Statistical Model Checking for Cops and Robbers Game on Random Graph Models

Abhishek Jain

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

Department of Computer Science & Engineering

June 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

Abhishek Jain

(Abhishek Jain)

CS16MTECH11001

(Roll No.)

Approval Sheet

This Thesis entitled Statistical Model Checking for Cops and Robbers Game on Random Graph Models by Abhishek Jain is approved for the degree of Master of Technology from IIT Hyderabad

N.R.Ar

(Dr. N.R.Aravind) Examiner
Dept. of Computer Science & Engineering
IITH

(Dr. Ramakrishna Upadrasta) Examiner
Dept. of Computer Science & Engineering
IITH

M.V.Pandurg R

(Dr. M. V. Panduranga Rao) Adviser
Dept. of Computer Science & Engineering
IITH

(Dr. Subrahmanyam Kalyanasundaram) Chairman
Dept. of Computer Science & Engineering
IITH

Acknowledgements

Firstly, I would like to thank Dr. M. V. Panduranga Rao, Associate Professor at Department of Computer Science & Engineering, Indian Institute of Technology, Hyderabad. This work would not have been possible without his constant support and sincere advice throughout the work. I would also like to thank all other faculty members of Department of CSE at IITH for their continuous review and feedback of the work and the motivation to keep pursuing the research interest.

I would also like to express my gratitude to Shiraj Arora, Research Scholar, CSE, IITH and Yenda Ramesh, Research Scholar, CSE, IITH for all the help and support while working.

I would also like to thank my parents and friends who have always motivated me to work hard and has been my constant support in all the situations.

Abhishek Jain
CS16MTECH11001
Department of Computer Science & Engineering
Indian Institute of Technology, Hyderabad

Dedication

To my parents, teachers, and friends.

Abstract

Cops and robbers problem has been studied over the decades with many variants and applications in graph searching problem. In this work, we study a variant of cops and robbers problem on graphs. In this variant, there are different types of cops and a minimum number of each type of cops are required to catch a robber. We studied this model over various random graph models and analyzed the properties using statistical model checking.

To the best of our knowledge this variant of the cops and robber problem has not been studied yet. We have used statistical techniques to estimate the probability of robber getting caught in different random graph models. We seek to compare the ease of catching robbers performing random walk on graphs, especially complex networks. In this work, we report the experiments that yields interesting empirical results. Through the experiments we have observed that it is easier to catch a robber in Barabási Albert model than in Erdös-Rényi graph model. We have also experimented with k-Regular graphs and real street networks.

In our work, the model is framed as the multi-agent based system and we have implemented a statistical model checker, SMCA tool which verifies agents based systems using statistical techniques. SMCA tool can take the model in JAVA programming language and support Probabilistic - Bounded LTL logic for property specification.

Contents

Chapter 1

Introduction

Most of the systems or phenomenon in the real world are probabilistic in nature because of the randomness and uncertainty exhibited by the applications. Because of the criticality in the behavior of such systems, the biggest question that arises is of the correctness of these systems. Therefore, in today's world, the systems and applications need some method and formalism to incorporate the non-determinism and randomness and assure the users of the correctness.

Model checking of the probabilistic systems has been studied for many years now but the wide nature of systems have made it difficult to apply model checking techniques to all the systems. In probabilistic model checking, the goal is to model the system using some well-defined formalism that meets the properties of the system and to verify the desired properties which are expressed using the appropriate logic. With the complex and large system, the state explosion problem[1] has been known to be the biggest obstacle with the numerical model checking. In numerical checking, the system is developed in such a way that the state space consists of all the states that the system can probably reach with all the possible transitions. Although numerical model checking can produce the results with higher accuracy the memory requirements become so large that it becomes almost impossible to verify complex systems with numerical model checking. For complex systems, it is easy to go wrong during the process of modeling a system–the model might not faithfully represent the system. Correct model checking of wrong models of the system is of no use, if not outright detrimental in the analysis of the system. Over the last decade, to counter the problem of state explosion various techniques involving statistics about the system have been studied. Younes in [2] has compared the use of numerical model checking and statistical model checking over the stochastic systems and concluded that statistical methods have better scaling chances than the numerical ones, though numerical methods are sure to verify the systems with higher accuracy. In the case of stochastic systems, modeling generally varies with the way movement of different entities is modeled in the application. In our work, we have also a technique to model the movement of different cops and robbers in the graph which will be discussed in detail in upcoming sections.

In this work, we have focused on the systems that consist of various agents performing a random walk in the system. A random walk is a process of exploring the graph within the constraints of the connectivity of the graph. We have studied the properties and behavior of the agents moving across an underlying graph structure. The movement of the agents is probabilistic based on the stochastic transition probabilities. Throughout this study, we have followed the randomized cops and robbers example where multiple cops and single robber are performing the random walk on the graph and we verify the properties through statistical methods. Our study and experiments are performed on different graph models. We have formalized our model as the collection of multiple transition systems where every transition system has their own probability transition matrix to make transitions from one location to another over the graph. The property to be verified on the system is an aggregation of all the transition systems of all the entities in the network and is verified by the statistical methods by collecting sample paths of the model. Verification results obtained through statistical model checking are not totally accurate and are bounded by certain confidence parameters. In our cops and robber example, we estimate the probability of robber getting caught in the network within the given time bound using probability estimation technique.

Also, our model can also be seen as the multi-agent system where the agents can be oblivious to or interacting with each other. Classically in a multi-agent system, there are various types of agent are present in the network and all the agents keep moving in the network either interacting with each other or with no interaction performing their respective tasks. In [3], verification of the team formation protocol through the collaboration of the interacting agents has been studied by forming the model into a two-player stochastic game. To implement and verify our model in the context of multi-agent systems, we have studied and verified the problem of epidemic spreading. In this problem, the population across the different locations in the network have different health status and have a tendency to get the disease through the infected people. The transfer of infection is however based on different heuristics and statistical methods are used to estimate the time and rate of epidemic ending across the network.

Majorly in this study, we aim to study the cops and robber problem where multiple cops try to catch the robber moving in the network. Our modeling of cops and robber problem varies from classical cops and robber in various aspects. In our model, we have the multi-color cops patrolling in the network. There is a threshold number of cops required to catch a robber at the same vertex as that of the robber at the same time. Each cop and robber is modeled as a separate agent which can have their own rules of movement on the graph. Though every agent in the networks belongs to a certain type of agent in the model. Through this example, we show that our modeling each type of agent as the separate entity has provided an easy way to model these kinds of stochastic systems.

The model discussed in this work can also be used to model other problems which involve the combination of transition systems and also the agents that interact with each other thought simulation. Example of such system in several areas like analyses of epidemics, rumor spreading, broadcast of the messages in the network, social media analytic. As most of these problems involve the agents walking in the network, we feel our model is appropriate to be applied to these problems and verify the properties with statistical model checking.

1.1 Randomized Multi Colored Cops and Robbers

In our model, we have given a different approach to cops and robber problem. Our theory of cops and robber problem differs significantly from the standard game-theoretic literature. We have considered our model to be fully random where all the cops and robber perform a random walk in the network. The major points in our modeling of cops and robber are described as follows :

- Movement of cops and the robber is simultaneous in our model at every time step, not alternative movements like in standard cops and robber.
- Cops and robbers do not know each other's position in the network. They can only see the other player when on the same vertex. Even the position of the players in the neighbor is not known.

- The movement of all the cops and the robbers in the network is completely random. All the players are performing a random walk in the graph and every player's movement is independent of any other player in the network.

- As the movement is completely random and independent, both the cops and the robber move without any strategy in the network. The next move of both the cops and the robber only depends on the connectivity of the graph from their current location and nothing else.

- The major difference in our model is the requirement to catch the robber. In our model, we require 3 cops to catch a single robber at the same time on the robber's vertex. In the case of multiple robbers, more than 1 robber can be caught at the same vertex with the help of same 3 cops at the same time unit.

- Also in our model, we have implemented two different settings of cops and robbers,

  3 colors cops : In this setting of cops and robber, we have 3 colors of cops moving in the network randomly. To catch the robber, 1 cop of each color is required at the robber's vertex at the same time.

  1 color cops : In this layout of cops and robber scenario, 3 cops of the same color are required to catch the robber on the robber's vertex at the same time.

Fig. 1.1 shows the initial setting of multi-colored cops and robber scenario, there are 3 different types of cops represented by the colored rectangles and the robber is using the triangle. The condition to apprehend the robber is the presence of at least one red cops, one green cop and one blue cop at the robber's vertex at any time instant.

Fig. 1.2 shows the different possibilities in the multi-color cops scenario at some time instant. In fig. 1.2a, the robber is not caught even when the two cops are present at the same location as that of the robber. While in fig. 1.2b, the robber is caught when all 3 color cops are present the robber's vertex at the same time.

