanation rather than perfect fidelity to the underlying biology; but after this we will discuss examples where the phenomenon at the heart of the example has been empirically observed in a variety of natural settings. For the example, let's consider a particular species of beetle, and suppose that each beetle's fitness in a given environment is determined largely by the extent to which it can find food and use the nutrients from the food effectively. Now, suppose a particular mutation is introduced into the population, causing beetles with the mutation to grow a significantly larger body size. Thus, we now have two distinct kinds of beetles in the population — small ones and large ones. It is actually difficult for the large beetles to maintain the metabolic requirements of their larger body size — it requires diverting more nutrients from the food they eat — and so this has a negative effect on fitness. If this were the full story, we'd conclude that the large-body-size mutation is fitness decreasing, and so it will likely be driven out of the population over time, through multiple generations. But in fact, there's more to the story, as we'll now see.

Interaction Among Players. The beetles in this population compete with each other for food – when they come upon a food source, there's crowding among the beetles as they each try to get as much of the food as they can. And, not surprisingly, the beetles with large body sizes are more effective at claiming an above-average share of the food. Let's assume for simplicity that food competition in this population involves two beetles interacting with each other at any given point in time. (This will make the ideas easier to describe, but the principles we develop can also be applied to interactions among many individuals simultaneously.) When two beetles compete for some food, we have the following possible outcomes.

- When beetles of the same size compete, they get equal shares of the food.
- When a large beetle competes with a small beetle, the large beetle gets the majority of the food.
- In all cases, large beetles experience less of a fitness benefit from a given quantity of food, since some of it is diverted into maintaining their expensive metabolism.

Thus, the fitness that each beetle gets from a given food-related interaction can be thought of as a numerical payoff in a two-player game between a first beetle and a second beetle, as follows. The first beetle plays one of the two strategies Small or Large, depending on its body size, and the second beetle plays one of these two strategies as well. Based on the two strategies used, the payoffs to the beetles are described by Figure 14.

		Beetle 2	
		Small	*Large*
Beetle 1	*Small*	5,5	1,8
	Large	8,1	3,3

Figure 6.1. The Body-Size Game

Notice how the numerical payoffs satisfy the principles just outlined: when two small beetles meet, they share the fitness from the food source equally; large beetles do well at the expense of small beetles; but large beetles cannot extract the full amount of fitness from the food source. (In this payoff matrix, the reduced fitness when two large beetles meet is particularly pronounced, since a large beetle has to expend extra energy in competing with another large beetle.)

6.2 Evolutionarily Stable Mixed Strategies

As a further step in developing an evolutionary theory of games, we now consider how to handle cases in which no strategy is evolutionarily stable. In fact, it is not hard to see how this can happen, even in two-player games that have pure-strategy Nash equilibria. Perhaps the most natural example is the Hawk-Dove Game from, and we use this to introduce the basic ideas of this section. Recall that in the Hawk-Dove Game, two animals compete for a piece of food; an animal that plays the strategy Hawk (H) behaves aggressively, while an animal that plays the strategy Dove (D) behaves passively. If one animal is aggressive while the other is passive, then the aggressive animal benefits by getting most of the food; but if both animals are aggressive, then they risk destroying the food and injuring each other. This leads to a payoff matrix as shown in Figure 15. Now let's consider the same game in a setting where each animal is genetically hard-wired to play a particular strategy.

		Animal 2	
		D	*H*
Animal 1	*D*	3,3	1,5
	H	5,1	0,0

Figure 6.2. Hawk-Dove Game

Neither D nor H is a best response to itself, and so using the general principles from the last two sections, we see that neither is evolutionarily stable. Intuitively, a hawk will do very well in a population consisting of doves — but in a population of all hawks, a dove will actually do better by staying out of the way while the hawks fight with each other. As a two-player game in which players are actually choosing strategies, the Hawk-Dove Game has two pure Nash equilibria: (D, H) and (H, D). But this doesn't directly help us identify an evolutionarily stable strategy, since thus far our definition of evolutionary stability has been restricted to populations in which (almost) all members play the same pure strategy. To reason about what will happen in the Hawk-Dove Game under evolutionary forces, we need to generalize the notion of evolutionary stability by allowing some notion of "mixing" between strategies.

Defining Mixed Strategies in Evolutionary Game Theory. There are at least two natural ways to introduce the idea of mixing into the evolutionary framework. First, it could be that each individual is hard-wired to play a pure strategy, but some portion of the population plays one strategy while the rest of the population plays another. If the fitness of individuals in each part of the population is the same, and if invaders eventually lose off, then this could be considered to exhibit a kind of evolutionary stability. Second, it could be that each individual is hard-wired to play a particular mixed strategy — that is, they are genetically configured to choose randomly from among certain options with certain probabilities. If invaders using any other mixed strategy eventually die off, then this too could be considered a kind of evolutionary stability. We will see that for our purposes here, these two concepts are actually equivalent to each other, and we will focus initially on the second idea, in which individuals use mixed strategies. Essentially, we will find that in situations like the Hawk-Dove game, the individuals or the population as a whole must display a mixture of the two behaviors in order to have any chance of being stable against invasion by other forms of behavior. The definition of an evolutionarily stable mixed strategy is in fact completely parallel to the definition of evolutionary stability we have seen thus far — it is simply that we now greatly enlarge the set of possible strategies, so that each strategy corresponds to a particular randomized choice over pure strategies. Specifically, let's consider the General Symmetric Game from Figure16.

		Player 2	
		S	T
Player 1	S	a	c
	T	b	d

Figure 6.3. A sample game matrix

A mixed strategy here corresponds to a probability p between 0 and 1, indicating that the player plays S with probability p and plays T with probability 1−p. As in our discussion of mixed strategies, this includes the possibility of playing the pure strategies S or T by simply setting p = 1 or p = 0. When Player 1 uses the mixed strategy p and Player 2 uses the mixed strategy q, the expected payoff to Player 1 can be computed as follows. There is a probability pq of an (X, X) pairing, yielding a for the first player; there is a probability p(1−q) of an (X, Y) pairing, yielding b for the first player; there is a probability (1−p) q of a (Y, X) pairing, yielding c for the first player; and there is a probability (1−p) (1−q) of a (Y, Y) pairing, yielding d for the first player. So the expected payoff for this first player is

$$V(p, q) = pqa + p(1-q)\, b + (1-p)\, qc + (1-p)\,(1-q)\, d. \tag{13}$$

As before, the fitness of a player is its expected payoff in an interaction with a random member of the population. We can now give the precise definition of an evolutionarily stable mixed strategy.

In the General Symmetric Game, p is an evolutionarily stable mixed strategy if there is a (small) positive number y such that when any other mixed strategy q invades p at any level $x < y$, the fitness of a player playing p is strictly greater than the fitness of an player playing q.

This is just like our previous definition of evolutionarily stable (pure) strategies, except that we allow the strategy to be mixed, and we allow the invaders to use a mixed strategy. An evolutionarily stable mixed strategy with $p = 1$ or $p = 0$ is evolutionarily stable under our original definition for pure strategies as well. However, note the subtle point that even if S were an evolutionarily stable strategy under our previous definition, it is not necessarily an evolutionarily stable mixed strategy under this new definition with $p = 1$. The problem is that it is possible to construct games in which no pure strategy can successfully invade a population playing S, but a mixed strategy can. As a result, it will be important to be clear in any discussion of evolutionary stability on what kinds of behavior an invader can employ. Directly from the definition, we can write the condition for p to be an evolutionarily stable mixed strategy as follows: for some y and any $x<y$, the following inequality holds for all mixed strategies $q \neq p$:

$$(1-x)\,V(p,p) + xV(p,q) > (1-x)\,V(q,p) + xV(q,q). \qquad (14)$$

This inequality also makes it clear that there is a relationship between mixed Nash equilibria and evolutionarily stable mixed strategies, and this relationship parallels the one we saw earlier for pure strategies. In particular, if p is an evolutionarily stable mixed strategy then we must have $V(p, p) \geq V(q, p)$, and so p is a best response to p. As a result, the pair of strategies (p, p) is a mixed Nash equilibrium. However, because of the strict inequality, it is possible for (p, p) to be a mixed Nash equilibrium without p being evolutionarily stable. So again, evolutionary stability serves as a refinement of the concept of mixed Nash equilibrium.

6.3 Evolutionarily Stable Mixed Strategies in the Hawk-Dove Game

Now let's see how to apply these ideas to the Hawk-Dove Game. First, since any evolutionarily stable mixed strategy must correspond to a mixed Nash equilibrium of the game, this gives us a way to search for possible evolutionarily stable strategies: we first work out the mixed Nash equilibria for the Hawk-Dove, and then we check if they are evolutionarily stable. As we saw in order for (p, p) to be a mixed Nash equilibrium, it must make the two players indifferent between their two pure strategies. When the other player is using the strategy p, the expected payoff from playing D is

$$3p+(1-p) = 1+2p, \qquad (15)$$

while the expected payoff from playing H is $5p$. Setting these two quantities equal (to capture the indifference between the two strategies), we get p =1 /3. So (1/3,1/3) is a mixed Nash equilibrium. In this case, both pure strategies, as well as any mixture between them, produce an expected payoff of 5/3 when played against the strategy p =1 /3. Now, to see whether p =1 /3 is evolutionarily stable, we must check Inequality when some other mixed strategy q invades at a small level x. Here is a first observation that makes evaluating this inequality a bit easier. Since (p,p) is a mixed Nash equilibrium that uses both pure strategies, we have just argued that all mixed strategies q have the same payoff when played against p. As a result, we have V (p,p)=V (q,p) for all q. Subtracting these terms from the left and right of Inequality, and then dividing by x, we get the following inequality to check: V (p,q) >V(q,q). The point is that since (p,p) is a mixed equilibrium, the strategy p can't be a strict best response to itself — all other mixed strategies are just as good against it. Therefore, in order for p to be evolutionarily stable, it must be a strictly better response to every other mixed strategy q than q is to itself. That is what will cause it to have higher fitness when q invades. In fact, it is true that V (p, q) >V(q, q) for all mixed strategies $q \neq p$, and we can check this as follows. Using the fact that p =1 /3, we have

$$V(p, q) = (1/3) \cdot q \cdot 3+(1/3)(1-q) \cdot 1+(2/3) \cdot q \cdot 5 = 4q + 1/3 \qquad (16)$$

while

$$V(q, q) = q^2 \cdot 3 + q(1-q) \cdot 1 + (1-q) \cdot q \cdot 5 = 6q-3q^2. \qquad (17)$$

Now we have

$$V(p, q) - V(q, q) = 3q^2 -2q + 1/3 = 1/3(9q^2 -6q + 1) = 1/3(3q-1)^2 \qquad (18)$$

This last way of writing V (p,q)−V (q,q) shows that it is a perfect square, and so it is positive whenever $q \neq 1/3$. This is just what we want for showing that V (p,q)>V(q,q) whenever $q \neq p$, and so it follows that p is indeed an evolutionarily stable mixed strategy.

7 MINIMUM SPANNING TREE AND HAMILTONIAN CIRCUIT

Landscapes are dynamic and characteristically possess structural (pattern) and functional (process) attributes. Patrolling paths, being integral components of landscapes law enforcement, are characterized by two distinct categories of components, namely, pattern and process components (Chetkiewicz et al. 2006). The structural patrolling path between the source and sink points of patrolling by the forest guards is given by the physical existence of the landscape between the patches. The functional patrolling path is a product of both – adversaries and vulnerability factors. Patrolling paths thus, may be considered as emergent phenomena, caused by the interaction between pattern and process attributes of the vulnerability factors in a wildlife reserve. The essential function and utility of patrolling paths is thus to connect at least one pair of source and sink of significance, and thus ensure gene flow between spatially separate populations of species, fragmented due to landscape modifications, by supporting the movements of processes (Baum et al. 2004; Beier and Loe 1992; Beier and Noss 1998; Briers 2002; Chetkiewicz et al. 2006; Dutta et al. 2013; Henein and Merriam 1990; Johnsingh et al. 1990; Lindenmeyer et al. 2008; Pulliam 1988; Sharma et al. 2013).

7.1 8 – Neighborhood traversing

Neighbourhood of a pixel p at position x,y is aset N(p) of pixels defined relative to p.

