CELLULAR AUTOMATA INSPIRED STOCHASTIC OPTIMIZATION ALGORITHMS FOR MITIGATING DDoS IN VANETs

A Thesis

submitted to Pondicherry University in partial fulfillment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

By

DEEPA THILAK K

(Reg. No. 15RCS0008)

Under the guidance of

Dr. A. AMUTHAN

Professor
Department of Computer Science and Engineering
Pondicherry Engineering College

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PONDICHERRY ENGINEERING COLLEGE
PUDUCHERRY – 605 014
INDIA

JANUARY 2018

Dr. A. AMUTHAN, B.Tech., M.E., Ph.D.,
Professor, Department of Computer Science and Engineering
Pondicherry Engineering College
(An Autonomous Institution of the Government of Puducherry)
Puducherry – 605 014

CERTIFICATE

Certified that this thesis entitled **"CELLULAR AUTOMATA INSPIRED STOCHASTIC OPTIMIZATION ALGORITHMS FOR MITIGATING DDoS IN VANETs"** submitted for the award of the degree of **DOCTOR OF PHILOSOPHY** in **COMPUTER SCIENCE AND ENGINEERING** of the Pondicherry University, Puducherry is a record of original research work done by **Ms. DEEPA THILAK. K** during the period of study under my supervision and that the thesis has not previously formed the basis for the award of any other degree by this or any other University/Institution. This thesis represents independent work on the part of the candidate.

Supervisor,

Place: Puducherry
Date: 16.01.2018 **(Dr. A. AMUTHAN)**

DECLARATION

I hereby declare that this thesis entitled **"CELLULAR AUTOMATA INSPIRED STOCHASTIC OPTIMIZATION ALGORITHMS FOR MITIGATING DDoS IN VANETs"** submitted to the Pondicherry University, Puducherry, India for the award of the degree OF **DOCTOR OF PHILOSOPHY** in **COMPUTER SCIENCE AND ENGINEERING** is a record of bonafide research work carried out by me under the guidance and supervision of **Dr. A. AMUTHAN,** Professor, Department of Computer Science and Engineering, Pondicherry Engineering College and that this has not formed the basis for the award of any other degree by any University/Institution before.

Place : Puducherry

Date :16.01.2018

(DEEPA THILAK K)

ACKNOWLEDGEMENTS

I would like to express my deep sense of respect and gratitude towards my supervisor **Dr. A. Amuthan,** Professor and Associate Dean (Autonomy and Accreditation), Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, who has been the guiding force behind this work. I thank him for introducing me to the field of Vehicular Ad-hoc Networks and giving me the opportunity to work under his supervision. Without his invaluable advice and assistance, it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life.

I express my sincere thanks to **Dr. P. Dananjayan,** Principal, Pondicherry Engineering College**, Dr. D. Govindarajulu**, Former Principal, Pondicherry Engineering College, Principal, Pondicherry Engineering College for their support and permission to carry out the research work. I thank **Dr. M. Sugumaran**, Professor and Head, and **Dr. D. Loganathan**, Professor and Former Head, Department of Computer Science and Engineering for their continuous encouragement and valuable advice.

I would like to express my deep and sincere gratitude to the Doctoral Committee members, **Dr. K. Vivekanandan,** Professor and Dean(Students), Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, and **Dr. K. Suresh Joseph,** Assistant Professor, Department of Computer Science, Pondicherry University, Puducherry, for their insightful comments and encouragement to complete the research.

I also thank all the faculty members of the Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry for their valuable suggestions and support throughout my research work.

K. DEEPA THILAK

ABSTRACT

Vehicular Ad-hoc Networks (VANETs) is a distributed and self-organized network that has evolved as a new powerful technology for enabling safe and comfortable driving. Numerous challenges related to VANET resources have invited potential researchers to focus on resolving issues that include routing, broadcasting, quality of service and security. Among the challenges, efficient routing in VANET is the most important issue that needs to be resolved for sending and receiving vital information between the vehicular nodes of the network in a timely manner. Routing attacks like Denial of Service attack, Black Hole attack, Wormhole attack, Sinkhole attack and Sybil attack drastically influence and degrade the availability factor of VANET resources. Distributed Denial of Service (DDoS) attack is the most crucial attack among the availability parameter influencing attacks in VANET that uses the group of malicious nodes to deny the services from a victim node. Collaboration among the neighbouring vehicles is important for reliable routing of packets from the source to destination. But the malicious activity of each neighbouring vehicle makes the task difficult and prevents the packet from reaching the destination node. Thus reliable neighbouring nodes of the interacting vehicular nodes need to be determined for mitigating the impact of DDoS attacks. The problem of finding the reliable neighbour in a large, dynamic, unstable topology network for routing the data is similar to finding the optimized path in a large network. Thus the selection of optimal neighbour is considered as the optimization problem which can be solved by applying various approximate solutions. Meta-Heuristics Stochastic Optimization algorithms are more preferable than heuristic approaches for their strong theoretical background, experimental confirmation and effective performance. Furthermore, the exploitation and the exploration levels of searching enabled by Meta-Heuristic Stochastic Optimization algorithms are confirmed to be quite significant.