We start with the description of cops and robber literature in our work in chapter 2, we describe how does our model differ from the existing game-theoretic cops and robber literature. Overview of the related work with the theoretical background over statistical model checking is discussed in chapter3. We also talk about the various techniques for estimating the probability of any property to be true in the system. We briefly discuss the various graph model used to generate the networks in our model. Chapter 4 presents our model framework with the architecture details. In this

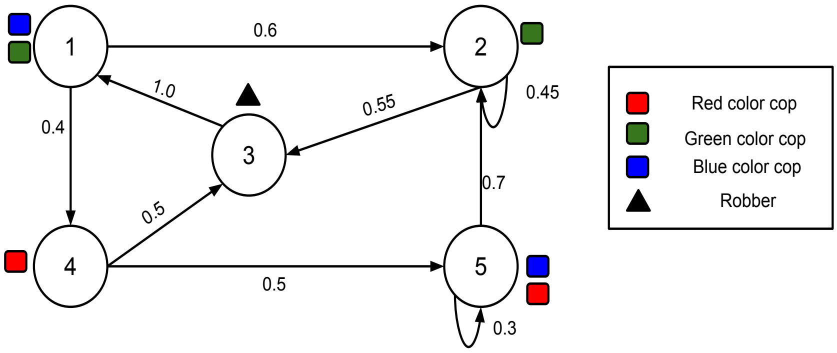
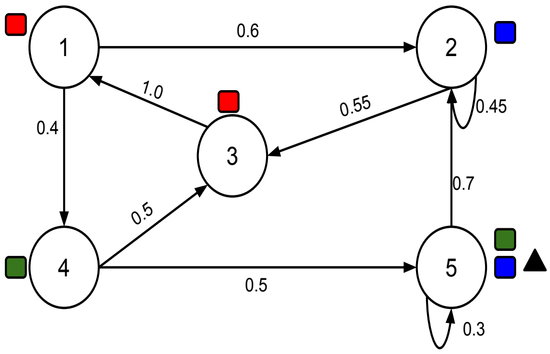
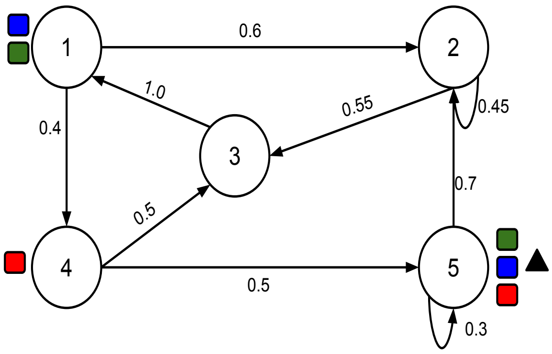


Figure 1.1: Multi-colored cops and robber scenario



(a) Robber is not caught



(b) Robber is caught

Figure 1.2: Different possibilities in multi-colored cops setting

section, we discuss the modeling of the system and specifications of the properties of the systems to be verified using the SMCA tool. Also, we describe the implementation details with the discussion over the set of experiments performed on the discussed model. In chapter 5, we will talk about the result and will conclude our work in chapter 6 with the future scope of this work.

Chapter 2

Cops and Robber Literature

The problem of cops and robber as two player pursuit-evasion game was introduced by Quilliot in [4] in his doctoral thesis and also independently by Nowakowski and Winkler in [5]. In both of their work, the idea of only one cop was considered. Later in 1984, Aigner and Fromme in [6] introduced the idea of Cop Number, which is the minimum number of cops required to catch the robber. Aigner and Fromme in their work proved that only 3 cops are necessary to catch the robber in planar graphs. Later the idea of cop number is extensively researched with variations of cops and robber scenario. Kehagias et. al [7] have studied the problem in which the cops are chasing a drunk robber i.e., the robber is performing a random walk on the graph. Moldenhauer et. al [8] have studied about the policies for the robber to escape the cops in the graph as opposed to the other works where the focus is only giving to formulate the optimal strategies for the pursuers. In their work, they have found that a greedy approach is better than all other approaches. Bonato et. al [9] have taken a different approach and studied the game for distance k cops and robber in which the cop can catch the robber from k distance. They have studied and proved the bound of cop number in the terms of the size of the graph.

Studies conducted on Cops and Robber problem are based on the alternate movement of cops and robber while in our model we have considered the simultaneous movement at every time instant. Also, in our model, both cops and robber have no pre-decided strategy to move in the graph and the movement is randomly based on their probability distribution.

In the cops and robber literature, there is a set of cops and a robber moving a network of nodes or locations. The cops objective is to catch the robber while the robber tries to escape the cops, the game ends whenever cops are successful in catching the robber. Over the years, this game theoretic approach has been studied widely in many different forms maintaining the core concept behind the literature. Studies have been conducted with both probabilistic, structural approaches for this problem. There exist versions of the problem where both cops and robber follow pre-defined strategies to win the game. The main idea has always been to estimate the time cops take to catch the robber successfully. A robber is caught when the cop is present at the same vertex as that of the robber at the same time instant. Although over the last decade, many studies have been conducted trying to estimate the *Cop Number* i.e., the number of cops required to catch the robber, over different types of graphs, several bounds on cop number have been proved for certain types of graphs.

In this work, we have modified the core concept of traditional cops and robber game theoretic literature. We have tried to induce more randomness through the way this game is played. Also, our model of cops and robber does not include any pre-defined strategy either for cops to catch the robber or for the robber to escape the cops. We have studied the characteristics and impact of randomness over cops and robber problem. We also attempt to model this problem as the multi-agent problem where there can be more than one type of cops present in the graphs. We will first discuss the background of classical cops and robber literature with different variants studied in the past and then will briefly discuss the modifications done in our model.

2.1 Traditional Cops and Robber

In this section, we briefly describe the traditional version of cops and robber problem being studied in most the research works. We will also talk about the variants of cops and robber game theoretic literature. Cops and robber is a pursuit-evasion game played on a graph $G = (V.E)$, where one or multiple cops initially place themselves on different nodes of the graph (though there can be more than one cop agent on the same node) and the robber is also placed initially on a node of the underlying network. The initial placement of both cops and robber can either be random or according to a strategy to play the game for upcoming steps. The motive of the cops is to capture the robber while the robber tries to run away from the cops. The robber is captured when a cop is present on the same location as that of the robber at the same time instant. In standard cops and robber theory, cops and robber know each others location and take turn alternatively to move from one node to another.

Combination of the moves of both the cops and the robber is considered to be as one round in the game. Both the cops and the robber can only move to its neighboring nodes and cannot jump over the nodes. At any time instant, either only the robber moves or the cops move to some other location. The cops win if they capture the robber successfully and the robber wins if he can escape the capture for the indefinite time period. Most of the studies are conducted in the context of cop number where the motive is to set a bound over the minimum number of cops required to catch a robber for the specific classes of graphs. So, the standard game is of perfect information where both the players know the graph and the positions of each other. Knowing each other's position in the graph help both the players to strategize the next move in order to win the game. Thus this game can be expressed as a sequence of events.

2.1.1 Variants of Cops and Robber Problem

With the core concept of established theory of cops and robber, several variants of this theory have been studied over the years. A number of variations for this game are possible in which constraints can be put over the different moves of the players or the underlying graph can be changed to a specific class of graphs.

1. **Distance k cops and robbers**

   Bonato et. al in [9] have proposed a variant of standard cops and robber game. In their work, they proposed a structure in which the cops win if any one of the cops is at most at the distance k from the robber in the graph G. They showed that there exists a polynomial time algorithm that can determine the *Cop Number*, c_k for the graph G under the constraint that the number of cops are fixed and are not dependent on the size of the graphs.

2. **Pursuing a fast robber**

   In [10], authors have modeled a variant of cops and robber problem where the robber has an advantage of moving faster than the cops in the graph. In a single move, the robber can traverse r number of edges whereas a cop can only move by a single edge in one move. In their work, they have proved that computing c_k on a given graph is NP-hard and also shown that for planar graphs the cop number c_k is unbounded if the robber is faster than the cops in the graph.

3. **Cops and invisible robber**

Kehagias et. al in [11] has examined a different version of cops and robber game in which the robber is invisible to the cops until he is captured. The robber in their game knows of the cops previous moves but not the upcoming ones. As the cops are unaware of robber's location, a pre-defined strategy might not work out all the times and thus the movement of cops can be randomized. They also study one more variant where the robber is supposed to be drunk and is performing a random walk. The authors have studied the invisible cost of drunkenness which they defined as the ratio of expected capture time in a standard case to that of the drunk robber case.

In [12], a different approach was followed for cops and robber game in which perfect information is not provided to the cop. They have only focused on the single cop version of the game. They studied the effect of reducing cop's visibility in the graph i.e., the robber can be seen by the cop only when the distance between them is at most to some threshold. It has been shown that even with small or no visibility a cop can catch the robber in a cop-win graph, no matter how far the robber can see. But the reduction in the cop visibility causes the capture time to increase exponentially. It is also shown that if the visibility ranges for both the players are same, then a greedy strategy is sufficient to capture the robber by a single cop.