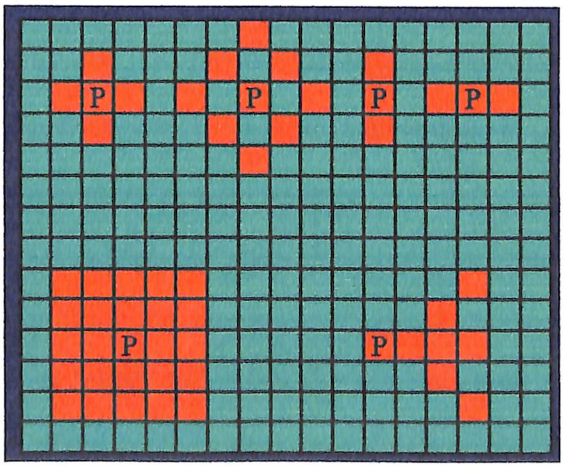


Figure 7.1. A sample neighborhood Example $N(p) = \{(x,y): |x-x_P|=1, |y-y_P| = 1\}$

Usually neighbourhoods are used which are close to discs, since properties of the Euclidian metric are often useful.The most prominent neighbourhoods are the 4-Neighborhood and the 8-

Neighborhood. We apply the logics of Minimum Spanning Tree and Hamiltonian Circuit on the 8-Neighborhood complex of each grid. The next critical grid comes to the central position in the succeeding steps and its 8-Neighborhood is checked for further identification of the critical grid and thus defining the path for the patrol guards.

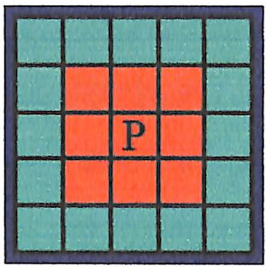


Figure 7.2. A sample 8-Neighborhood Example.

7.2 Minimum spanning tree

A graph $\Gamma(V(\Gamma), E(\Gamma), \psi_\Gamma)$ (henceforth Γ) is an ordered triple comprising a set $V(\Gamma)$ of vertices, a set $E(\Gamma)$ of edges, such that $V \cap E = \phi$, and an incidence function $\psi_\Gamma : E \to [V]^2$ where $[V]^2$ is the set of unordered pair of (not necessarily distinct) vertices of Γ, $\ni$ $e \mapsto \psi_\Gamma(e) = \{v_i, v_j\}, v_i, v_j \in V, \forall e \in E$. The vertices v_i and v_j are incident with the edge e, and vice versa. In the aforesaid, the edge e joins the vertices v_i, v_j, which, in turn, are the end vertices of e. Also, v_i, v_j connected via the incidence function ψ_Γ, are adjacent to each other. Γ, as defined thus, is an undirected graph. Γ is finite if both V and E are finite sets. Then, $|V|$ the order and $|E|$ the size, define the two parameters of Γ respectively. The degree of a vertex $v_i \in \Gamma$ is the number of edges for which v_i is an end vertex. A path in Γ is a sequence of vertices $v_1, v_2, ..., v_n$ and a sequence of distinct edges $e_1, e_2, ..., e_{n-1}$ such that each successive pair of vertices v_k, v_{k+1} are adjacent and are the end vertices of e_k. A path that begins and ends at the same vertex is a cycle. Γ is acyclic if it contains no cycle and is connected if there exists a path from any vertex to any other vertex in Γ. For the present work, we shall consider Γ to be undirected and finite graph.

A tree T is a connected acyclic graph, and a vertex of the tree that has degree exactly one is a leaf of the tree. If there exists a vertex $v_0 \in T$ such that there exists a unique path from v_0 to every other vertex in T but no path from v_0 to v_0, then v_0 is the root of the tree T. A tree T is a spanning tree of the connected graph Γ if it is a spanning subgraph of Γ with vertex set $V(\Gamma)$. We omit the proofs of the following propositions and theorems that we mention for the sake of providing the basis for our arguments and deductions in the paper.

Proposition 11. *In a tree, any two vertices are connected by exactly one path.*

Proposition 12. *Every nontrivial tree has at least two leaves.*

Theorem 13. *If* $T(V(T), E(T))$ *is a tree, then* $|E(T)| = |V(T)| - 1$.

Let T be a tree in the graph Γ. If $|V(T)| = |V|$, then T is a spanning tree of Γ.

Theorem 14. *A graph is connected if and only if it has a spanning tree.*

7.3 Hamiltonian circuit

A Hamiltonian path is a path that visits each vertex exactly once. A graph that contains a Hamiltonian path is called a traceable graph.

A Hamiltonian circuit is a cycle that visits each vertex exactly once (except for the vertex that is both the start and end, which is visited twice). A graph that contains a Hamiltonian circuit is called a Hamiltonian Graph. The first algorithm for finding a Hamiltonian cycle on a directed graph was the enumerative algorithm of Martello. There aren! different sequences of vertices that *might* be Hamiltonian paths in a given *n*-vertex graph (and are, in a complete graph), so a brute force search algorithm that tests all possible sequences would be very slow. There are several faster approaches. A search procedure by Frank Rubin divides the edges of the graph into three classes: those that must be in the path, those that cannot be in the path, and undecided. As the search proceeds, a set of decision rules classifies the undecided edges, and determines whether to halt or continue the search. The algorithm divides the graph into components that can be solved separately. Also, a dynamic programming algorithm of Bellman, Held, and Karp can be used to solve the problem in time O ($n^2 2^n$). In this method, one determines, for each set S of vertices and each vertex v in S, whether there is a path that covers exactly the vertices in S and ends at v. For each choice of S and v, a path exists for (S, v) if and only if v has a neighbor w such that a path exists for $(S - v, w)$, which can be looked up from already-computed information in the dynamic program (Bellman, R.and Held, M.; Karp, R. M., 1962).

Andreas Björklund provided an alternative approach using the inclusion–exclusion principle to reduce the problem of counting the number of Hamiltonian cycles to a simpler counting problem, of counting cycle covers, which can be solved by computing certain matrix determinants. Using this method, he showed how to solve the Hamiltonian cycle problem in arbitrary *n*-vertex graphs by a Monte Carlo algorithm in time $O(1.657^n)$; for bipartite graphs this algorithm can be further improved to time $o(1.415^n)$ (Björklund, Andreas, 2010).

For graphs of maximum degree three, a careful backtracking search can find a Hamiltonian cycle (if one exists) in time $O(1.251^n)$ (Iwama, Kazuo; Nakashima, Takuya, 2007).

The problem of finding a Hamiltonian cycle or path is in FNP; the analogous decision problem is to test whether a Hamiltonian cycle or path exists. The directed and undirected Hamiltonian cycle problems were two of Karp's 21 NP-complete problems. They remain NP-complete even for undirected planar graphs of maximum degree three, for directed planar graphs with indegree and outdegree at most two, for bridgeless undirected planar 3-regularbipartite graphs, and for 3-connected 3-regular bipartite graphs. However, putting all of these conditions together, it remains open whether 3-connected 3-regular bipartite planar graphs must always contain a Hamiltonian cycle, in which case the problem restricted to those graphs could not be NP-complete; see Barnette's conjecture.

In graphs in which all vertices have odd degree, an argument related to the handshaking lemma shows that the number of Hamiltonian cycles through any fixed edge is always even, so if one Hamiltonian cycle is given, then a second one must also exist. However, finding this second cycle does not seem to be an easy computational task. Papadimitriou defined the complexity classPPA to encapsulate problems such as this one.

8 MODELLING

For the purpose of the present work, we assume that the patrol *chaukis* in the wildlife reserve constitute the vertices and the collection of patrolling paths that connect any two of the patrol *chaukis* constitute the edges, comprising the focal wildlife reserve as a graph $\Gamma(V, E, \psi_\Gamma)$.

In consonance with the objective of estimating the presence of a patrolling network across the focal wildlife reserve, the modelling considers only the topology of the network between the different patrol *chaukis*. We assume that the flux between any twopatrol *chaukis (can be self-repeating* patrol *chaukis also)*would be symmetric on the network.

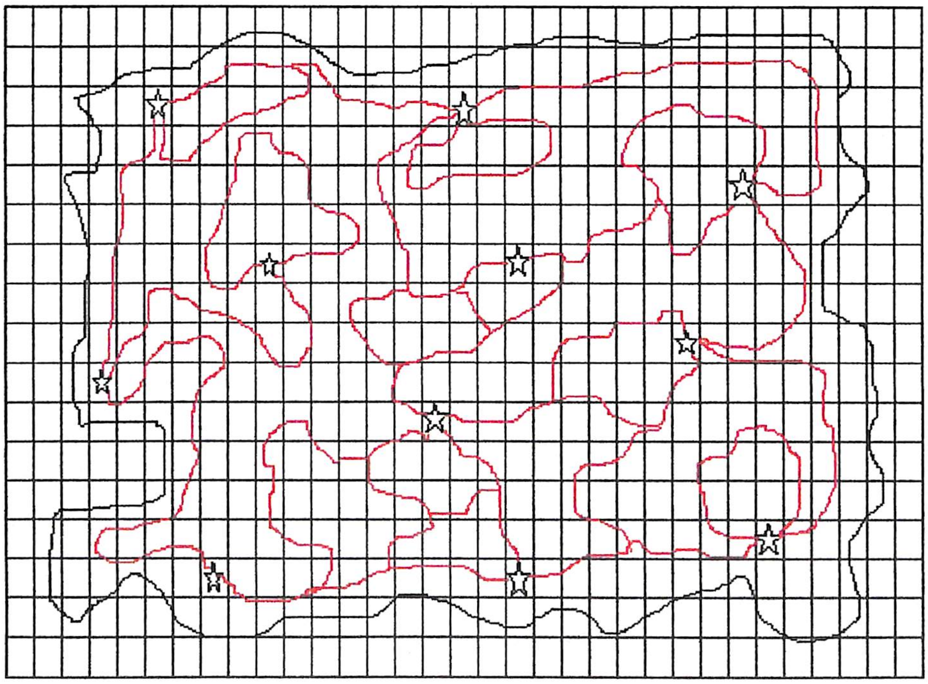


Figure. 8.1 Hypothetical wildlife reserve (black curve in the grid pattern) showing patrol *chaukis* (star shapes), patrol paths between the *chaukis* (red curves joining the shapes, also self-looped) and the matrix (grid pattern)

In Figure. 17, the wildlife reserve is represented by a black curve in the grids, while the star vertices represent *chaukis* for the patrol guards; with the connections between the *chaukis* represented by the red lines. The grid pattern in the figure represents the matrix, a component of the wildlife reserve that is neither *chaukis* nor patrol path in the landscape (Chetkiewicz et al. 2006). The objective for the work is to compute a path joining the different *chaukis*, which would minimize the risk of the passage of patrol guards through the wildlife reserve and maximize the probability of penetration detection.

To model the possible paths to serve as an optimal passage for the patrol guards from a source *chauki* to a destination *chauki* within the wildlife reserve, we first identify a set of vulnerability factors, which may be anthropogenic or natural, and each of which may either promote or constrain the passage of the patrol guards through the wildlife reserve to various degrees, and hence become the major determinants in the structural optimization of patrol paths. The wildlife reserve vulnerability factors that we take into consideration are:

1. Electric poles – presence or absence
2. Water base – percentage presence or absence
3. Prey base – percentage presence or absence
4. Anthropogenic Disturbances
   i) Agricultural Land – presence or absence
   ii) Forest Land (*usability)* – percentage presence or absence
   iii) Encroachment - presence or absence
   iv) Roadways/Railways – presence or absence
5. Previous Disturbance sightings
   i) Camp fires – presence or absence
   ii) Poacher's Sighting – presence or absence
   iii) Electrocution Sightings – presence or absence
   iv) Carcass Poisoning Sightings –presence or absence
   v) Human Animal Conflict Sights –presence or absence
6. Grassland – percentage presence or absence
7. Forest types
   i) Open – percentage presence or absence
   ii) Dense – percentage presence or absence
   iii) Moderately Dense – percentage presence or absence

We assume that the patrol guards in the wildlife reserve (Θ_1) and the adversary combined with each of the above mentioned vulnerability factors (Θ_2) constitute the two rational agents that play a mixed strategy Hawk and Dove game G iterated over a number of generations. The players may use a number of strategies in the game in order to optimize their payoff. These payoffs are the costs incurred by the patrol guards while patrolling the reserve in order of law enforcement and conservation.