In this research, existing Meta-Heuristic Stochastic Optimization algorithms such as Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) and Tabu Search (TS) are phenomenally improved for finding the global best solution that plays a vital role in mitigating DDoS attacks. The proposed variant algorithms of ACO, ABC and TS such as Cellular Automata-based Improved Ant Colony Optimization Algorithm (CA-IACOA), Cellular Automata-based Improved Artificial Bee Colony Algorithm (CA-IABCA), Cellular

Automata-based Modified Artificial Bee Colony Algorithm (CA-MABCA), and Cellular Automata-based Modified Tabu Search Algorithm (CA-MTSA) integrate cellular automata with the respective meta-heuristics stochastic algorithms for improving its efficiency under DDoS mitigation.

These proposed CA-IACOA, CA-IABCA, CA-MABCA, and CA-MTSA algorithms use vehicle's reference velocity, reliability and trust factor for evaluating the fitness values of each vehicular node of the network. The quantified fitness value of vehicular nodes aids the proposed algorithms to compare the co-operative behaviour of vehicular nodes and choose an optimal node among them for mitigating DDoS nodes of the network.

The proposed algorithms are evaluated by simulation and analytical method. Then the proposed and the compared benchmark algorithms are compared using evaluation metrics like prediction variance, average prediction variance, packet delivery ration, and end-to-end delay metrics for testing the significance of the proposed CA-IACOA, CA-IABCA, CA-MABCA, and CA-MTSA algorithms. From the result analyses, it is found that the proposed algorithms give better packet delivery ratio and less prediction variance compared to the existing algorithms. The analytical testing of the proposed algorithms with standard benchmark functions also inferred better rate of convergence, precision, robustness and performance in increased dimensions.

TABLE OF CONTENTS

Chapter No.	Title	Page No.

Chapter No.	Title		Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

VANETs	Vehicular Ad hoc Networks
DSRC	Dedicated Short Range Communication
WAVE	Wireless Access in Vehicular Environment
DoS	Denial of Service
DDoS	Distributed Denial of Service
OBU	On Board Unit
RSU	Road Side Unit
CA	Cellular Automata
1 D	One Dimension
2 D	Two Dimension
FCC	Federal Communication Commission
GSM	Global System for Mobile
UMTS	Universal Mobile Telecommunications System
Wi-Max	Worldwide Interoperability for Microwave Access
V-V	Vehicle to Vehicle
V-I	Vehicle to Infrastructure
MANET	Mobile Ad-hoc Network
AODV	Ad-hoc On-Demand Distance Vector
ACO	Ant Colony OPtimization
IACO	Improved Ant Colony Optimization
ABC	Artificial Bee Colony
IABC	Improved Artificial Bee Colony
TS	Tabu Search
GEM	Grenade Explosion Method
CO	Cauchy Operator
CA-IABCA	Improved Stochastic Optimization Algorithm Based Grenade Explosion Method and Cauchy Operator