4. **Hunter - Rabbit game**

   In this version of cops and robber game discussed in [13], a hunter tries to catch a rabbit on an undirected graph. Both the players are assumed to know the graph in advance but cannot see each other until they are at the same vertex i.e., until rabbit is caught by the hunter. The movement of both the players over the graph is randomized strategies. It has been shown that there can be a strategy for the hunter such that the expected number of rounds to catch the rabbit is $O(nlogn)$, where n is the number of vertices in the graph. Also, a slightly different variation is also studied in which the rabbit has the ability to jump over the nodes i.e., not necessarily following the edge connectivity in the graph.

5. **Lazy Cop and robber**

   In [14] an interesting variant of cops and robber has been studied in which the cops on the graph G are lazy. At every turn of the cops, only one of the cop change its position and the rest remains on the same vertex as they were before.

Authors in this work have studied the lazy cop number and for the binomial random graphs, a bound on lazy cop number is provided.

6. **Firefighter**

   In this problem, fire breaks out at some vertex of the graph and is continuously spreading to all the neighboring vertices in the graph. The fire can be contained by the firefighters by putting out the fire at each vertex turn by turn. Studies conducted in this context estimates the expected number of firefighters required to contain the fire in the graph and also to try to minimize the number of vertices burnt. The major difference with the standard cops and robber problem is that the vertices which are saved by the firefighters cannot be burnt again by the fire whereas, in cops and robber, both the players can visit the same vertex again and again. A detailed survey about this problem is discussed in [15].

7. **Angel and the devil**

   This game proposed by Conway in [16] is played on the infinite chess board by the two players, Angel and the Devil. The angel has the power to move a maximum of k squares on the board in a single move while the devil places a block on any one of the squares other than that of the angel. The devil wins the game if the angel is captured in such a way that it is not possible for him to move at all in his turn and the angel wins the game if it can move for an indefinite period.

8. **Wall cops and robber**

   Another variation of the cops and robber game motivated by the angel and the devil theory was studied in [17]. In this game the cops have a special power to build a wall over a vertex they move to and the robber cannot move to a vertex with a wall. The robber can only move the vertices which do not have a wall. The robber is captured once he is surrounded by the walls. In their study, the aim is to estimate the *Wall Capture Time*, the time taken by a single cop to catch the robber by building walls on the vertices.

For further reading interested readers may refer to [18, 19] in which cops and robber problem have been explored for different types of graphs and for different domains and also the open problems related to this theory have been discussed.

Chapter 3

Background

In this chapter, we will briefly discuss the terms used in the background of this work for the better understanding of the rest of the chapters. We will start by briefly discussing transition system and temporal logic and then will describe the technique of model checking. We will specifically look into the concepts of statistical model checking and probability estimation. Then we will end the chapter with the discussion of graph models which are used for the experiments over our model.

3.1 Transition System

To demonstrate the behavior of the discrete system, a concept of transition system is used in the field of computation. A transition system basically models the dynamic behavior of any system in the form of states and transition between those states. It can be seen as the representation of the processes which are defined by some pre-decided rules.

A transition system, T is a tuple $(S, s_{init}, Act, P, AP, L)$ where,

- S is the set of states in T
- s_{init} is the set of initial states and $s_{init} \subseteq S$
- Act is a set of actions
- $P \subseteq S\ X\ Act\ X\ S$, is the transition relation
- AP is the set of atomic propositions
- $L\ :\ S\ \rightarrow\ 2^{AP}$ is the labelling function

A transition system can be both finite or infinite. Fig. 3.1 shows a transition system which generates a binary string. In this system the set $\{s_0, s_1, s_2, s_3\}$ represents the set of states out of which s_0 is the initial state. {0, 1, r, s} is the set of actions, {restart, stop} is the set of atomic propositions.

Transition systems can also be stochastic i.e., the transition relation, P will be a stochastic relation and will be defined as follows:

$$P : SXS \rightarrow [0, 1]$$

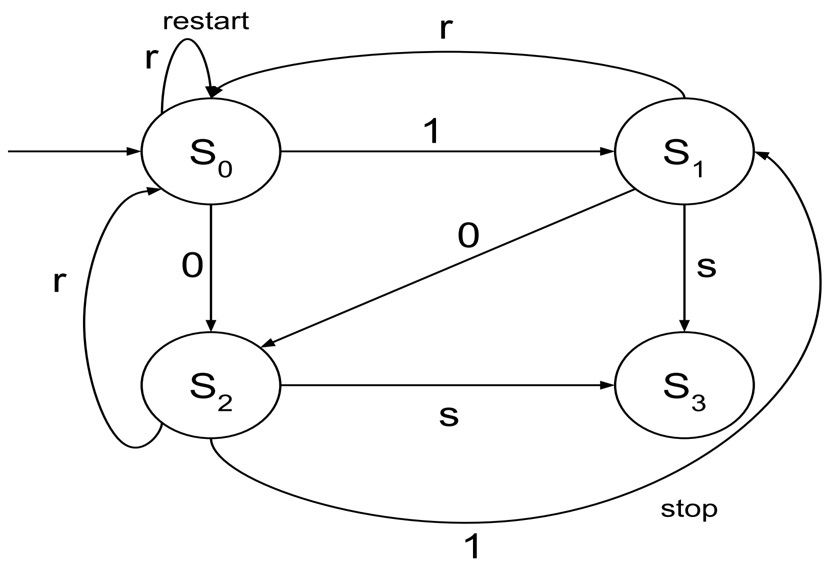


Figure 3.1: Transition system to generate a binary string

3.2 Temporal Logic

For the better understanding of the model checking and the model developed in our work, it is important to understand the theory behind the temporal logic which is used to specify the properties to be verified over the model using model checking techniques.

Temporal logic is a branch of logic which is concerned with the propositions that have their values dependent on time. It can be used to represent all the approaches based on time. This idea of temporal logic was first introduced in 1960 by Arthur Prior and then later on expanded by other scientists. We will now look in to into Probabilistic version of Bounded Linear Temporal Logic (BLTL) which has been used in our work to specify the properties to be verified on the model.

3.2.1 Linear Temporal Logic (LTL)

LTL was first introduced by Amir Pnueli in [20] for the verification of the programs based on temporal reasoning. LTL is basically a modal temporal logic that can be used to specify the paths generated by any system with the time steps. It defines the infinite sequence of events over any system based on the linear time properties.
ϕ is an LTL formula which is defined over a set of atomic propositions AP for any transition system M.
Syntax :

$$\phi \ ::= \ T \mid a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid X\phi \mid \phi_1 \cup \phi_2$$

where :

- a is an atomic proposition
- ϕ, ϕ_1, ϕ_2 are the valid LTL formulas
- X is the next operator
- $\cup$ is the Until operator

Operators :
For the given LTL formulas ϕ_1 and ϕ_2,

- $X\phi_1$: It is the next operator which evaluates to true if ϕ_1 holds in the next state.
- $G\phi_1$: G is the global operator which states that the formula ϕ_1 is true in all the future states.
- $F\phi_1$: It states eventually ϕ_1 i.e., the formula should hold true for some future state.
- $\phi_1 \cup \phi_2$: Until operator which states that the formula ϕ_1 is true for all the states until the formula ϕ_2 becomes true for some state.

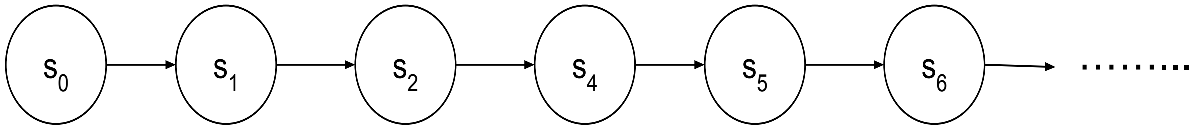


Figure 3.2: Example of a trace for a transition system

Semantics :
LTL formula ϕ is evaluated for the truth value over a trace of the transition system M. A trace of any transition system is the sequence of states visited in the system in discrete time steps. Traces can be either finite or infinite. Let us consider an infinite trace t for M. As shown in Fig. 3.2 is the infinite trace of the transition system.
We always evaluate the LTL formula always on the first state of the trace i.e., s_0.
So in a trace, $\sigma : [s_0, s_1, s_2, s_3, s_4, ...]$
$\sigma \models true$
$\sigma \models a \quad \Leftrightarrow \quad a \epsilon s_0$
$\sigma \models \phi_1 \wedge \phi_2 \;\; \Leftrightarrow \;\; \sigma \models \phi_1 \; and \; \sigma \models \phi_2$
$\sigma \models \neg\phi \;\; \Leftrightarrow \;\; \sigma \neq \phi$
$\sigma \models X\phi \;\; \Leftrightarrow \;\; \phi \models s_1$
$\sigma \models \phi_1 \cup \phi_2 \;\; \Leftrightarrow \;\; \exists j \geq 0, \sigma[j...] \models \phi_2 \; and \; \sigma[i...] \models \phi_1, 0 \leq i < j$

3.2.2 Probabilistic BLTL

The idea of Bounded LTL (BLTL) was introduced by Kamide in [21]. This logic is the enhancement of the Linear Temporal Logic (LTL) with the constraints of bounds in the form of steps or time units. In BLTL, all the temporal operators of LTL i.e., (F, G, U) are bounded using the temporal bound. The semantics of BLTL are similar to the LTL within the restrictions of temporal bound.