Next we code the different patrol chaukis in the wildlife reserve, based on their GPS coordinates and the grid number of the grid in which the corresponding *chauki* lies, a sample of the coding is shown in Table 1:

Table. 8.1 Coding for the Patrol *chaukis* in the wildlife reserve:

S. No	Patrol *chauki*	Latitude	Longitude	Grid Number	Code
1.	Patrol *chauki* 1	38°53′23″N	77°00′27″W	2251	2251-53-00-1
2.	Patrol *chauki* 2	38°63′23″N	77°01′27″W	2278	2278-63-01-2
3.	Patrol *chauki* 3	38°73′23″N	77°02′27″W	2256	2256-73-02-3
4.	Patrol *chauki* 4	38°83′23″N	77°03′27″W	2241	2241-83-03-4
5.	Patrol *chauki* 5	38°93′23″N	77°04′27″W	2298	2298-93-04-5
6.	Patrol *chauki* 6	39°03′23″N	77°05′27″W	2266	2266-03-05-6
7.	Patrol *chauki* 7	39°13′23″N	77°06′27″W	2387	2387-13-06-7
8.	Patrol *chauki* 8	39°23′23″N	77°07′27″W	2354	2354-23-07-8
9.	Patrol *chauki* 9	39°33′23″N	77°08′27″W	2376	2376-33-08-9
10.	Patrol *chauki* 10	39°43′23″N	77°09′27″W	2309	2309-43-09-10
11.	Patrol *chauki* 11	39°53′23″N	77°10′27″W	2498	2498-53-10-11
12.	Patrol *chauki* 12	39°63′23″N	77°11′27″W	2465	2465-63-11-12
13.	Patrol *chauki* 13	39°73′23″N	77°12′27″W	2453	2453-73-12-13
14.	Patrol *chauki* 14	39°83′23″N	77°13′27″W	2477	2477-83-13-14
15.	Patrol *chauki* 15	39°93′23″N	77°14′27″W	2567	2567-93-14-15
16.	Patrol *chauki* 16	40°03′23″N	77°15′27″W	2543	2543-03-15-16
17.	Patrol *chauki* 17	40°13′23″N	77°16′27″W	2599	2599-13-16-17
18.	Patrol *chauki* 18	40°23′23″N	77°17′27″W	2512	2512-23-17-18
19.	Patrol *chauki* 19	40°33′23″N	77°18′27″W	2609	2609-33-18-19
20.	Patrol *chauki* 20	40°43′23″N	77°19′27″W	2687	2687-43-19-20

We next compute the costs incurred by the patrol guards in traversing the patrol pathsthrough each grid in the given wildlife reserve. With each grid, we associate a numeric weight *c*, thus rendering Γ a weighted graph. We designate the weight assigned to a grid as the cost incurred by the patrol guard for passage through that grid, and define this cost function as the mapping

$$c : E \to \aleph$$
$$\ni e \mapsto c(e) = r \in \aleph, \forall e \in E, \aleph = \{0,1,\ldots\}.$$

We assume that the cost of travelling through a grid is a numeric proxy for the perceived (by the patrol guard) penetration detection or (even physical) risk associated with the grid, and hence to the probability of penetration detectionobserved by the patrol guard in traversing that grid. We further assume that the probability of penetration detection being essentially and only based on the presence or the absence of even one or all, of the above mentioned vulnerability factors.

The costs to each of the possible grid is assigned taking into consideration the possible kind of vulnerability factors mentioned in the foregoing, that a traversing patrol guard is likely to encounter while negotiating that grid. The payoff matrix for the game G is constructed based on these costs. One of the prime objectives in designing intelligent patrolling path would be to minimize the risk(mortality or physical injury) and maximizing the probability of penetration detection, we describe the research problem as: *Given an undirected, connected landscape* $\Gamma(V, E, \psi_\Gamma)$, *an index set* $I = \{0,1,\ldots,n\} \subseteq \aleph$, $\ni v_i \in V, e_i \in E$ *with* $i \in I$, *the cost function* $c_i = c(e_i) \forall i \in I$, *compute a spanning treeTor the Hamiltonian circuit H such that* $\sum_{i \in N} c_i$ *is minimum.*

Therefore, the objective of our work is to compute a spanning tree or the Hamiltonian circuit for the given condition, such that the sum total of the costs incurred by the patrol guards in its passage between the *chaukis* embedded in the given wildlife reserve, through the landscape matrix, is minimized.

One of the most commonly used solution procedure to address the research problem is the Boruvka-Kruskal algorithm (Kruskal's algorithm) (Boruvka 1926; Kruskal 1956, 1997). Kruskal's algorithm is a tree-search algorithm that accepts as input a weighted connected graph, and returns as output an optimal spanning tree. The execution of the Algorithm starts with $|V|$ isolated trees in the forest (a set of trees, and hence essentially an acyclic graph), each initially with 1 vertex. The Algorithm then constructs a spanning tree edge-by-edge, by making a decision to select the least cost path that connects two trees, to return a single tree in the forest. At the termination of the Algorithm, the forest has only 1 component, namely, the output spanning tree. Being a greedy algorithm, Kruskal's algorithm makes a 'greedy' (locally optimal) decision at each stage of its run, without being concerned about the impact of this decision on the global optimality of the output.

A major advantage of using Kruskal's algorithm for solving our defined problem is that the Algorithm has a linear time complexity, given by $O(|E|\log|E|)$. Additionally, for Kruskal's Algorithm, the following theorem guarantees the optimality of the output spanning tree:

Theorem 15 *Every Bourvka-Kruskal tree is an optimal tree* (Bondy and Murty 2008).

In computing the payoff matrix, we further assume that the players involved in this game choose to play both mixed and pure strategies. The reasoning for various vulnerability factors that we consider as impacts on patrol paths in the wildlife reserve, and their corresponding cost assignments and subsequent payoff evaluations are as below:

1. **Electric poles**: Electrocution is one of the major poaching practices in wildlife reserves in India. The presence of electric poles in a grid of resolution 5 X 5 or 2 X 2 makes the complete grid vulnerable for electrocution and also the adjacent grids which must be patrolled on every movement to check for the adversary detection. So as the complete grid may get effected and support the adversary so we consider only presence and absence and no membership of this factor in the grid. So, as a dominant factor, the presence of electric poles in a grid makes the adversary to behave as a HAWK and the patrol guards as DOVE for the hawk and dove game and thus provides a score of -5 to the grid.
2. **Water Base**: Poisoning of the water bodies present inside the wildlife reserve like lakes, ponds, percolation pits, etc. is done by the poachers in order to kill the animals as they always visit these water bodies to drink water. As the poisoning of the water bodies can only take place in the regions where water would be available, so we check the percentage cover of a water body in the grid. So if a water body covers 30% of the area of a grid then the membership function of the factor μ_w= 0.3 and as a vulnerable factor, the score of water base in the particular grid = μ_wX -5 = 0.3 X -5 = -1.5.

Similar to the above two explained factors, the scores calculated for the other vulnerability factors, which have been discussed is shown in the following table:

Table. 8.2 Scores contributed by each vulnerability factor to the grid through presence or membership value:

Factor	Sub-Factors	Factor Code	Membership Function Value	Hawk-Dove Score	Score Provided To The Grid
Electric Pole		A	1	-5	- 5
Water Base		B	μ_w	-3	- μ_wX 3
Prey Base		C	μ_p	-5	- μ_pX 5
Anthropogenic Disturbances	Agricultural Land	D	μ_a	-5	- μ_aX 5
	Forest Land (*Usability*)	E	μ_f	-3	- μ_fX 3
	Encroachment	F	1	-5	- 5
	Roadways / Railways	G	μ_r(depends on the distance from the grid and decreases uniformly as 0.1 with every grid layer distance)	-5	- μ_wX 5
Previous Disturbance Sightings	Camp Fires	H	1	-5	- 5
	Poacher Sightings	I	1	-5	- 5
	Electrocution Sightings	J	1	-5	- 5
	Carcass Poisoning Sightings	K	1	-5	- 5
	Human Animal Conflict Sights	L	1	-5	- 5
Grassland		M	μ_g	-5	- μ_gX5
Forest Types	Open	N	μ_o	-3	- μ_oX 3

	Dense	O	μ_d	-5	$- \mu_d X 5$
	Moderately Dense	P	μ_m	-4	$- \mu_m X 4$

Based on the above criteria of scoring, the various factors with respect to the forest guards using the strategy pair of Hawk and Dove game the following cost matrix is obtained, with scores for the grids entered in the matrix based on Figure. 18, Figure 19, Table 3.

1110	1115	1120	1125	1130	1135	1140	1145	1150	1155	1160	1165	1170	1175	1180	1185	1190	1195	1200
2110	2115	2120	2125	2130	2135	2140	2145	2150	2155	2160	2165	2170	2175	2180	2185	2190	2195	2200
3110	3115	3120	3125	3130	3135	3140	3145	3150	3155	3160	3165	3170	3175	3180	3185	3190	3195	3200
4110	4115	4120	4125	4130	4135	4140	4145	4150	4155	4160	4165	4170	4175	4180	4185	4190	4195	4200
5110	5115	5120	5125	5130	5135	5140	5145	5150	5155	5160	5165	5170	5175	5180	5185	5190	5195	5200
6110	6115	6120	6125	6130	6135	6140	6145	6150	6155	6160	6165	6170	6175	6180	6185	6190	6195	6200
7110	7115	7120	7125	7130	7135	7140	7145	7150	7155	7160	7165	7170	7175	7180	7185	7190	7195	7200
8110	8115	8120	8125	8130	8135	8140	8145	8150	8155	8160	8165	8170	8175	8180	8185	8190	8195	8200
9110	9115	9120	9125	9130	9135	9140	9145	9150	9155	9160	9165	9170	9175	9180	9185	9190	9195	9200
10110	10115	10120	10125	10130	10135	10140	10145	10150	10155	10160	10165	10170	10175	10180	10185	10190	10195	10200
11110	11115	11120	11125	11130	11135	11140	11145	11150	11155	11160	11165	11170	11175	11180	11185	11190	11195	11200
12110	12115	12120	12125	12130	12135	12140	12145	12150	12155	12160	12165	12170	12175	12180	12185	12190	12195	12200
13110	13115	13120	13125	13130	13135	13140	13145	13150	13155	13160	13165	13170	13175	13180	13185	13190	13195	13200
14110	14115	14120	14125	14130	14135	14140	14145	14150	14155	14160	14165	14170	14175	14180	14185	14190	14195	14200
15110	15115	15120	15125	15130	15135	15140	15145	15150	15155	15160	15165	15170	15175	15180	15185	15190	15195	15200
16110	16115	16120	16125	16130	16135	16140	16145	16150	16155	16160	16165	16170	16175	16180	16185	16190	16195	16200
17110	17115	17120	17125	17130	17135	17140	17145	17150	17155	17160	17165	17170	17175	17180	17185	17190	17195	17200
18110	18115	18120	18125	18130	18135	18140	18145	18150	18155	18160	18165	18170	18175	18180	18185	18190	18195	18200
19110	19115	19120	19125	19130	19135	19140	19145	19150	19155	19160	19165	19170	19175	19180	19185	19190	19195	19200
20110	20115	20120	20125	20130	20135	20140	20145	20150	20155	20160	20165	20170	20175	20180	20185	20190	20195	20200
21110	21115	21120	21125	21130	21135	21140	21145	21150	21155	21160	21165	21170	21175	21180	21185	21190	21195	21200
22110	22115	22120	22125	22130	22135	22140	22145	22150	22155	22160	22165	22170	22175	22180	22185	22190	22195	22200
23110	23115	23120	23125	23130	23135	23140	23145	23150	23155	23160	23165	23170	23175	23180	23185	23190	23195	23200
24110	24115	24120	24125	24130	24135	24140	24145	24150	24155	24160	24165	24170	24175	24180	24185	24190	24195	24200
25110	25115	25120	25125	25130	25135	25140	25145	25150	25155	25160	25165	25170	25175	25180	25185	25190	25195	25200
26110	26115	26120	26125	26130	26135	26140	26145	26150	26155	26160	26165	26170	26175	26180	26185	26190	26195	26200

Figure. 8.2 Hypothetical wildlife reserve (black curve in the grid pattern) showing patrol *chaukis* (red grids), and the matrix (grid pattern with particular grid numbers)

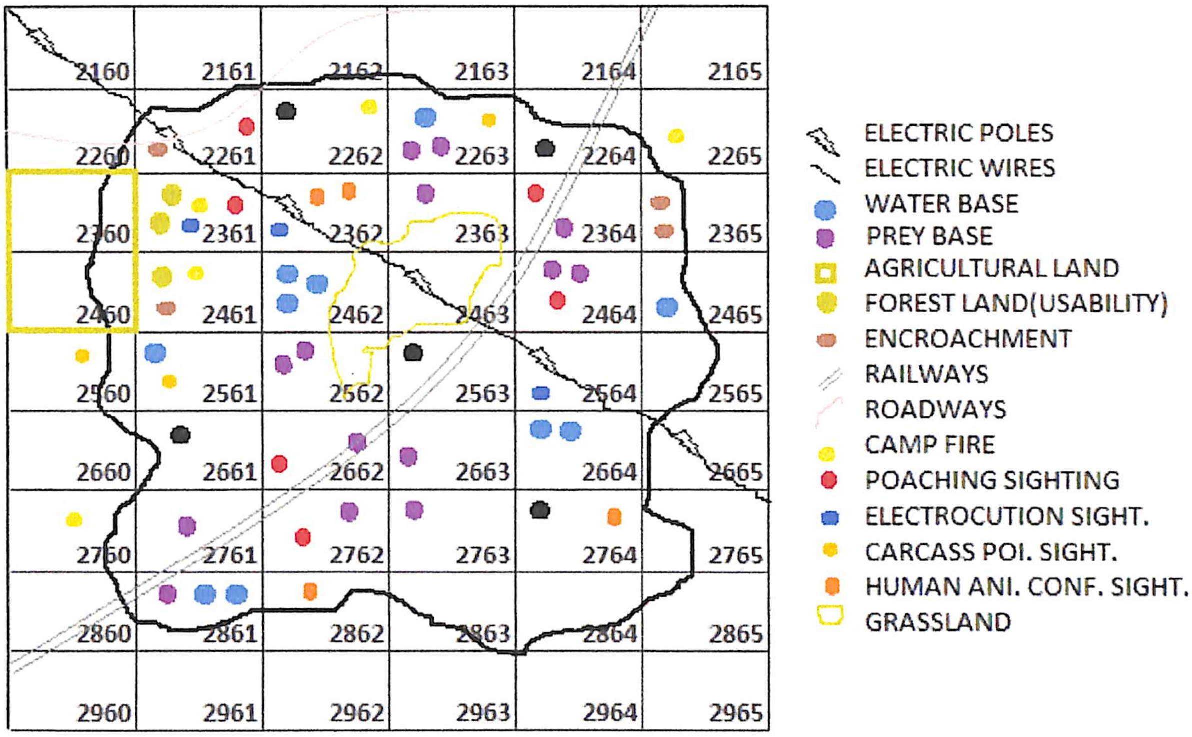


Figure. 8.3 Hypothetical wildlife reserve (black curve in the grid pattern) showing patrol *chaukis* (black spots), and the matrix (grid pattern with particular grid numbers) with all the factors represented in the map.

Table. 8.3 Percentage cover of each considered forest type in each grid of the considered landscape shown in Figure. 8.3:

-	-	FOREST TYPE %age COVER	-
GRID NUMBER	OPEN FOREST	MODERATELY DENSE FOREST	DENSE FOREST
2160	82.945	2.055	0
2161	78.624	6.376	0
2162	79.433	5.567	0
2163	81.234	3.766	0
2164	76.443	8.557	0
2165	89.543	4.543	0
2260	79.267	5.733	0
2261	64.344	20.656	0
2262	57.789	27.211	0
2263	34.567	50.433	28.1299
2264	56.986	28.014	0
2265	92.765	7.765	0
2360	5.34	79.66	0

2361	67.54	17.46	0
2362	12.45	62.55	34.765
2363	17.765	37.235	33.1705
2364	23.765	61.235	31.3705
2365	89.665	4.665	0
2460	4.311	80.689	0
2461	64.543	20.457	0
2462	9.13	0	4.261
2463	12.34	0	33.798
2464	11.24	73.76	35.128
2465	69.398	15.602	0
2560	65.654	19.346	0
2561	36.543	48.457	27.5371
2562	12.234	72.766	0
2563	16.432	68.568	33.5704
2564	5.34	79.66	36.898
2565	67.432	17.568	0
2660	79.432	5.568	0
2661	23.234	61.766	31.5298
2662	13.123	71.877	34.5631
2663	2.234	82.766	37.8298
2664	3.456	31.544	37.4632
2665	64.654	20.346	0
2760	93.567	8.567	0
2761	34.654	50.346	28.1038
2762	23.432	61.568	31.4704
2763	11.234	73.766	35.1298
2764	6.321	78.679	36.6037
2765	82.432	2.568	0
2860	25.567	9.433	0
2861	31.123	53.877	29.1631
2862	23.432	61.568	31.4704
2863	12.345	72.655	34.7965
2864	7.432	77.568	36.2704
2865	68.934	16.066	0
2960	71.432	13.568	0
2961	87.322	2.322	0
2962	74.876	10.124	0
2963	73.744	11.256	0
2964	79.324	5.676	0
2965	84.432	0.568	0

Table. 8.4 Membership value of each factor to be considered in each grid of the sample wildlife reserve considered in Figure. 8.3:

GRID#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2160	1	0	0	0	0	1	0.6	0	0	0	0	0	0	0.82945	0	0.02055
2161	1	0	0	0	0	0	0.6	0	0	0	0	0	0	0.78624	0	0.06376
2162	0	0	0	0	0	0	1	0	0	0	0	0	0	0.79433	0	0.05567
2163	0	0	0	0	0	0	1	0	0	0	0	0	0	0.81234	0	0.03766
2164	0	0	0	0	0	0	1	0	0	0	0	0	0	0.76443	0	0.08557
2165	0	0	0	0	0	0	1	0	0	0	0	0	0	0.89543	0	0.04543
2260	1	0	0	0.022	0	0	1	0	0	0	0	0	0	0.79267	0	0.05733
2261	1	0	0	0.03	0	1	1	0	1	0	0	0	0	0.64344	0	0.20656
2262	1	0	0	0	0.01	0	0.6	1	0	0	0	0	0	0.57789	0	0.27211
2263	0	0.1412	0.2745	0	0	0	0.6	0	0	0	1	0	0	0.34567	0.281299	0.50433
2264	0	0	0	0	0	0	1	0	0	0	0	0	0	0.56986	0	0.28014
2265	0	0	0	0	0	1	0.6	1	0	0	0	0	0	0.92765	0	0.07765
2360	0	0	0	1	0	0	0.6	0	0	0	0	0	0	0.0534	0	0.7966
2361	1	0	0	0.01	0.23	0	0.6	1	1	1	0	0	0	0.6754	0	0.1746
2362	1	0	0	0	0	0	0.6	0	0	1	0	1	0.1	0.1245	0.34765	0.6255
2363	1	0	0.1123	0	0	0	0.6	0	0	0	0	0	0.3	0.17765	0.331705	0.37235
2364	0	0	0.0932	0	0	0	1	0	1	0	0	0	0	0.23765	0.313705	0.61235
2365	0	0	0	0	0	1	0.6	1		0	0	0	0	0.89665	0	0.04665
2460	0	0	0	1	0	0	0.2	0	0	0	0	0	0	0.04311	0	0.80689
2461	0	0	0	0.02	0.31	1	0.2	1	0	0	0	0	0	0.64543	0	0.20457
246	1	0.412	0	0	0	0	0.	0	0	0	0	0	0.45	0.0913	0.0426	0

2		6					6								1	
2463	1	0	0	0	0	0	1	0	0	0	0	0	0.75	0.1234	0.31798	0
2464	1	0	0.2879	0	0	0	1	0	1	0	0	0	0	0.1124	0.35128	0.7376
2465	0	0.14	0	0	0	1	0.6	0	0	0	0	0	0	0.69398	0	0.15602
2560	0	0	0	0.02	0	0	0	0	0	0	1	0	0	0.65654	0	0.19346
2561	0	0.14	0	0.08	0	0	0.2	0	0	0	1	0	0	0.36543	0.275371	0.48457
2562	0	0	0.2345	0	0	0	0.6	0	0	0	0	0	0.348298	0.12234	0	0.72766
2563	1	0	0	0	0	0	1	0	0	0	0	0	0	0.16432	0.335704	0.68568
2564	1	0	0	0	0	0	0.6	0	0	1	0	0	0	0.0534	0.36898	0.7966
2565	1	0	0	0	0	0	0.2	0	0	0	0	0	0	0.67432	0	0.17568
2660	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0.79432	0	0.05568
2661	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0.23234	0.315298	0.61766
2662	0	0	0.1123	0	0	0	1	0	1	0	0	0	0	0.13123	0.345631	0.71877
2663	0	0	0.1123	0	0	0	1	0	0	0	0	0	0	0.02234	0.378298	0.82766
2664	1	0.33	0	0	0	0	0.6	0	0	0	0	0	0	0.03456	0.374632	0.31544
2665	1	0	0	0	0	0	0.2	0	0	0	0	0	0	0.64654	0	0.20346
2760	0	0	0	0	0	0	0.6	1	0	0	0	0	0	0.93567	0	0.08567
2761	0	0	0.1123	0	0	0	1	0	0	0	0	0	0	0.34654	0.281038	0.50346
2762	0	0	0.1123	0	0	0	1	0	1	0	0	0	0	0.23432	0.314704	0.61568
2763	0	0	0.1123	0	0	0	0.6	0	0	0	0	0	0	0.11234	0.351298	0.73766
2764	0	0	0	0	0	0	0.2	0	0	0	0	1		0.06321	0.366037	0.78679
2765	1	0	0	0	0	0	0	0	0	0	0	0	0	0.82432	0	0.02568
2860	0	0.37	0	0	0	0	1	0	0	0	0	0	0	0.25567	0	0.09433
2861	0	0	0.1123	0	0	0	1	0	0	0	0	0	0	0.31123	0.291631	0.53877

2862	0	0	0	0	0	0	0.6	0	0	0	0	1	0	0.23432	0.314704	0.61568
2863	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0.12345	0.347965	0.72655
2864	0	0	0	0	0	0	0	0	0	0	0	0	0	0.07432	0.362704	0.77568
2865	0	0	0	0	0	0	0	0	0	0	0	0	0	0.68934	0	0.16066
2960	0	0	0	0	0	0	1	0	0	0	0	0	0	0.71432	0	0.13568
2961	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0.87322	0	0.02322
2962	0	0	0	0	0	0	0	0	0	0	0	0	0	0.74876	0	0.10124
2963	0	0	0	0	0	0	0	0	0	0	0	0	0	0.73744	0	0.11256
2964	0	0	0	0	0	0	0	0	0	0	0	0	0	0.79324	0	0.05676
2965	0	0	0	0	0	0	0	0	0	0	0	0	0	0.84432	0	0.00568

Table. 8.5 Cost of each factor to be considered in each grid of the sample wildlife reserve considered in Figure. 8.3 based on Table. 8.4 membership values and Table. 8.2 cost constraints:

GRID #	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2160	-5	0	0	0	0	-5	-3	0	0	0	0	0	0	-2.5	0	-0.082
2161	-5	0	0	0	0	0	-3	0	0	0	0	0	0	-2.4	0	-0.255
2162	0	0	0	0	0	0	-5	0	0	0	0	0	0	-2.4	0	-0.223
2163	0	0	0	0	0	0	-5	0	0	0	0	0	0	-2.4	0	-0.151
2164	0	0	0	0	0	0	-5	0	0	0	0	0	0	-2.3	0	-0.342
2165	0	0	0	0	0	0	-5	0	0	0	0	0	0	-2.7	0	-0.182
2260	-5	0	0	-0.11	0	0	-5	0	0	0	0	0	0	-2.4	0	-0.229
2261	-5	0	0	-0.15	0	-5	-5	0	-5	0	0	0	0	-1.9	0	-0.826
2262	-5	0	0	0	-0.03	0	-3	-5	0	0	0	0	0	-1.7	0	-1.088
2263	0	-0.424	-1.373	0	0	0	-3	0	0	0	-5	0	0	-1	-1.4	-2.017

2264	0	0	0	0	0	0	-5	0	0	0	0	0	0	-1.7	0	-1.121
2265	0	0	0	0	0	-5	-3	-5	0	0	0	0	0	-2.8	0	-0.311
2360	0	0	0	-5	0	0	-3	0	0	0	0	0	0	-0.2	0	-3.186
2361	-5	0	0	-0.05	-0.69	0	-3	-5	-5	-5	0	0	0	-2	0	-0.698
2362	-5	0	0	0	0	0	-3	0	0	-5	0	-5	-0.5	-0.4	-1.7	-2.502
2363	-5	0	-0.562	0	0	0	-3	0	0	0	0	0	-1.5	-0.5	-1.7	-1.489
2364	0	0	-0.466	0	0	0	-5	0	-5	0	0	0	0	-0.7	-1.6	-2.449
2365	0	0	0	0	0	-5	-3	-5	0	0	0	0	0	-2.7	0	-0.187
2460	0	0	0	-5	0	0	-1	0	0	0	0	0	0	-0.1	0	-3.228
2461	0	0	0	-0.1	-0.93	-5	-1	-5	0	0	0	0	0	-1.9	0	-0.818
2462	-5	-1.238	0	0	0	0	-3	0	0	0	0	0	-2.3	-0.3	-0.2	0
2463	-5	0	0	0	0	0	-5	0	0	0	0	0	-3.8	-0.4	-1.6	0
2464	-5	0	-1.44	0	0	0	-5	0	-5	0	0	0	0	-0.3	-1.8	-2.95
2465	0	-0.42	0	0	0	-5	-3	0	0	0	0	0	0	-2.1	0	-0.624
2560	0	0	0	-0.1	0	0	0	0	0	0	-5	0	0	-2	0	-0.774
2561	0	-0.42	0	-0.4	0	0	-1	0	0	0	-5	0	0	-1.1	-1.4	-1.938
2562	0	0	-1.173	0	0	0	-3	0	0	0	0	0	-1.7	-0.4	0	-2.911
2563	-5	0	0	0	0	0	-5	0	0	0	0	0	0	-0.5	-1.7	-2.743
2564	-5	0	0	0	0	0	-3	0	0	-5	0	0	0	-0.2	-1.8	-3.186
2565	-5	0	0	0	0	0	-1	0	0	0	0	0	0	-2	0	-0.703
2660	0	0	0	0	0	0	-1	0	0	0	0	0	0	-2.4	0	-0.223
2661	0	0	0	0	0	0	-3	0	0	0	0	0	0	-0.7	-1.6	-2.471
2662	0	0	-0.562	0	0	0	-5	0	-5	0	0	0	0	-0.4	-1.7	-2.875
2663	0	0	-	0	0	0	-	0	0	0	0	0	0	-	-	-