Probabilistic BLTL (P-BLTL) is the enhancement of BLTL with the nested probability operator, that checks whether the probability of the property expressed in BLTL is more or equal to the given threshold value. So, the basic property to be checked through P-BLTL will look like this,

$$Pr \;\geq\; \theta \; [property]$$

where, *property* is some Bounded LTL formula to be checked over the traces of transition system.

3.3 Model Checking

For the complex systems, it is challenging and important to verify the working of the system i.e., to check whether the system is behaving as it is supposed to. Verification of the critical systems, like ATMs, space jets is even more important and difficult at the same time because the margin of error is negligible in such systems. In this section, we will briefly discuss the model checking technique that can be used for the verification of the systems.

Model checking is verification technique based on the model of the system under verification. Precise modeling of the system in a mathematical manner is utmost important for the model checking task. The properties of the system under verification are to expressed in using some temporal logic discussed in the previous section.

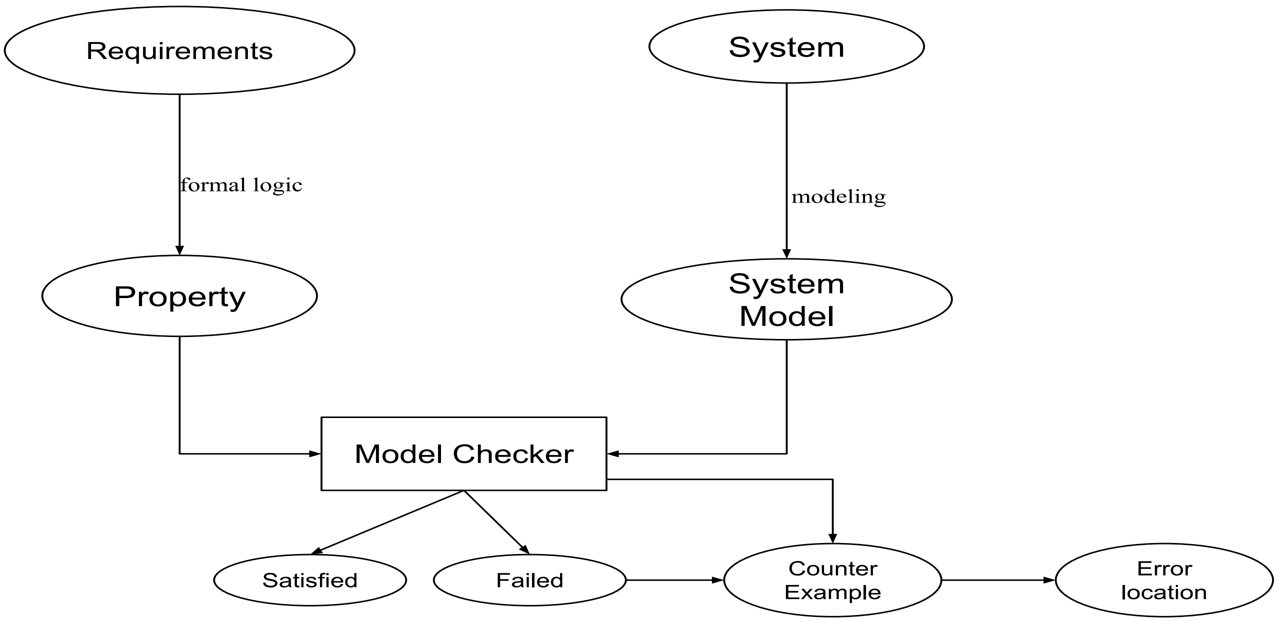


Figure 3.3: Model Checking Technique

Fig. 3.3 describes the basic technique of model checking. Firstly, the system to be verified is converted into a model using some modeling specifications like Discrete Time Markov Chains and then the requirements of the system have to be expressed using a formal logic which coverts the requirements into property specifications. Now, the system model and the property specifications are given as the inputs to the software tool known as *Model Checker*. The task of the model checker is to verify the property in question over all the possible system scenarios in an ordered manner. If the property is satisfied in all the system scenarios then model checker will return *satisfied* for the property, otherwise *failed* will be given as the output by the model checker with the system scenario where the property fails. This failed scenario can then be used to determine the cause of the error and the same process can be repeated to verify the property again.

It is important to note here that the modeling of the system should be accurate and the model should demonstrate the design of the system precisely. Also, the formal specification of the property should also express the requirements of the system strictly. Any inaccuracy in either the model or the requirement of the system can result in unexpected results. Model checker just reads the model and verifies the property over that model in a brute force manner. Also whenever the model checking over a complex and large system is done, the state space can become so large that it may not be possible to store the state spaces in available memory of the system.

3.3.1 Probabilistic Model Checking

There are a lot of probabilistic systems in the real world scenario which exhibits non-deterministic and probabilistic behavior. To model such systems, probabilistic models such as discrete time Markov chains (DTMCs) are used. Probabilistic Model Checking (PMC) is an adaption of the actual model checking technique which is used to verify the models which have the stochastic behavior. This is one of the famous techniques to model and to verify the real world applications. As the inputs for standard model checking technique, it also takes the model of the system in question and the property specification expressed in some probabilistic temporal logic like PCTL (Probabilistic Computation Tree Logic) as the inputs and verify the properties of the system. The transitions in the model of the system are based on the probability expressed in the transition matrix of the model.
For further reading on model checking, readers may refer to chapter 1 and 10 of the book [22].

3.3.2 Statistical Model Checking

Model checking of stochastic systems can be done either using numerical or statistical techniques. While numerical model checking algorithms generate the exact probability of a property being true in a given system, they often do it at the cost of time and memory needed. For the large and complex systems, use of numerical model checking causes state explosion problem and although the results generated are accurate but the time is taken to verify the system is too large.

Statistical model checking, on the other hand, requires lesser time and memory for the computation but generates results with some confidence parameter. Statistical model checking is based on simulating finitely many executions of the system and using various statistical techniques to infer whether a property is satisfied or not. Also, these execution paths or traces generated through the simulations can be used to estimate the probability of the property being satisfied in the system. Seminal work done on statistical model checking can be found in [23, 24, 25, 26]. Authors in [27, 28, 29] survey the recent advancements in the area of statistical model checking.

There are mainly two techniques used in statistical model checking viz. Hypothesis Testing and Estimation methods. Hypothesis testing is used to verify whether the probability of a property being true crosses the required threshold probability or not. Estimation methods, on the contrary, estimate the probability of the given property to be true.

Hypothesis Testing

In hypothesis testing, a hypothesis $H : \ p \geq \theta$ is tested against $K : \ p < \theta$ where p is the probability of any formula ϕ being satisfied by the model. As discussed above, the results produced by this technique are not exact but the likelihood of error happening can be bounded using the parameters α and β. These parameters determine the strength of the test and limit the error probability, the probability of accepting K when H holds is bounded by α and is called Type-I error. Similarly, the probability of accepting H when K holds is bounded by β and is called Type-II error. In the ideal case, we would like to have a really low probability for Type-I and Type-II error but the nature of these errors makes it impossible to do that. To resolve this issue, an indifference region (p_0, p_1), where $p_0 \geq p_1$ is introduced. Now the hypothesis $H_0 : p \geq p_0$ is tested against $H_1 : p \leq p_1$. If the value obtained is between $p_0 and p_1$, then it lies in the indifference region and we are not sure which hypothesis is accepted. We will now look at the two methods for hypothesis testing proposed by Younes in [25].

Single Sampling Plan (SSP) : In this method, for n simulations of the model, the value is accumulated and it crosses a constant c then H_0 is accepted, otherwise

H_1 is accepted. The difficult task in this method is to find the values of n and c such that the error bounds are also low and also the testing does not take too long. There are some techniques proposed by Younes to find the optimal values of n and c in [25].
Sequential Ratio Probability Test (SPRT) : In SSP the sample size, n is fixed and the individual observation is not taken into account. In SPRT , two numbers a_1 and a_2 are chosen and with every observation the ratio $\frac{p_{1m}}{p_{0m}}$ is calculated and if the ratio is greater then a_1, then H_0 is accepted, or if the ratio is less than equal to a_2, then H_1 is accepted, otherwise the next sample is drawn for the model. So, in this, there is no fixed number of samples and it is decided with each sample whether the next sample is required or not.

Estimation Methods

Hypothesis testing methods only check whether the hypothesis is accepted or not within the specified error bounds. But to estimate the probability of the property being satisfied in the model, estimation methods are used which based on the simulations drawn from the model estimates the probability. In [30], the author has discussed an approach based on approximate model checking. Based on the Chernoff-Hoeffding, a lower bound for the number of samples, N required to estimate the probability is derived.

$$N \geq \frac{ln(\frac{2}{\delta})}{2\varepsilon^2} \tag{3.1}$$

where δ is the confidence parameter and ε is the approximation parameter.
Once we get the value of N, the number of samples required of the system, we keep the count of samples in which the property is satisfied and the ratio of samples in which the property is satisfied to the total number of samples gives the probability of property being true in the system within the approximation bounds.
A detailed discussion of these techniques can be found in [30, 25].