			0.562				5							0.1	1.9	3.311
2664	-5	-0.99	0	0	0	0	-3	0	0	0	0	0	0	-0.1	-1.9	-1.262
2665	-5	0	0	0	0	0	-1	0	0	0	0	0	0	-1.9	0	-0.814
2760	0	0	0	0	0	0	-3	-5	0	0	0	0	0	-2.8	0	-0.343
2761	0	0	-0.562	0	0	0	-5	0	0	0	0	0	0	-1	-1.4	-2.014
2762	0	0	-0.562	0	0	0	-5	0	-5	0	0	0	0	-0.7	-1.6	-2.463
2763	0	0	-0.562	0	0	0	-3	0	0	0	0	0	0	-0.3	-1.8	-2.951
2764	0	0	0	0	0	0	-1	0	0	0	0	-5	0	-0.2	-1.8	-3.147
2765	-5	0	0	0	0	0	0	0	0	0	0	0	0	-2.5	0	-0.103
2860	0	-1.11	0	0	0	0	-5	0	0	0	0	0	0	-0.8	0	-0.377
2861	0	0	-0.562	0	0	0	-5	0	0	0	0	0	0	-0.9	-1.5	-2.155
2862	0	0	0	0	0	0	-3	0	0	0	0	-5	0	-0.7	-1.6	-2.463
2863	0	0	0	0	0	0	-1	0	0	0	0	0	0	-0.4	-1.7	-2.906
2864	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.2	-1.8	-3.103
2865	0	0	0	0	0	0	0	0	0	0	0	0	0	-2.1	0	-0.643
2960	0	0	0	0	0	0	-5	0	0	0	0	0	0	-2.1	0	-0.543
2961	0	0	0	0	0	0	-1	0	0	0	0	0	0	-2.6	0	-0.093
2962	0	0	0	0	0	0	0	0	0	0	0	0	0	-2.2	0	-0.405
2963	0	0	0	0	0	0	0	0	0	0	0	0	0	-2.2	0	-0.45
2964	0	0	0	0	0	0	0	0	0	0	0	0	0	-2.4	0	-0.227
2965	0	0	0	0	0	0	0	0	0	0	0	0	0	-2.5	0	-0.023

Table. 8.6 Cumulative Cost to be considered in each grid and the RANK of each grid for concern based on the costs for the sample wildlife reserve considered in Figure. 8.3 based on Table. 8.4. Table 8.5 and Table. 8.2 cost constraints:

GRID #	**COST OF GRID**	**RANK OF GRID ACCORDING TO CONCERNING COSTS**
2160	-15.57055	10
2161	-10.61376	28
2162	-7.60567	42
2163	-7.58766	43
2164	-7.63557	41
2165	-7.86801	36
2260	-12.71733	19
2261	-22.90656	3
2262	-15.85211	8
2263	-14.256925	16
2264	-7.83014	38
2265	-16.09355	6
2360	-11.3466	22
2361	-26.4646	1
2362	-23.11375	2
2363	-13.742375	17
2364	-15.196875	13
2365	-15.87655	7
2460	-9.35689	31
2461	-14.78457	15
2462	-11.97475	21
2463	-15.7101	9
2464	-21.4835	4
2465	-11.12602	26
2560	-7.84346	37
2561	-11.231425	23
2562	-9.19165	32
2563	-14.9142	14
2564	-18.1915	5
2565	-8.72568	34
2660	-3.60568	49
2661	-7.74415	39
2662	-15.558425	11
2663	-10.83065	27
2664	-12.2286	20
2665	-8.75346	33

2760	-11.14969	25
2761	-10.02015	30
2762	-15.3007	12
2763	-8.60565	35
2764	-11.166975	24
2765	-7.57568	44
2860	-7.25433	45
2861	-10.108425	29
2862	-12.7392	18
2863	-6.016375	46
2864	-5.1392	47
2865	-2.71066	50
2960	-7.68568	40
2961	-3.71254	48
2962	-2.65124	52
2963	-2.66256	51
2964	-2.60676	53
2965	-2.55568	54

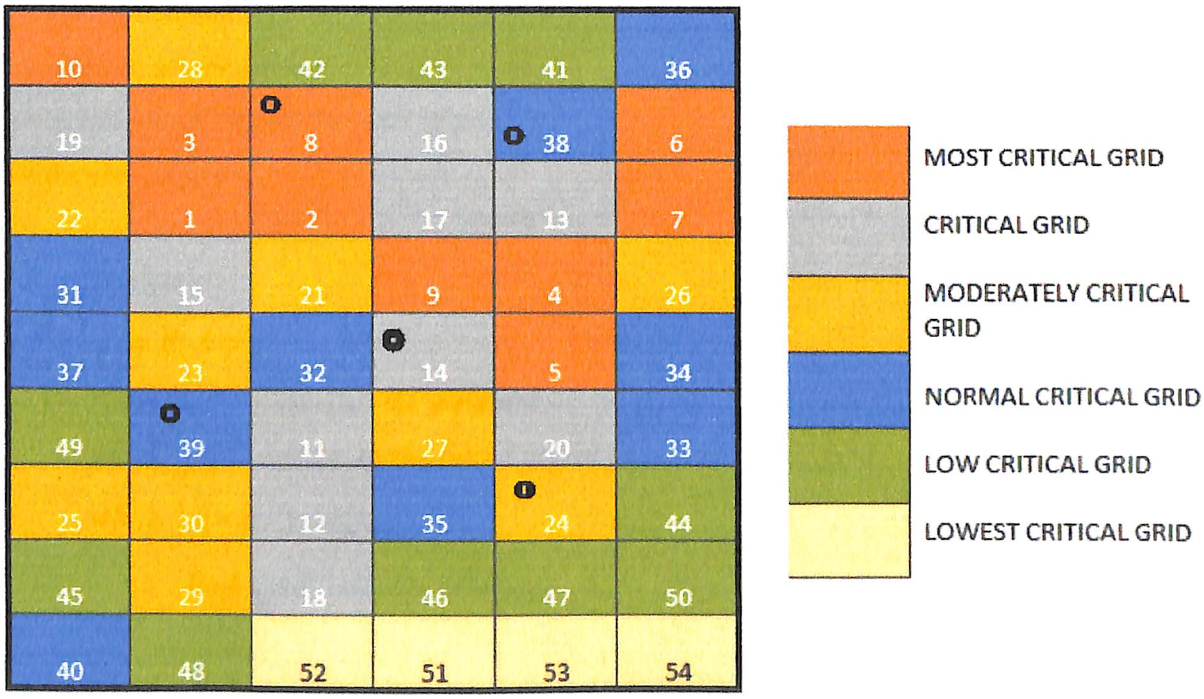


Figure. 8.4Criticality of each grid for surveillance for law enforcement in the sample wildlife reserve of Figure.

Table. 8.7 Coding for the Patrol *chaukis* in the sample wildlife reserve shown in Figure. 8.3:

S. No	Patrol *chauki*	Latitude	Longitude	Grid Number	Code
1.	Patrol *chauki* 1	38°53′23″N	77°00′27″W	2262	2262-53-00-1
2.	Patrol *chauki* 2	38°63′23″N	77°01′27″W	2264	2264-63-01-2
3.	Patrol *chauki* 3	38°73′23″N	77°02′27″W	2563	2563-73-02-3
4.	Patrol *chauki* 4	38°83′23″N	77°03′27″W	2661	2661-83-03-4
5.	Patrol *chauki* 5	38°93′23″N	77°04′27″W	2764	2764-93-04-5

The pseudo code for Kruskal's algorithm for generating the minimum spanning tree if the source and sink Patrol *chaukis* are different is as below:

Procedure Kruskal (Γ, c)

START

DECLARE

E = *set of Grids*

A = *Patrol chaukis*

F = *storage set for grids*

a = *initial patrol chauki for start of patrol*

n = *number of grids*

e = *grids for movement*

BEGIN

$F := E; A = \phi$

Set initial e = *min (all the weights)*

Set the grid containing the initial patrol chauki as initial grid

while $|A| < n-1$ *loop*

find $e \in F \ni c(e)$ *is minimum*

$F := F - \{e\}$

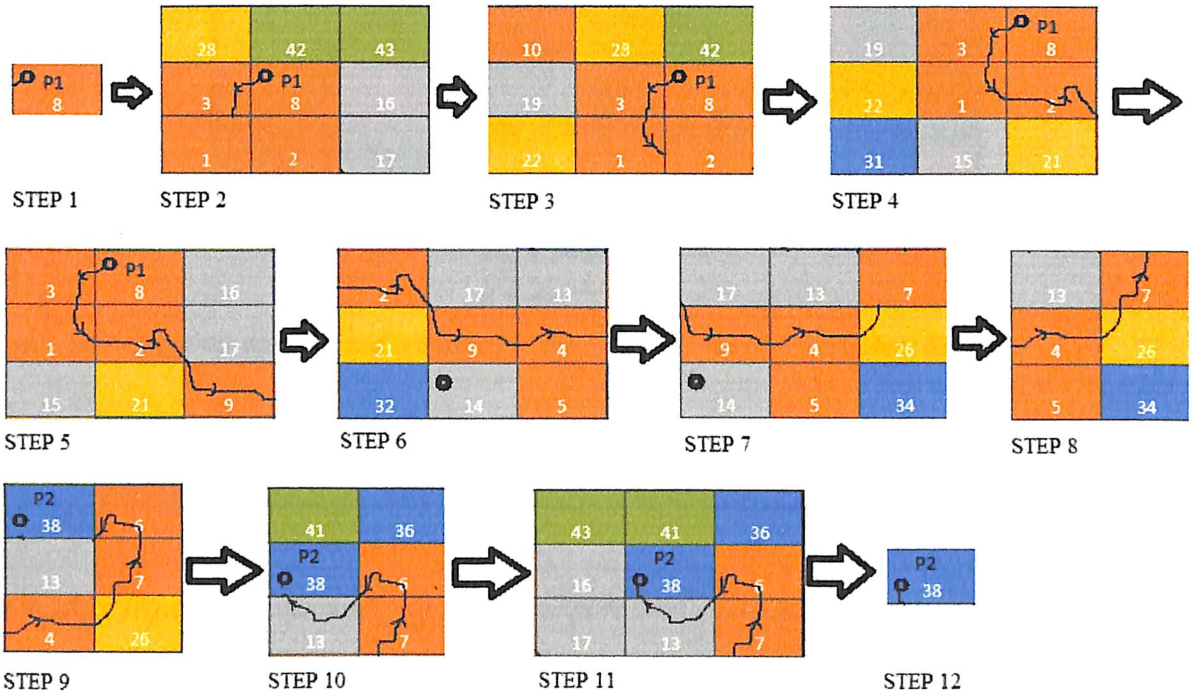
if $\Gamma(A \cup \{e\})$ *acyclic then*

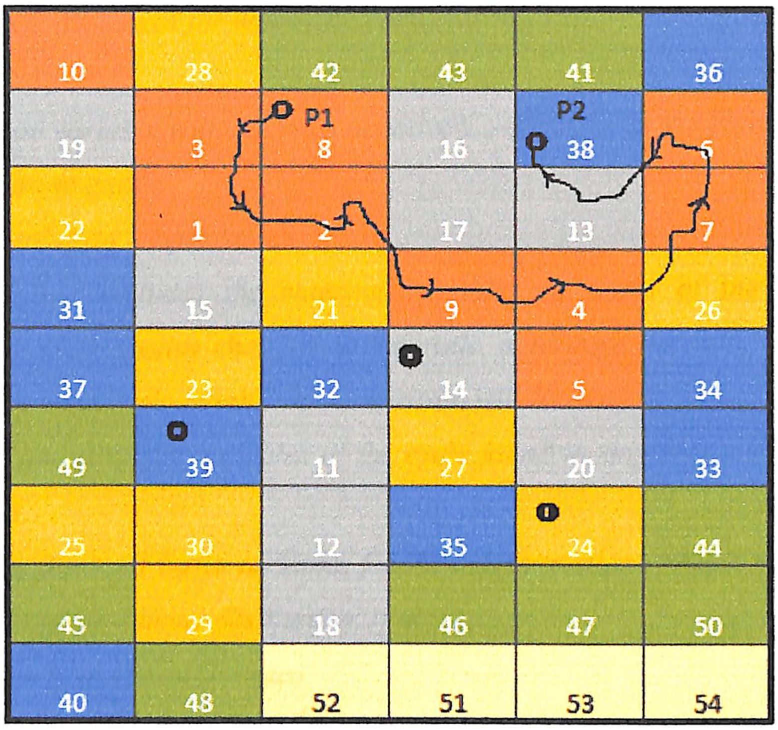
$A := A \cup \{e\};$

end if; end loop $\Gamma(A)$ *is a minimum spanning tree end Kruskal; END*

A minimum spanning tree (MST) for the focal complex, obtained on using Kruskal's algorithm, with its optimality guaranteed by Theorem 15, is shown in Figure. 21:

Let us assume that a patrol guard starts from the *chauki* 2262-53-00-1 and aims to move till the *chauki*2264-63-01-2. The path for the patrol guard is decided based on the pseudo code mentioned above i.e. *Procedure Kruskal*. The steps for patrolling are decided by the algorithm 1, algorithm 2 and algorithm 3 of prediction of probability of penetration detection. Suppose the guard enters the distance to be moved by the guard as 14 kilometers. Therefore, considering the criticality levels, grid resolution of 2 X 2 and the distance to be travelled in the grid, the patrol path is decided and presented to the guard for movement.





P1: 2262-53-00-1

P2: 2264-63-01-2

Figure. 8.5Path for the patrol guard is decided based on the pseudo code *Procedure Kruskal.*

The pseudo code for Travelling salesman problemusing transportation simplex method for generating the Hamiltonian circu*it if the source and sink Patrol chaukis are same is as below:*

Procedure travelling salesman problem (transportation simplex method)

START

DECLARE

U_i= *variable associated with the i-th supply constraint*

V_j= *variable associated with the j-th demand constraint*

$Z_{ij} = U_i - V_j$

C_{ij}=*cost of grid i. j.*

BEGIN

Find an initial basic feasible solution by some starting procedure. Then,

1. Set $U_1 = 0$. *Solve for the other* U_i*'s and* V_j*'s by:*

$C_{ij} - U_i + V_j = 0$ *for basic variables.*

Then calculate the $C_{ij}–Z_{ij}$ values for non-basic variables by:

$C_{ij} – Z_{ij} = C_{ij} – U_i + V_j$

Choose the non-basic variable with the most negative $C_{ij}–Z_{ij}$ value as the entering variable. If all $C_{ij}–Z_{ij}$ values are non-negative,

STOP; the current solution is optimal.

2. Find the cycle that includes the entering variable and some of the BASIC variables. Alternating positive and negative changes on the cycle, determine the "change amount" as the smallest allocation on the cycle at which a subtraction will be made.

3. Modify the allocations to the variables of the cycle found in step 2 by the "change amount" and return to step 1.

Note: there must be m + n - 1 basic variables for the transportation simplex method to work!

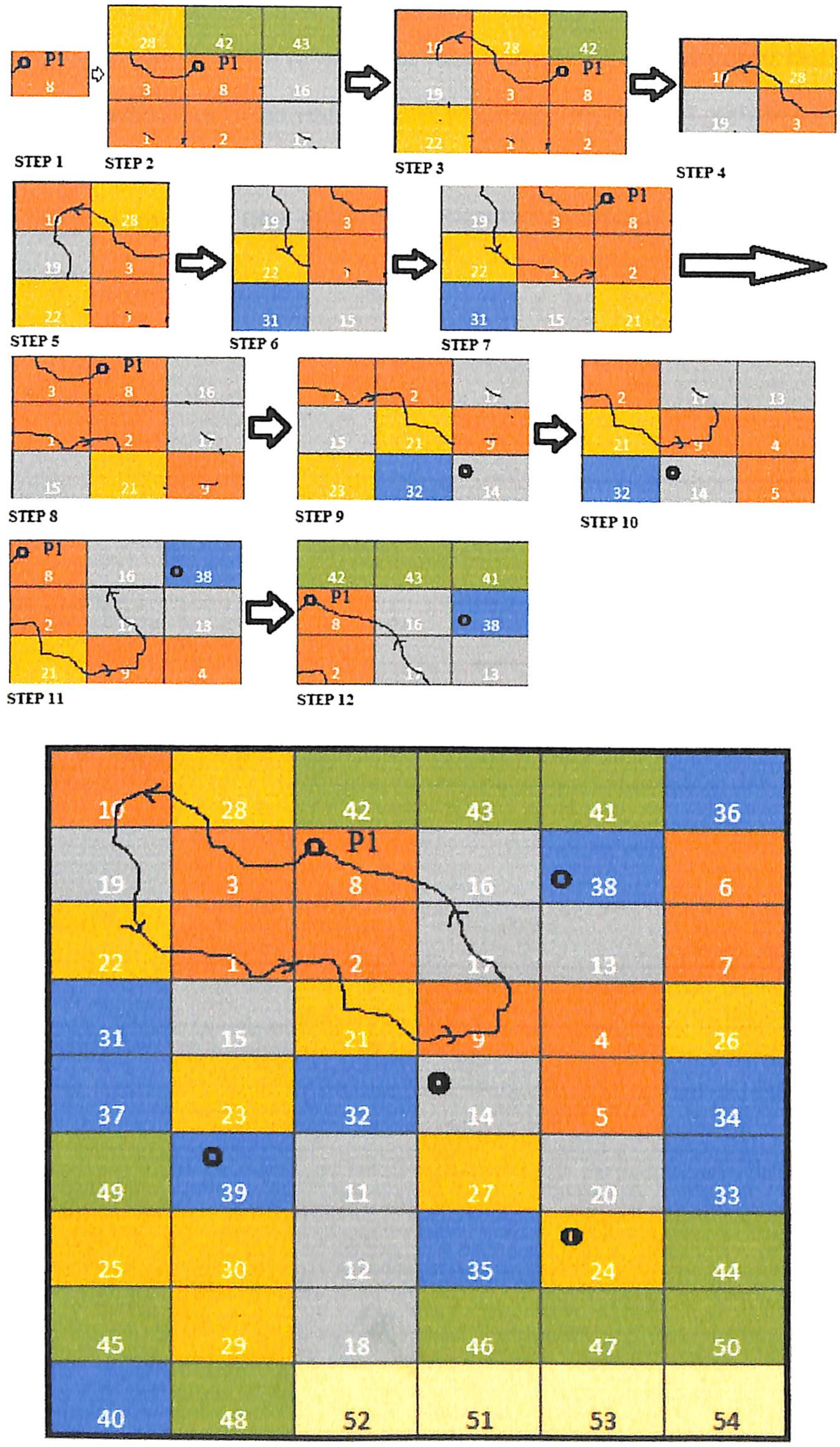
=> Add dummy source or dummy destination, if necessary

(m=# of sources and n=# of destinations)

END;

A Hamiltonian circuit for the focal complex, obtained on using Travelling salesman problem algorithm, shown in Figure. 22:

Let us assume that a patrol guard starts from the *chauki* 2262-53-00-1 and aims to return back to the *chauki* 2262-53-00-1. The path for the patrol guard is decided based on the pseudo code mentioned above i.e.*Procedure travelling salesman problem (transportation simplex method).* The steps for patrolling are decided by the algorithm 1, algorithm 2 and algorithm 3 of prediction of probability of penetration detection. Suppose the guard enters the distance to be moved by the guard as 14 kilometers. Therefore, considering the criticality levels, grid resolution of 2 X 2 and the distance to be travelled in the grid, the patrol path is decided and presented to the guard for movement.



P1: 2262-53-00-1

Figure. 8.6Path for the patrol guard is decided based on the pseudo code*Procedure travelling salesman problem (transportation simplex method).*

With every movement in the wildlife reserve for monitoring, the patrol guard is expected to note down all the different activities observed by him walking through the various grids and update the database accordingly. Once the database is updated the calculation of new costs is done automatically in the real time and new paths are generated accordingly.

9.ALGORITHM

Algorithm Intelligent_Patrolling_System

// input:

- *Grid resolution details (N X N)*
- *Factors to be considered:*
  - *Electric poles*
  - *Water base*
  - *Prey base*
  - *Anthropogenic disturbances*
  - *Previous Disturbance sightings*
  - *Grassland*
  - *Forest types.*
- *Membership value of each considered factor (μ_i)*
- *Distance to be travelled (d)*

// output: Intelligent Patrolling Path Description for Forest Patrol Guards.

Procedure FindFunc(d,t)

Create matrix M of size (2d + 1)(2d + 1), initialized with 0s

Fill out all entries in M as follows:

$M[2d + 1, 2d + 1] = 1$

for $i \leftarrow 1$ *to 2d do*

$M[i, max\{i + 1, 2d + 1\}] = p$

$M[i, min\{1, i-2\}] = 1-p$

Compute $MT = M^t$

Res = vector of size d initialized with 0s

for $1 \leq loc \leq d$ *do*

V = vector of size 2d + 1 initialized with 0s.

$V[2loc] \leftarrow 1$

$Res[loc] = V \times MT[2d + 1]$

Return Res

Procedure FindP(d,t)

$F \leftarrow$ Algorithm FindFunc(d,t).

Set $p_{opt} \leftarrow 0$.

for $F_{pivot} \leftarrow F_{1,\dots,d}$ do

Compute local maxima (p_{max}, $F_{pivot}(p_{max})$) of F_{pivot} in the range (0,1).

for each F_i, $1 \le i \le d$ do

Compute intersection point p_i of F_i and F_{pivot} in the range (0,1).

if $F_{pivot}(p_i) > F_{pivot}(p_{max})$ and $F_{pivot}(p_i) \le F_k(p_i)$ $\forall k$ then

$p_{opt} \leftarrow p_i$.

if $F_{pivot}(p_{max}) > F_{pivot}(p_i)$ and $F_{pivot}(p_i) \le F_k(p_i)$ $\forall k$ then

$p_{opt} \leftarrow p_{max}$.

Return (p_{max}, $F_{pivot}(p_{max})$).

Procedure MaximinFence(d,t)

$M \leftarrow$ FindFencePPD(d,t)

for $i \leftarrow 1$ to d do

OpP[i] $\leftarrow$ FindP(d,t) with additional given input M[i] as a vector of ppd functions.

Return OpP

Load entrance patrol chauki = P_s

Load ending patrol chauki = P_e

For each grid

For each considered factor

Cost of grid $C_{ij} = \mu_i$ score from Hawk and Dove game*

End for

End for

Place the grid consisting P_s as origin grid (0,0)

For (k = 0, k = 0 to d/N, k++)

If each unit is described according to the resolution, then

Check grids (-N, -N), (-N, 0), (-N, N), (0, -N), (N, N), (N, 0), (N, N), (0, N) for most critical grid.

If most critical grid found = C_i

Place connectivity between (0, 0) and C_i

End for.

CONCLUSION

This work presents the problem of multi-patrol guards patrolling in strong, full-knowledge, adversarial environments. In this problem a team of patrol guards is required to repeatedly visit some path, in our basic case a set of grids, and detect penetrations that are controlled by an adversary. We assume the patrol guards act in a strong adversarial model, in which the adversary has full knowledge of thepatrol guards and uses this knowledge in order to penetrate through the weakest and most critical spot of the patrol. We describe a framework for the basic case of multipatrol guards patrol around a polygon, and use this framework for developing, in polynomial time, an optimal patrol algorithm, i.e., an algorithm that strengthens the weakest and most critical spot of the patrol. This framework is then extended in order to solve the problem also in an environment and in various movement and sensing models of the patrol guards. The work makes several assumptions allowing the computation of an optimal strategy for the patrolling patrol guards. One such assumption is the first order Markovian strategy of the patrolling patrol guards. Although proving or disproving the optimality of using first order Markovian strategy is hard, it could be interesting to examine the case of higher order Markovian strategies and compare their time complexity and performance to the solution discussed here.