3.4 Random Graph Models

In this section, we briefly describe the graph models used in the experiments run on our model.

3.4.1 k-Regular Graphs

The graph with the same degree for all its vertices is known as the regular graph. If the degree for all the vertices is k, then the regular graph is called the k-Regular graph and is denoted by G_K. Now there are various possibilities of k-regular graph for any number of nodes, n. So for n nodes and a certain value of K, randomly a graph is picked from all possibilities of the k-regular graphs for n nodes and is denoted as $G(n, K)$.

A regular graph can be both directed as well as undirected but in case of directed graphs, a strict condition should be satisfied where the out degree and in degree of all the nodes has to be equal.

3.4.2 Erdös-Rényi (ER) Model

This model is used for the generation of the random graphs, named after Paul Erdös and Alfréd Rényi. It has two closely related models to generate the network.

G(n,m) : In this model, from the collection of all the graphs, a graph is chosen uniformly at random with n nodes and m edges.

G(n,p) : In this model, each edge is included with a probability p. Inclusion of every edge in the graph is independent of any other edge.

Approximately in both the models, expected number of edges are same.

3.4.3 Barabási Albert (BA) Model

This model is used for the generation of random scale-free networks, named after Albert-László Barabási and Réka Albert. The preferential attachment mechanism, which is the process of distributing things among entities based on what they already have, is used for the generation of the network.

Generation of the network starts with the connected graph of initial m nodes. Now, nodes are added to the graph one at a time and each new node is connected to n ($\leq$m) existing nodes. The probability of an edge from a new node to an existing node depends on the number of edges existing node already have. Formally,

$$p_i = \frac{k_i}{\sum_j k_j}, \tag{3.2}$$

where k_i is the degree of node i and denominator denotes the sum of the degree of all the existing nodes.

BA model is considered to represent the real networks in a better way than the ER model generated networks.

3.4.4 Street Network

We have also run some experiments for the real street networks in our cops and robber setting. To generate the street network, we have used a Python library OSMnx. Through this library, we can easily generate and visualize the complex real street networks in the form of graphs. Latitude and longitude information is used to generate the networks. For further reading about the OSMnx library and other functions available, readers may refer to [31].

Chapter 4

Framework

In this chapter, we will discuss the details of our model framework. Firstly, we will give the theoretical details of our model and then will give the implementation details of our modeling.

4.1 Theoretical Framework

4.1.1 Problem Statement

Given

1. a directed graph

2. an initial placement of n_r robbers and n_b blue cops, n_w white cops and n_g green cops on the vertices of the graph

3. transition functions defining the movement of the cops and robbers on the graph and,

4. capturing condition i.e., how many cops of each color are needed to be there on the same vertex as that of the robber for capture, c_g, c_w, and c_b.

Find:

1. What is the probability that all the robbers are caught within k steps?

2. What is the cop number to ensure capture within k steps?

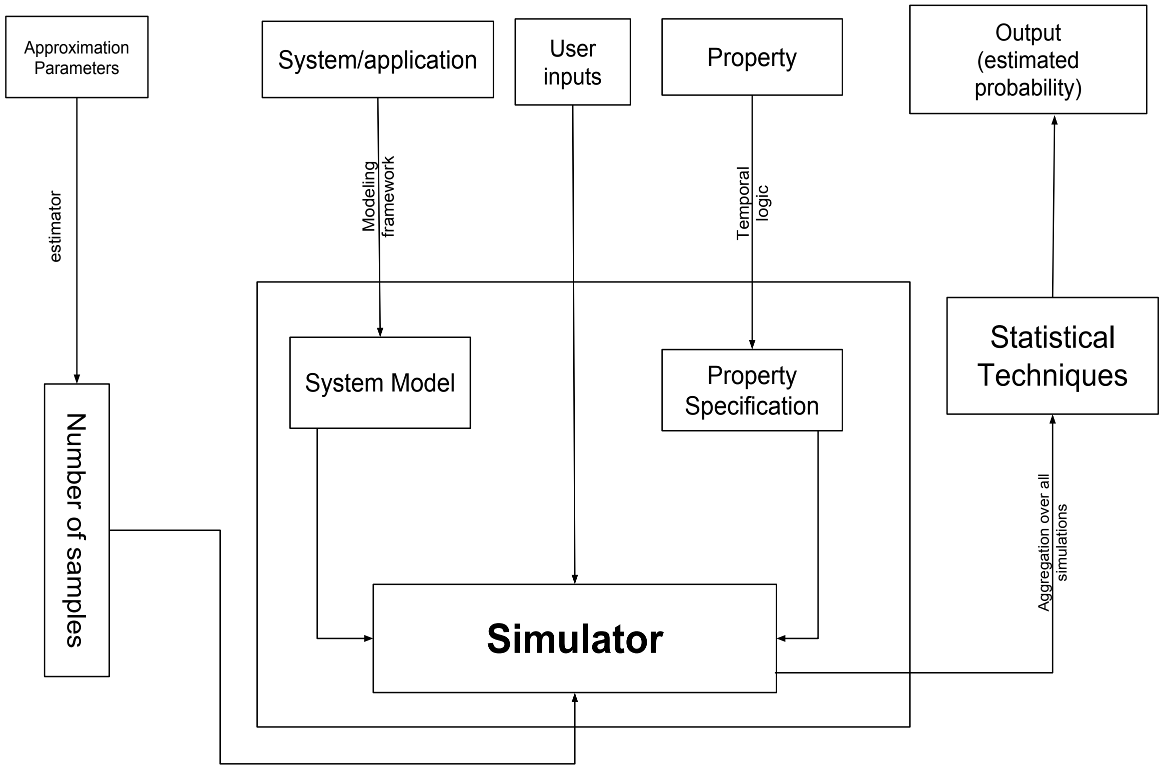


Figure 4.1: Architectural Framework of SMCA Tool

4.1.2 SMCA Tool

Statistical Model Checker for Agent based systems (SMCA) tool implements our model for the group of agents performing the random walk in the network. Agent based modeling of the real world systems has been gaining popularity over the past few years. In our modeling of cops and robbers game, we have presented the multi-colored cops as the different types of agents and the robber as the other type of agent. Now in our example, collectively 3 types of cops agent are trying to capture the robber agent.

Fig. 4.1 shows the architectural framework for SMCA tool which is consists of several different components. We briefly describes the component of the tool as follows:

1. **System Model:** This component of the tool takes input from the user in the form of the JAVA code which tells the model on how the various agents move in the network in every time step. This is given as input by the user in the pre-described format of the tool and can be changed as per the requirements of the user based on their system.

2. **Property Specification:** It specifies the property user wants to check over the modeling of the system and is given as the input in the pre-defined format by the user. This specification should basically define on what parameter should be checked over all the simulations of the model for all the agents in the network.

3. **Simulator:** It is one of the important components of SMCA tool which simulates the given model for the modeling of the system given by the user and the property specification also provided by the user as the inputs to the simulator. It simulates the system for the required number of samples based on the bound over errors and provides the output in the form of estimated probability.

4. **Statistical Techniques:** This component has options for the statistical techniques to be applied to estimate the probability of the property being satisfied in the model with the given constraints. This component takes into account all the simulations and produces the result in the form of estimated probability.

In the next section, we will further describe the implementation details of SMCA tool for agent based modeling with our setting of cops and robbers.

4.2 Implementation Details

The total number of agents is denoted by n and indexed by i. An agent is described completely by a tuple of variables $\mathbf{v_i} = \langle \mathbf{v_1}, \mathbf{v_2}, \ldots \mathbf{v_k} \rangle_\mathbf{i}$. Each variable has an associated domain from which it takes values. An assignment of values to these variables is called a *valuation*. There is a type associated with every agent.

The instantaneous description at time t of an agent i is given by the valuation of each of the variables at that instant: $v_i(t) = \langle v_{i1}, v_{i2}, \ldots v_{ik} \rangle_i(t)$. We denote the set of all possible tuples (valuations) by S. We now define the input format to the simulator generator. This input essentially describes the model:
$\mathbf{T}_j : n_j : \mathbf{MR}_j : \mathbf{init}_j$, $1 \leq j \leq k$, where
$\mathbf{T}_j$ is the name of the type of agent, n_j is the number of agents of type $\mathbf{T}_j$, $\mathbf{MR}_j : S \to S$ and $\mathbf{init}_j$ is a function that assigns an initial valuation v_i for each agent of type $\mathbf{T}_j$.
Therefore the modeling segment of the input file would look like, for example:
M:
$\mathbf{T}_1 : 100 : \mathbf{MR}_1 : \mathbf{init}_1$
$\mathbf{T}_2 : 50 : \mathbf{MR}_2 : \mathbf{init}_2$

Examples of procedures **MR** and **init** will be detailed in the next sections. Constructing a PBLTL input formula involves first defining atomic propositions of the form $x \sim \nu$.

The tool provides functions that can be used to construct atomic propositions. The functions that are currently supported are:
Type$[i]$: returns the Type of agent i.
$\mathbf{V}_l[i](t)$: returns $v_{il}(t)$–the value of the l^{th} variable of agent i at the t^{th} time step.
$\mathbf{A}(l, y, t)$: returns a list of all agents that have the value of their l^{th} variable equal to y at time instant t.
The tool supports construction of atomic propositions of the following type:
$\sum_{i \in I} \sum_{k \in K} \sum_{t \in T} \mathbf{V}_k[i](t) \sim \nu$ for some $\nu \in \mathbb{R}$.
The sets I, K, and T are populated through user-defined procedures:
populate$I$, to populate the set I of agent indices.
populate$K$, to populate the set K of agent variables.
populate$T$ to populate the set T of time indices.

Atomic propositions can then be combined in accordance with the syntax to obtain larger and more complex PBLTL queries.
To summarize, the query segment of the input file would look like, for example:
Q:
$\mathbf{A}_1$: **populate**I_1 : **populate**K_1 : **populate**T_1;
$\mathbf{A}_2$: **populate**I_2 : **populate**K_2 : **populate**T_2;
$Pr_{\geq 0.3}(\mathbf{A}_1 U \mathbf{A}_2)$

Populate$J(t)$:
$\mathbf{V}_1[Robber](t)$
$J \leftarrow \{x\}$
Populate$I(t)$: (1 type cops)
for all $i \neq Robber$
if $\mathbf{V}_1[i] == x$
then $I \leftarrow I \cup i$.
Populate$I(t)$: (3 type cops)
for all $i \neq Robber$
if $\mathbf{V}_1[i] == x$
then $I \leftarrow I \cup i$.

$I \leftarrow distinct\_type(I)$

$\mathbf{A}_1 : \{\mathbf{populate}I(t) : \mathbf{populate}J(t)\} \geq 3$

$Pr_{\geq\theta}(\mathbf{True}\ U\ \mathbf{A}_1)$

where, k is the number of simulation time steps, *isCaught* is the atomic proposition in our cops and robber model that represents the condition where the robber is captured.

Chapter 5

Results

In this chapter, we describe the experiments and discuss the results we achieved on different random graph models. We describe the results based on the different graph models and across all the experiments cops and robbers are performing a random walk in the graph.

5.1 k-Regular Graphs

For the k-Regular graphs, our experiments were mainly based on the changing the value of k across the graphs and to analyze how our model and statistical model checking methods would respond to that. We have performed two different set of experiments for the k-Regular graphs wherein one set we took 5 robbers performing a random walk in the network and in the other set number of robbers are 10. The number of nodes across all the experiments is set to 500 and the number of cops in the network is set to 300. In 3 colors cops setting, at least one of each color is required at robber's vertex at the same time to capture the robber whereas in 1 color cop setting 3 of the same color are required. More than 1 robber can be captured at the same vertex in a single time unit.

Query : For k-Regular graphs, we estimate the probability of all the robbers getting caught within the given time bound. Time bound is taken as 150 across all the experiments on k-Regular graphs. Formally the query can be expressed as follows:

$$Pr_{\geq\theta}[TRUE \cup^{\leq k} allCaught]$$

where, θ is the estimated probability for the property to be true in the system, k

is the time bound with in which the property will be verified and *allCaught* is an atomic proposition described in the model.

5.1.1 5 robbers

In this experiment, we considered 5 robbers moving in the network trying to escape the cops indefinitely and estimated the probability of all the robbers getting caught with in the specific time bound. We experimented with two variants of cops described above as the 3 color cops and 1 color cop. Table 5.1 shows the probability values obtained for different values of K in k-Regular graphs.

The interesting observation through this experiment is that the probability value of

K values	3 colors cops	1 color cop
50	0.0738	0.8473
100	0.0754	0.8539
150	0.0727	0.8507
200	0.0722	0.8535

Table 5.1: Probabilities for different values of K for 5 robbers

robber getting caught is almost unaffected with the changing K values which tells us the the probability of all the robber getting caught does not depend on how dense the graph is. Moreover, as expected the probability values in case of 1 color cop are higher than for 3 color cops because in 3 color cops scenario we need the combination of 3 different colored cops at the same vertex as that of robber to catch him which is less likely to happen than the 3 same color cops being present at same vertex in case of 1 color cops.

5.1.2 10 robbers

In this experiment, 10 robbers are considered to be performing the random walk in the graph. The idea of this experiment is to verify the results for previous experiments with 5 robbers and also to see how the probability values for the property checked changes with the increase in the number of robbers in the network.

Table 5.2 shows the probability estimation results for 10 robbers experiment. It is clear from the results that the probability values does not depend too much on the value of K in k-Regular graphs, rather it depends on the number of cops chasing the robbers. Also, as expected with the increase in the number of robbers in the network,

K values	3 colors cops	1 color cops
50	0.0057	0.7194
100	0.0062	0.7249
150	0.0054	0.7253
200	0.0057	0.7217

Table 5.2: Probabilities for different values of K for 10 robbers

it has become difficult to catch all the robbers with in the given time bound.
The results obtained for the experiments on k-Regular graphs shows that the value of K does not have a pronounced effect on the probability of robber getting caught. This also agrees with the result that Cooper et al [32] derived for single colored cops.

5.2 ER Model & BA Model

The set of experiments performed on ER model and BA model networks are same. So, in this section we will first describe the set of experiments done and then will discuss the results of all the experiments for both the graph models.
Query :

$$Pr_{\geq\theta}[TRUE \cup^{\leq k} isCaught]$$

where θ is the estimated probability for the property to be true in the system, k is the time bound within which property will be verified and $isCaught$ is an atomic proposition described in the model.

5.2.1 Varying the number of cops

In this experiment, we wish to analyze the impact of changing the number of cops performing random walks in the network. In this experiment, only one robber is walking randomly in the network. Both the cops and the robber are unaware of each other's positions. The objective here is to estimate the probability of robber getting caught within the given time bound. This experiment has been conducted for both the variants i.e., 3 color cops and 1 color cops. As discussed in the previous chapter, to capture the robber in 3 color cops scenario we need at least 1 cop of each color to be the same vertex as that of the robber at the same time whereas in 1 color cops scenario 3 cops are required to capture the robber on the same vertex as that of the robber at the same time instant. Nodes in the network are 500 and the number of edges in the network is 61,915 for both the graph models. The number of cops in the network varies from 60 to 420 in this experiment and the time bound is fixed to 150 in all the runs of this experiment. The number of cops is evenly distributed in 3 color cops layout i.e., the number of cops of each type is equal whereas, in 1 color cops scheme, all the cops are of the same color.

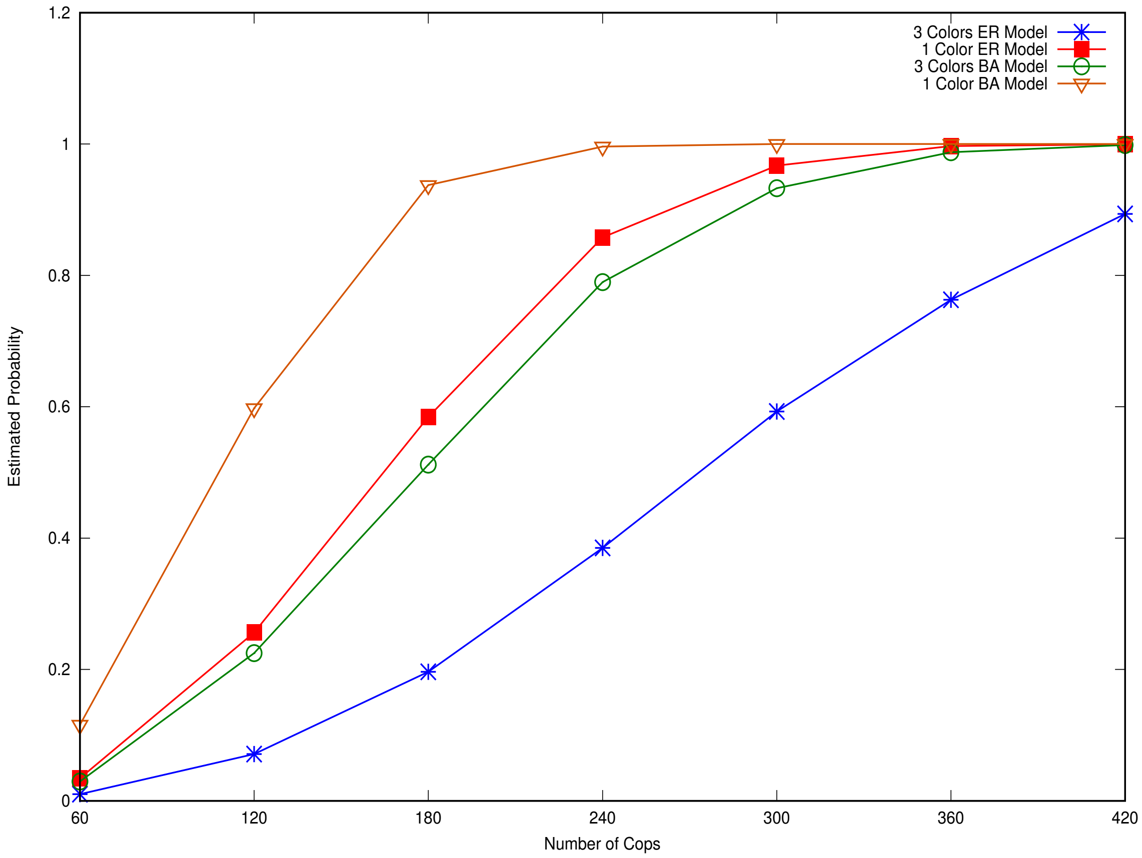


Figure 5.1: Estimated probability varying the number of cops

Fig. 5.1 shows the comparison of the estimated probability in this experiment for both the graph models. As seen from the results, the probability of robber getting caught increases with the increase in the number of cops as with more cops in the network it has become difficult for the robber to escape the capture. Also as intuitively expected, estimated probability values for 1 color cops scenario are more than the 3 colors cops scenario because the condition to capture the robber is stricter in 3 color cops scheme.