The present work has been developed with objectives to

a) obtain a cost-wise optimal and feasible patrol network, using a replicable computational procedure and
b) identify the most critical spots, along with their underlying vulnerability structure so as to focus efforts towards securing them.

In this work, we have used Kruskal's algorithm and travelling salesman problem to obtain a minimum spanning tree and a Hamiltonian circuit respectively that could serve as a model framework for a real-world intelligent patrolling system.

REFERENCES

1. Adleman, Leonard (November 1994), "Molecular computation of solutions to combinatorial problems", Science266 (5187): 1021–1024, doi:10.1126/science.7973651, JSTOR 2885489, PMID 7973651.
2. Agmon, N., Kaminka, G. A., & Kraus, S. (2008). Multi-robot fence patrol in adversarial domains. In Proceedings of the Tenth Conference on Intelligent Autonomous Systems (IAS-10), pp. 193–201. IOS Press.
3. Agmon, N., Kraus, S., & Kaminka, G. A. (2008). Multi-robot perimeter patrol in adversarial settings. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
4. Agmon, N., Kraus, S., & Kaminka, G. A. (2009). Uncertainties in adversarial patrol. In Proc. of the IJCAI 2009 workshop on Quantitative Risk Analysis for Security Applications (QRASA).
5. Ahmadi, M., & Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
6. Akiyama, Takanori; Nishizeki, Takao; Saito, Nobuji (1980–1981), "NP-completeness of the Hamiltonian cycle problem for bipartite graphs" (PDF), Journal of Information Processing 3 (2): 73–76, MR 596313.
7. Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V., & Chevaleyr, Y. (2004). Recent advances on multi-agent patrolling. Lecture Notes in Computer Science, 3171, 474–483.
8. Amigoni, F., Gatti, N., & Ippedico, A. (2008). Multiagent technology solutions for planning in ambient intelligence. In Proceedings of Agent Intelligent Technologies (IAT-08).
9. Anna Christiernsson (2006), Legal Limits and Adaptive Management of Wildlife Populations in Swedish Law, 11th Biennial Global Conference of The International Association for the Study of Common Property (IASCP) Bali, Indonesia.
10. Axelrod, R., 1984. The Evolution of Cooperation, Basic Books Inc., New York.
11. Axelrod, R., Hamilton, W. D., 1981. The evolution of cooperation. Science**211**: 1390 – 1396.
12. Basilico, N., Gatti, N., & Amigoni, F. (2009a). Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), pp. 565–572.
13. Basilico, N., Gatti, N., & Amigoni, F. (2009b). Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In AAMAS, pp. 57–64. Chevaleyre, Y. (2004). Theoretical analysis of the multi-agent patrolling problem. In Proceedings of Agent Intelligent Technologies (IAT-04).

14. Baum, K. A., Haynes, K. J., Dillemuth, F. P., Cronin, J. T., 2004. The matrix enhances the effectiveness of corridors and stepping stones. Ecology **85(10)**: 2671 – 2676.
15. Bavelas A (1948) A mathematical model for group structures. Human Organization **7**: 16 – 30.
16. Bavelas A (1950) Communication patterns in task-oriented groups. The Journal of the Acoustical Society of America **22**: 271 – 288.
17. Beier, P., Noss, R. F., 1998. Do habitat corridors provide connectivity? Conservation Biology**12(6)**: 1241 – 1252.
18. Beier, P., Loe, S., 1992. A checklist for evaluating impacts to wildlife movement corridors. Wildlife Society Bulletin**20**: 434 – 440.
19. Bellman, R. (1962), "Dynamic programming treatment of the travelling salesman problem", Journal of the ACM9: 61–63, doi:10.1145/321105.321111.
20. Björklund, Andreas (2010), "Determinant sums for undirected Hamiltonicity", Proc. 51st IEEE Symposium on Foundations of Computer Science (FOCS '10), pp. 173–182, arXiv:1008.0541, doi:10.1109/FOCS.2010.24, ISBN 978-1-4244-8525-3.
21. Bonacich, 1972. Factoring and weighting approaches to status scoresand clique identification. Journal of Mathematical Sociology **2(1)**: 113 - 120
22. Bondy, J., Murty, U. S. R., 2008. Graph Theory, Springer, New Delhi.
23. Borgatti S.P, Everett M.G (2006) A Graph-theoretic perspective on centrality. Social Networks **28**:466 – 484.
24. Boruvka, O., 1926. On a minimal problem. Prace Moravske Pridovedecke Spolecnosti**3**: 37 – 58.
25. Brandes U, Erlebach T (2005) Network Analysis: Methodological Foundations, Springer
26. Briers, R. A., 2002. Incorporating connectivity into reserve selection procedures. Biol. Conservation**103**: 77 – 83.
27. Bull, A. G., Urban, D. L., Keitt, T. H., 1992. Landscape connectivity: a conservation application of graph theory. Jour. Environ. Manag.**59**: 265 – 278.
28. Cantwell, M. D., Forman, R. T. T., 1993. Landscape graphs; ecological modelling with graph-theory to detect configurations common to diverse landscapes. Landscape Ecol.**8**: 239 – 255.
29. Caughley, M. S.,1977. Corridors for conservation: integrating pattern and process. Annu. Rev. Ecol. Evol. Syst. **37**: 317 – 42.
30. Conard, J., Gomes, C., Hoeve, W., Sabharwal, A., Suter, J., 2010. Incorporating economic and ecological into the optimal design of wildlife corridors. Computing and Information Science Technical Reports, Cornell University, Ithaca, NY, USA. http://hdl.handle.net/1813/17053.
31. Coppersmith, D., Doyle, P., Raghavan, P., & Snir, M. (1993). Random walks on weighted graphs and applications to on-line algorithms. Journal of the ACM, 40(3). Duarte, M. F., & Hu, Y. H. (2004). Distance-based decision fusion in a distributed wireless sensor network. Telecommunication Systems, 26(2-4), 339–350.

32. Davis., 1982. Network structure and biodiversity loss in food webs: robustness increases with connectance.Ecological Lett.**5**:558 – 567.
33. Dodd, Murphy, (2002) Community structurein social and biological networks. Proc. Natl. Acad. Sci. USA **99**: 7821 – 7826.
34. Dutta, T., Sharma, S., Maldonado, J. E., Wood, T. C., Panwar, H. S., Seidensticker, J., 2013. Gene flow and evolutionary history of leopards (*panthera pardus*) in central Indian highlands. Evolutionary Applications. doi: 10.1111/eva12078.
35. Elmaliach, Y., Agmon, N., & Kaminka, G. A. (2007). Multi-robot area patrol under frequency constraints. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
36. Elmaliach, Y., Agmon, N., & Kaminka, G. A. (2009). Multi-robot area patrol under frequency constraints. Annals of Math and Artificial Intelligence, 57(3-4), 293–320.
37. Elmaliach, Y., Shiloni, A., & Kaminka, G. A. (2008). A realistic model of frequencybased multi-robot fence patrolling. In Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08).
38. Estrada E. (2010) Generalized walks-based centrality measures for complex biological networks. Journal of Theoretical Biology. **263**: 556 – 565.
39. Estrada E., Bodin O. (2008) Using network centrality measures to manage landscape connectivity. 18(7): 1810 – 1825.
40. Estrada E., Rodriguez-Velaquez J.A. (2005) Subgraph centrality in complex networks. Physical Review. **71**: 056103.
41. Fall, A., Fortin, M-J., Manseau, M., O'Brien, D., 2007. Spatial graphs: principles and applications for habitat connectivity. Ecosystems**10**: 448 – 461.
42. Garey, M. R.; Johnson, D. S.; Stockmeyer, L. (1974), "Some simplified NP-complete problems", Proc. 6th ACM Symposium on Theory of Computing (STOC '74), pp. 47–63, doi:10.1145/800119.803884.
43. Gary W. Witmer (2005), Wildlife population monitoring:some practical considerations, Wildlife Research, 2005,32,259-263
44. Gastner, M. T., Newman, M. E. J., 2006. The spatial structure of networksEuropean Physical Jour.**B49**: 247 – 252.
45. González A. M. M, Dalsgaard B, Olesen J.M (2010) Centrality measures and the importance of generalist species in pollination networks. Ecological complexity. **7**: 36 – 43.
46. Gopal, R., Qureshi, Q., Bhardwaj, M., Singh, R. K. J., Jhala, Y. V., 2010. Evaluating the status of the endangered tiger *Panthera tigris* and its prey in Panna Tiger Reserve, Madhya Pradesh, India. Oryx**44**: 383 – 398.
47. Gustavson, F. G. (1978). Two fast algorithms for sparse matrices: Multiplication and permuted transposition. ACM Trans. Math. Softw., 4, 250–269.
48. Hanski, I., 1998. Metapopulation dynamics. Nature**396**: 41 – 49.

49. Hanski, I., Gilpin, M., 1991. Metapopulation dynamics – brief-history and conceptual domain. Biol. Journal of the Linnean Society**42**: 3 – 16.
50. Hanski, I., Ovaskainen, O., 2000. The metapopulation capacity of a fragmented landscape. Nature **404**: 755 – 758.
51. Harris, L. D., Gallagher, P. B., 1989. New initiatives for wildlife conservation: the need for movement corridors. In *Preserving Communities and Corridors* G. Mackintosh (ed.) Defenders of Wildlife, Washington D.C. pp.96.
52. Haynes, T., & Sen, S. (1995). Evolving behavioral strategies predators and prey. In IJCAI95 Workshop on Adaptation and Learning in Multiagent Systems, pp. 32–37. Lynch, N. A. (1996). Distributed Algorithms.
53. Held, M.; Karp, R. M. (1962), "A dynamic programming approach to sequencing problems", J. Siam 10 (1): 196–210, doi:10.1137/0110015.
54. Henein, K., Merriam, G., 1990. The elements of connectivity where corridor quality is variable. Landscape Ecology **4**: 157 – 170.
55. Higham N., (2008) Functions of matrices: Theory and Computation, SIAM, U.S.
56. Hofbauer, J., Sigmund, K., 1998. Evolutionary Games and Population Dynamics, Cambridge University Press, Cambridge.
57. http://math.stackexchange.com/questions/7130/reduction-from-hamiltonian-cycle-to-hamiltonian-path/1290804#1290804
58. Iwama, Kazuo; Nakashima, Takuya (2007), "An improved exact algorithm for cubic graph TSP", Proc. 13th Annual International Conference on Computing and Combinatorics (COCOON 2007), Lecture Notes in Computer Science 4598, pp. 108–117, doi:10.1007/978-3-540-73545-8_13, ISBN 978-3-540-73544-1.
59. Jhala, Y .V., Qureshi, Q., Gopal, R., Sinha, P. R., (eds.) 2011. Status of Tigers, Co-predators and Prey in India, 2010. National Tiger Conservation Authority, Govt. of India, New Delhi, and Wildlife Institute of India, Dehradun. TR2011/003 pp-302.
60. Jhala, Y. V., Gopal, R., Qureshi, Q., (eds.) 2008. Status of Tigers, Co-predators and Prey in India. National Tiger Conservation Authority, Govt. of India, New Delhi, and Wildlife Institute of India, Dehradun. TR08/001 pp-151.
61. Jiang L., Zhang W 2015. Determination of keystone species in CSM food web: A topological analysis of network structure. Network biology **5**(1): 13 – 33.
62. Johnsingh, A. J. T., Goyal, S. P., 2005. Tiger conservation in India: the past, present and the future. Indian Forester**131**: 1279 – 1296.
63. Johnsingh, A. J. T., Negi, A. S., 1998. Only disturbance-free well-managed habitats can save the tigers. Cat News28(4).
64. Johnsingh, A. J. T., Negi, A. S., 2003. Status of tiger and leopard in Rajaji-Corbett conservation unit, Northern India. Biol. Conservation**111**: 385 – 393.
65. Johnsingh, A. J. T., Prasad, S. N., Goyal, S. P., 1990. Conservation status of the Chila-Motichur Corridor for elephant movement in Rajaji-Corbett National Parks area. India.Biol. Conservation**51**: 125 – 138.