The interesting observation, however, is the difference in the probability values between the ER model and the BA model. Values acquired for BA model are quite higher than for ER model which shows the effect of the graph structure on the probability values of the robber getting captured.

5.2.2 Varying the time bound

In this experiment, we analyze the impact of the time bound specified to check the property over the cops and robbers model. We change the time bound value in this experiment to see whether the time bound specified plays a significant role on the probability of capturing the robber. This experiment is also run for both 3 colors cops and 1 color cops scheme in both the graph models. The number of nodes in the graphs is fixed to 500 with 61,915 edges in the network. The number of cops is set to 300 for all the simulations in this experiment. In 3 colors cops layout, 300 cops are evenly divided for each color i.e., 100 cops if each color while in 1 color cops scheme, all 300 cops are of the same color. Value of the time bound in this analysis varies from 30 to 210.

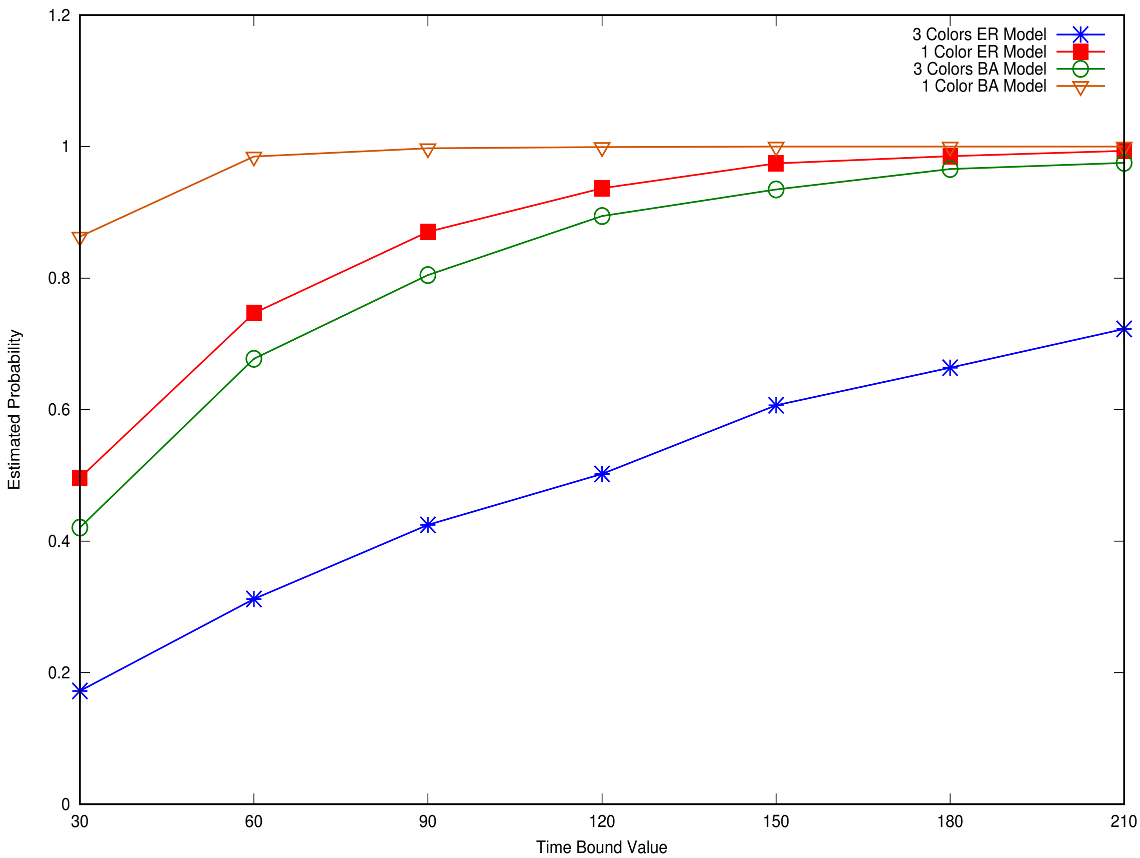


Figure 5.2: Estimated probability varying the time bound value

The results of this experiment are shown in Fig. 5.2 which clearly indicates that with the increase in the value of time bound, probability of capturing the robber increases, but for the higher time bound values probability tends to be very close to 1.0 which shows that with high time bound limit in the experiments, it is becoming difficult for the robber to escape the capture indefinitely even when the cops are performing random walks in the graph. Also as observed in the experiment with varying number of cops, in this experiment also the probability values for BA model are significantly higher than the probability estimated for ER model, thus it again proves that capturing of the robber definitely depends on the topology of the network. Also, as expected the probability of robber getting caught is higher in 1 color cops setting.

5.2.3 Varying the number of edges in the network

Through this experiment, we analyze whether the size of the network where the robber and cops are executing their random walks, have any effect on the chances of the robber getting captured. Thus in this analysis, we vary the number of edges in the network and estimate the probability of apprehending the robber within the specified time bound. The time bound value is specified as 150 for all the simulations in this experiment. The number of cops in this experiment is fixed to 300 which are divided equally in 3 colors cops setting whereas all 300 are of the same color in 1 color cops setting.

Fig. 5.3 demonstrates the results of the probability estimation by varying the size of the network in our model. Against our intuition, it can be observed from the results that changing the size of the network does not have a major effect on the probability of apprehending the robber in the network generated through ER model. Although for 3 color cops setting in BA model, the probability decreases after a point if the size of the graph is increased, but 1 color cops setting, estimated probability almost remains same. This experiment also shows that in BA model it is more likely to catch the robber than in ER model and also that it is easier to catch the robber in 1 color cops setting.

5.2.4 Multiple cops & multiple robbers

In this experiment, we have considered more than 1 robber in the graph for ER and BA models and observe the probability estimation values for them. The objective here is to estimate the probability all the robber getting captured by the cops within

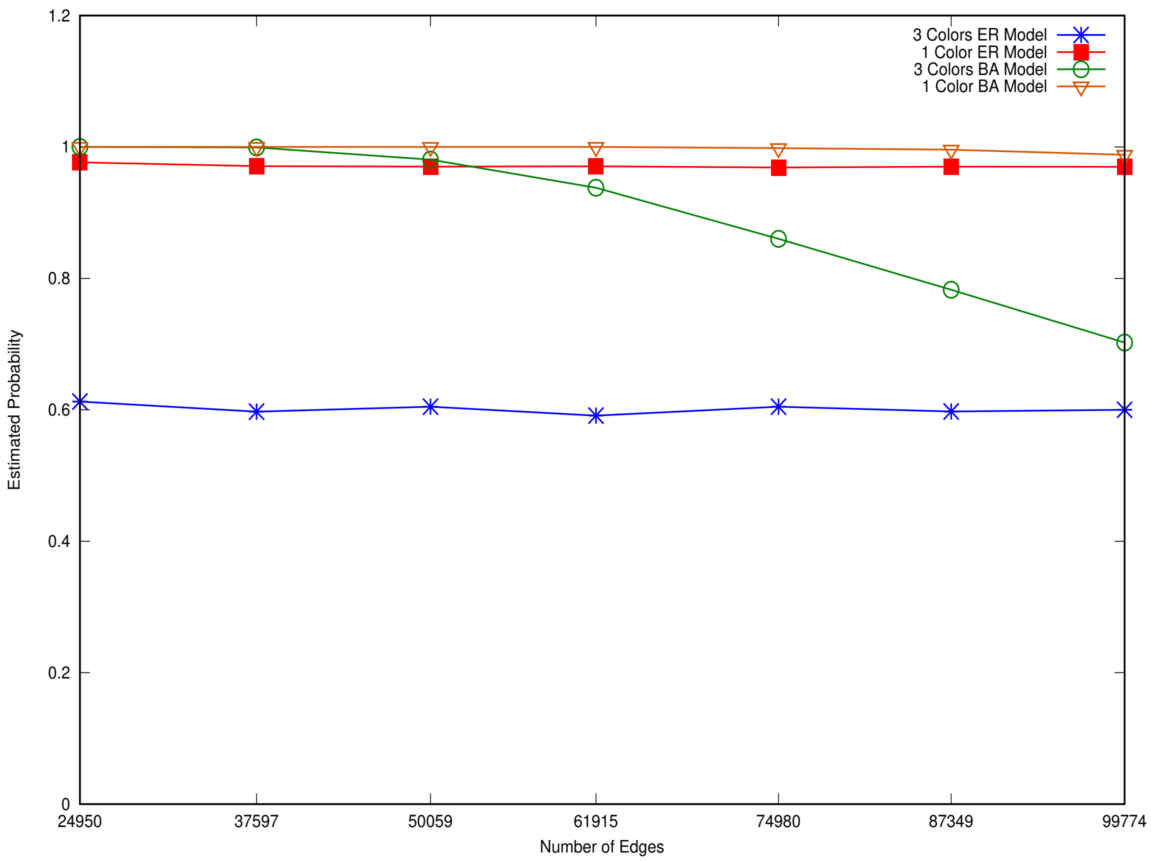


Figure 5.3: Estimated probability varying the number of edges in the network

the specified bounded time. This experiment has only been done for 3 color cops setting with 300 total number of cops. Each color cops are 100 in color and to catch a single robber we need at least one of each color on the robber's vertex at the same time vertex. More than 1 robber can be caught at the same vertex in a single time unit.

No. of Robbers	ER Model	BA Model
5	0.072	0.72
10	0.0034	0.529
20	0.0	0.2834
30	0.0	0.1454
40	0.0	0.0731
50	0.0	0.0456

Table 5.3: Estimated probability for multiple robbers in ER and BA graph models

Table 5.3 shows the values of probability estimation of capturing all the robbers in the network. It can be observed from the results that in ER model when the

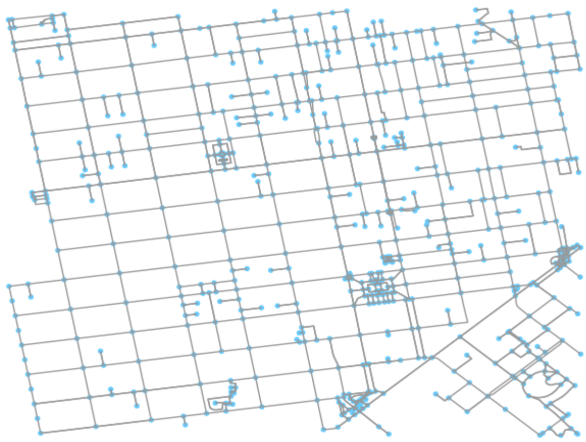


Figure 5.4: Graph for street network

number of robbers crosses a certain number, probability of capturing becomes 0, In BA model networks, the probability value decreases as the number of robber increases but is still significantly higher than ER model. Thus, this again shows that BA model offers better chances of capturing the robber in cops and robber scenario.

5.3 Street Network

In this section we will discuss the results of the experiments run on the real street network generated through OSMnx library. For this experiment we have used the graph of a location in San Francisco, U. S. A., fig. 5.4 shows the graph we have used for the experiment on the street network.

This street graph has 654 nodes and 1400 edges. Some edges in the network are unidirectional and others are bidirectional. Our motive here is to estimate the cop number, i.e., the minimum number of cops required to catch the robber surely within the given time limit.

Number Cops required to catch robber	Estimated Cop Number
1	182
2	762
3	1385

Table 5.4: Estimated Cop Number

Table 5.4 shows the estimated Cop Number in the street network considered for simulations when different number of cops are required at the robber's location at the same time instant to catch the robber.

Chapter 6

Conclusion

In this work, we have studied a variation of cops and robber problem where multi-colored are performing random walks in the network and aim to apprehend the robber in the network. In our work, we have studied our model on various random graph models and analyzed the effect of changing different parameters in the network. We have used statistical model checking methods to estimate the probability of all the robbers getting captured within the given specified time limit. As discussed in our results, it is clear that it is easier to catch the robbers in BA graph model than ER model. Also, in random k-Regular graphs, the chances of catching the robber does not really depend on the K value of the graphs.

6.1 Future Scope

We believe that it would be interesting to investigate the cops and robbers problem with more complex cop/robber behavior using statistical model checking. The technique will prove particularly useful when obtaining closed form solutions is difficult, and complex queries have to be asked of the system. To that end, we are in the process of building a statistical model checker optimized for generic versions of the cops and robbers problem.

References

[1] M. Kot. The state explosion problem. *Retrieved May* 18, (2003) 2015.

[2] H. L. Younes, M. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer* 8, (2006) 216–228.

[3] T. Chen, M. Kwiatkowska, D. Parker, and A. Simaitis. Verifying team formation protocols with probabilistic model checking. In International Workshop on Computational Logic in Multi-Agent Systems. Springer, 2011 190–207.

[4] A. Quilliot. Problemes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes. *These d'Etat, Université de Paris VI* 131–145.

[5] R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics* 43, (1983) 235–239.

[6] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics* 8, (1984) 1–12.

[7] A. Kehagias and P. Prałat. Some remarks on cops and drunk robbers. *Theoretical Computer Science* 463, (2012) 133–147.

[8] C. Moldenhauer and N. R. Sturtevant. Evaluating Strategies for Running from the Cops. In IJCAI, volume 9. 2009 584–589.

[9] A. Bonato, E. Chiniforooshan, and P. Prałat. Cops and robbers from a distance. *Theoretical Computer Science* 411, (2010) 3834–3844.

[10] F. V. Fomin, P. A. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theoretical Computer Science* 411, (2010) 1167–1181.

[11] A. Kehagias, D. Mitsche, and P. Prałat. Cops and invisible robbers: The cost of drunkenness. *Theoretical Computer Science* 481, (2013) 100–120.

[12] V. Isler and N. Karnad. The role of information in the cop-robber game. *Theoretical Computer Science* 399, (2008) 179–190.

[13] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking. Randomized pursuit-evasion in graphs. *Combinatorics, Probability and Computing* 12, (2003) 225–244.

[14] D. Bal, A. Bonato, W. B. Kinnersley, and P. Pralat. Lazy cops and robbers played on random graphs and graphs on surfaces. *International Journal of Combinatorics* 7, (2016) 627–642.

[15] S. Finbow and G. MacGillivray. The firefighter problem: a survey of results, directions and questions. *Australasian Journal of Combinatorics* 43, (2009) 6.

[16] J. H. Conway. The angel problem. *Games of no chance* 29, (1996) 3–12.

[17] A. Bonato and F. Mc Inerney. The Game of Wall Cops and Robbers. In Computational Intelligence, Cyber Security and Computational Models, 3–13. Springer, 2016.

[18] M. Boyer, S. El Harti, A. El Ouarari, R. Ganian, T. Gavenciak, G. Hahn, C. Moldenauer, I. Rutter, B. Thériault, and M. Vatshelle. Cops-and-robbers: remarks and problems. *Journal of Combinatorial Mathematics and Combinatorial Computing* 85.

[19] A. Bonato. The game of cops and robbers on graphs. American Mathematical Soc., 2011.

[20] A. Pnueli. The temporal logic of programs. In Foundations of Computer Science, 1977., 18th Annual Symposium on. IEEE, 1977 46–57.

[21] N. Kamide. Bounded linear-time temporal logic: A proof-theoretic investigation. *Annals of Pure and Applied Logic* 163, (2012) 439–466.

[22] C. Baier and J.-P. Katoen. Principles of model checking. MIT press, 2008.

[23] K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In International Conference on Computer Aided Verification. Springer, 2004 202–215.

[24] K. Sen, M. Viswanathan, and G. Agha. On Statistical Model Checking of Stochastic Systems. In Computer Aided Verification, 17th Intl. Conf., CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005, Proc. 2005 266–280.

[25] H. L. Younes. Verification and planning for stochastic processes with asynchronous events. Technical Report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE 2005.

[26] H. L. S. Younes and R. G. Simmons. Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling. In Computer Aided Verification, 14th Intl. Conf., CAV 2002,Copenhagen, Denmark, July 27-31, 2002, Proc. 2002 223–235.

[27] A. Legay, B. Delahaye, and S. Bensalem. Statistical Model Checking: An Overview. In H. Barringer, Y. Falcone, B. Finkbeiner, K. Havelund, I. Lee, G. Pace, G. Roşu, O. Sokolsky, and N. Tillmann, eds., Runtime Verification, volume 6418 of *Lecture Notes in Computer Science*, 122–135. Springer Berlin Heidelberg, 2010.

[28] G. Agha and K. Palmskog. A Survey of Statistical Model Checking. *ACM Trans. Model. Comput. Simul.* 28, (2018) 6:1–6:39.

[29] A. Legay, B. Delahaye, and S. Bensalem. Statistical model checking: An overview. In International Conference on Runtime Verification. Springer, 2010 122–135.

[30] V. Nimal. Statistical approaches for probabilistic model checking. Ph.D. thesis, University of Oxford 2010.

[31] G. Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems* 65, (2017) 126–139.

[32] C. Cooper, A. Frieze, and T. Radzik. Multiple random walks and interacting particle systems. In International Colloquium on Automata, Languages, and Programming. Springer, 2009 399–410.