66. Johnsingh, A. J. T., Ramesh, K., Qureshi, Q., 2004. Conservation status of tiger and associated species in the terai Arc Landscape, India. Technical Report, Wildlife institute of India, Dehradun.
67. Jordan F., Liu W.C., Davis A. J.,2006. Topological keystone species: measures ofpositional importance in food webs. Oikos **112**: 535 - 546.
68. Keitt, T. H., 2003. Network theory: an evolving approach to landscape conservation. In Ecological Modelling for ResourceManagement,Dale, V. H. (ed), Springer, New York.
69. Kruskal, J. B. Jr., 1956. On the shortest spanning subtree of a graph and the travelling salesman problem. Proc. Amer. Math. Soc.**7 (1)**: 48 – 50.
70. Kruskal, J. B., 1997. A reminiscence about shortest spanning subtrees. Archivum Mathematicum339**(1-2)**: 13 – 14.
71. Lindenmeyer, D., Hobbs, R. J., Montague-Drake ,R., Alexandra, J., Bennett, A., Cale, P., Calhoun, A., Cramer, V., Cullen, P., Driscoll, D., Fahrig, L., Fischer, J., Franklin, J., Haila, Y., Hunter, M., Gibbons, P., Lake, S., Luck, G., MacGregor, C., McIntyre, S., MacNally, R., Manning, A., Miller, J., Mooney, H., Noss, R., Possingham, H., Saunders, D., Schmigelow, F., Scott, M., Simberloff, D., Sisk, T., Tabor, G., Walker, B., Wiens, J., Woinarski, J., Zavaleta, E., 2008. A checklist for ecological management of landscapes for conservation. Ecology Letters**11**: 78 – 91.
72. Martello, Silvano (1983), "An Enumerative Algorithm for Finding Hamiltonian Circuits in a Directed Graph", ACM Transactions on Mathematical Software (TOMS)9 (1): 131–138, doi:10.1145/356022.356030
73. Michael R. Garey and David S. Johnson (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, ISBN 0-7167-1045-5 A1.3: GT37–39, pp. 199–200.
74. Minor, E. S., Urban, D. L., 2008. A graph-theory framework for evaluating landscape connectivity and conservation planning.Conservation Biol. **22(2)**: 297 – 307.
75. Minor, E., Urban, D., 2007. Graph theory as a proxy for spatially explicit population models in conservation planning. Ecological Applications**17**: 1771 – 1782.
76. Morgan Kaufmann. Paruchuri, P., Pearce, J. P., Tambe, M., Ordonez, F., & Kraus, S. (2007). An efficient heuristic approach for security against multiple adversaries. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08).
77. MsTripes
78. Opdam, P., Steingrover, E., van Rooij, S., 2006. Ecological networks: a spatial concept for multi-actor planning of sustainable landscapes. Landscapes and Urban Planning**75**: 322 – 332.
79. Opdam, P., Verboom, J., Powels, R., 2003. Landscape cohesion: an index for the conservation potential of landscapes for biodiversity.Landscape Ecology**18**: 113 – 126.

80. Ozgur A., Vu T., Erkan G., Radev D.R., (2008) Identifying gene-disease associations using centrality on a literature mined gene-interaction network. Bioinformatics **24(13)**: i277 – i285
81. Paladugu S. R., Zhao S., Ray A., Raval A., (2008) Mining protein networks for synthetic genetic interactions. BMC Bioinformatics **9(1)**: 426
82. Papadimitriou, Christos H. (1994), "On the complexity of the parity argument and other inefficient proofs of existence", Journal of Computer and System Sciences 48 (3): 498–532, doi:10.1016/S0022-0000(05)80063-7, MR 1279412.
83. Plesńik, J. (1979), "The NP-completeness of the Hamiltonian cycle problem in planar digraphs with degree bound two" (PDF), Information Processing Letters 8 (4): 199–201, doi:10.1016/0020-0190(79)90023-1.
84. Pulliam, H. R., 1988. Sources, sinks and population regulation. American Naturalist**132**: 652 – 661.
85. Quy., Tambe, M., Ordonez, F., & Kraus, S. (1993). Security in multiagent systems by policy randomization. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-07).
86. Rayfield B, Fortin M-J, Fall A (2011) Connectivity for conservation: a framework to classify network measures. Ecology **92**(4): 847 – 858.
87. Rubin, Frank (1974), "A Search Procedure for Hamilton Paths and Circuits", Journal of the ACM21 (4): 576–80, doi:10.1145/321850.321854.
88. Sharma, S., Dutta, T., Maldonado, J. E., Wood, T. C., Panwar, H. S., Seidensticker, J., 2013. Forest corridors maintain historical gene flow in a tiger metapopulation in the highlands of Central India, Proc. R. Soc. B **280**: 20131506.
89. Taylor, P. D., Fahrig, L., Henein, K., Merriam, G., 1993. Connectivity is a vital element of landscape structure. Oikos**68**: 571 – 573.
90. Taylor, P. D., Fahrig, L., With, K. A., 2006. Landscape connectivity: a return to the basics. In: Connectivity conservation, Crooks, K.R. & Sanjayan, M. (eds.), pp.29 – 43, Cambridge University Press, Cambridge.
91. Thomason, A. G. (1978), "Hamiltonian cycles and uniquely edge colourable graphs", Advances in Graph Theory (Cambridge Combinatorial Conf., Trinity College, Cambridge, 1977), Annals of Discrete Mathematics 3, pp. 259–268, doi:10.1016/S0167-5060(08)70511-9, ISBN 9780720408430, MR 499124.
92. Urban D L, Minor E S, Treml E A, Schick R S (2009) Graph models of habitat mosaics. Ecology Lett. **12**: 260 – 273.
93. Urban, D. L., Keitt, T. H. 2001. Landscape connectivity: a graph-theoretic perspective. *Ecology* **82**: 1205 – 1218.
94. Webb, J. N., 2007. Game Theory: Decisions, Interactions and Evolution, Springer, New Delhi.

95. Wikramanayake E D, Dinerstein E Robinson , J G, Karanth U, Rabinowitz A, Olson D, Mathew T, Hedao P, Conner M, Hemley G, Bolze D (1998) An ecology based method for defining priorities for large mammal conservation: the tiger as case study. Conservation Biol. **12**(4): 865 – 878.
96. Wikramanayake E, McKnight M, Dinerstein E, Joshi A, Gurung B, Smith D (2004) Designing a conservation landscape for tiger in human-dominated environments. Conservation Biol. **18**(3): 839 – 844.
97. Yumnam B, Jhala Y D, Qureshi Q, Maldonado J E, Gopal R, Saini S, Srinivas Y, Fleischer R C (2014) Prioritizing tiger conservation through landscape genetics and habitat linkages. PLOS ONE **9**(11):e111207.

APPENDIX A: Plagiarism Report (Turnitin)

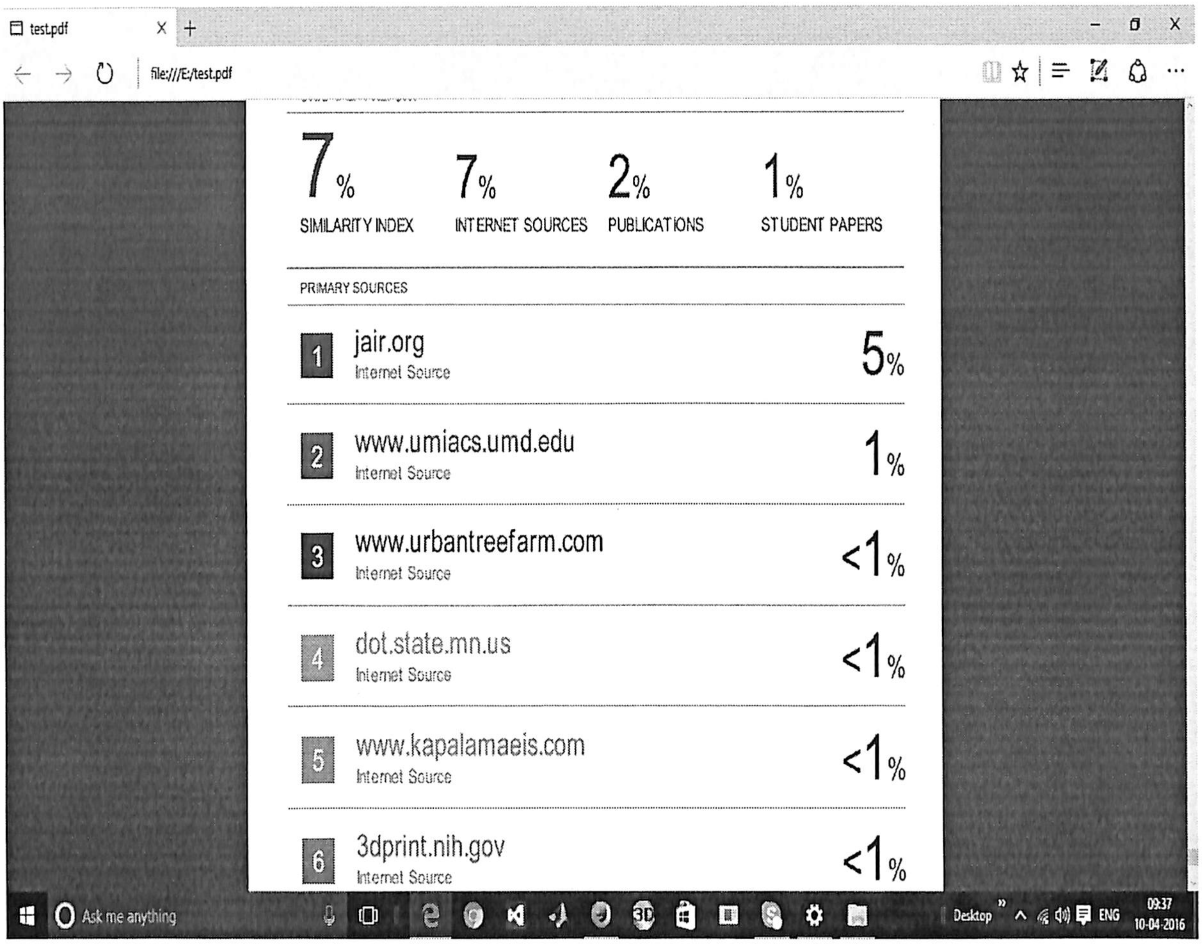


Figure A. plagiarism report for the thesis A Computational model for wildlife reserve management.

11/04/16.

APPENDIX B: Paper for Publication

A COMPUTATIONAL MODEL FOR MANAGEMENT OF WILDLIFE RESERVES: AN INTELLIGENT PATROLLING SYSTEM APPROACH

Saurabh Shanu[1], Qamar Qureshi[2], Yadvendradev Jhala[3], Anil Kumar[4*]

[1]Centre for Information Technology, University of Petroleum and Energy Studies, Dehradun 248007, Uttarakhand, India

[2]Department of Landscape Level Planning and Management,

[3]Department of Animal Ecology and Conservation Biology,

Wildlife Institute of India, Post Box # 18, Chandrabani, Dehradun 248 001, Uttarakhand, India

[4]Centre for Information Technology, University of Petroleum and Energy Studies, Dehradun 248007, Uttarakhand, India

[*]Author for correspondence. E-mail: ANIL.KUMAR@ddn.upes.ac.in

Abstract

The problem of adversarial multi-path guard dependent patrol has gained interest in recent years, mainly due to its immediate relevance to various security applications. In this problem, patrol guards are required to repeatedly visit a target area in a way that maximizes their chances of detecting an adversary trying to penetrate through the patrol path. When facing a strong adversary that knows the patrol strategy of the guards, if the guards use a deterministic patrol algorithm, then in many cases it is easy for the adversary to penetrate undetected (in fact, in some of those cases the adversary can guarantee penetration). Therefore, this project presents a non-deterministic patrol framework for the guards. Assuming that the strong adversary will take advantage of its knowledge and try to penetrate through the patrol's weakest spot, hence an optimal algorithm is one that maximizes the chances of detection in that point. We therefore present a polynomial-time algorithm for determining an optimal patrol under the Markovian strategy assumption for the guards, such that the probability of detecting the adversary in the patrol's weakest spot is maximized. We build upon this framework and describe an optimal patrol strategy for several patrol guards based on their movement abilities (directed or undirected) and sensing abilities (perfect or imperfect), and in different environment models - either patrol around a perimeter (closed polygon) or an open fence (open polyline).

In this work, we use game theory and graph theory to model and design a patrolling guard path web. We construct a graph using the patrol chaukis as vertices and the possible paths between these vertices as edges. A cost matrix is constructed to indicate the cost incurred by the patrol guard for passage between the habitat patches in the landscape, by modelling a Hawk and Dove game. A minimum spanning tree or a Hamiltonian path, depending on the start and end point is then obtained by employing Kruskal's algorithm or Travelling Salesman problem, which would suggest a feasible adversary detection path for the patrol guards within the landscape complex.

Keywords: Patrol Guards, adversary, Hawk and Dove game, Graph theory, Minimum spanning tree, Hamiltonian Path, Kruskal's algorithm, Travelling Salesman Problem.