DE	Differential Evolution
COBL	Chaotic and Opposition Based Learning
CA-MTSA	Improved Tabu Search Based Cellular Automata
CA-IACOA	Cellular Automata based Improved Ant Colony optimization Algorithm
CA-IABCA	Cellular Automata based Improved Artificial Bee Colony Algorithm
CA-MABCA	Cellular Automata based Variant Artificial Bee Colony Algorithm
UV-CAST	Urban Vehicular BroadCAST
ZoI	Zone of Interest
PSO	Particle Swarm Optimization
CAPSO	Cellular Automata based Particle Swarm Optimization
IPCPSO	Improved Probabilistic Cellular automata based Particle Swarm Optimization
LDDOS	Low-rate Distributed Denial of Service
DDIACS	Distributed Detection and Identification Ant Colony System
RoC	Rate of Change
SMT	Steiner Minimum Tree
MABC	Micro Artificial Bee Colony
QoS	Quality of Service
PS-ABC	Prediction and Selection based Artificial Bee Colony
LSACA	Local Search Improved Ant Colony Algorithm
MACO	Mutated ant colony optimization
PMACO	Pheromone Mutation based Ant Colony Optimization
TSP	Travelling Salesman Problem
TSRP	Tabu Search based Routing Protocol
TSRA	Tabu Search based Routing Algorithm
VAST	Volume adaptive searching technique
RMRPTS	Reliable Multi-level Routing Protocol with Tabu Search
UOTabu	Uni-Objective Tabu Search
MOTabu	Multi-Objective Tabu Search

CA-ACOA	Cellular Automata based Ant Colony Optimization Algorithm
CA-PSO	Cellular Automata based Particle Swarm Optimization
CA-GA	Cellular Automata based Genetic Algorithm
SOA-ABCA	Stochastic Optimization Algorithm-Artificial Bee Colony Algorithm
SOA-PSO	Stochastic Optimization Algorithm- Particle Swarm Optimization
SOA-ACO	Stochastic Optimization Algorithm- Ant Colony Optimization
PV	Probabilistic Value
CS	Candidate Solution
OCS	Old Candidate Solution
PD	Population Diversity

CHAPTER 1

INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) is the emerging network of Intelligent Transportation System (ITS), conceived to enhance comfort and safety of road travel. The services of VANETs are dependent on the security requirements like availability, authentication, privacy protection, non-repudiation and integrity. The security features are vulnerable to many forms of attacks and are recognizable only when the network has all resources available at all times to all users. Availability turns to be the most important requirement to be focused on serving the users on the road [1].

1.1 Preamble

The services in VANETs are threatened by black-hole, grey-hole, DoS and DDoS [2] attacks. The DDoS is a serious form of attack which deprecates the resources of a victim in different forms. The attack can be performed by injecting fake messages, dropping packets and jamming the channel by the combined malicious activity of many attackers. The reliable data dissemination in packet forwarding relies on the co-operation among the vehicular nodes. Higher degree of co-operation facilitates good performance in packet forwarding.

The performance of packet routing is highly influenced when the vehicular nodes are compromised by DDoS attackers. The extent of co-operation is affected by the malicious activity of the attackers who prevent the packet from reaching the destination, leading to the neighbor unreachable problem. The case is severe under traffic management system and collision warning system.

To tackle the problem of neighbor unreacheability the proposed research selects the reliable neighbor to handle DDoS attack to deliver the packets on time to the intended destination. To select the reliable optimal node in a large network like VANETs is an

NP-hard problem which can be handled by meta-heuristics stochastic optimization techniques. The significance of meta-heuristic stochastic optimization for optimizing global search space can be applied along with the benefits of Cellular Automata (CA) to find the best optimal (reliable) vehicle to forward packet.

The Cellular Automata is a simple, abstract discrete lattice used as a model of complexity derived by using the behaviors and relationships of the small discrete components called the cell. The relevance of any system can be verified using CA. The CA operation is based on the neighborhood of the lattice, where the applicability of CA in VANETs is found. Cellular Automata based stochastic optimization algorithms can be formulated to mitigate the DDoS attack in VANETs in order to increase the availability of the network.

1.1.1 Vehicular ad hoc Networks

The network formed with smart vehicles as moving nodes is named as VANETs in which the vehicles take intelligent decisions to drive safely and smoothly on the highway. The application services supported in VANETs drive the attention of many researchers towards gaining more knowledge and improving its functionalities. The VANETs also gains funding from Government and private agencies to carry out many research projects to enhance the security and comfort of users in a vehicle.

The architecture of VANETs includes major components as smart vehicles, Infrastructure, communication channels, and backbone network to enable communication inside and outside the vehicular network respectively as shown in Figure 1.1. The vehicles on the road communicate using DSRC (Dedicated Short Range Communications) and WAVE (Wireless Access in Vehicular Environment) standard. DSRC/WAVE [8] is the only wireless technology that has potential to fulfill the requirements of VANETs services. Robustness, scalability and short latency are the key requirements supported by these standards. The packet format, channel allocation, and other rules to be followed for vehicular communication are regularized by DSRC/WAVE standard.

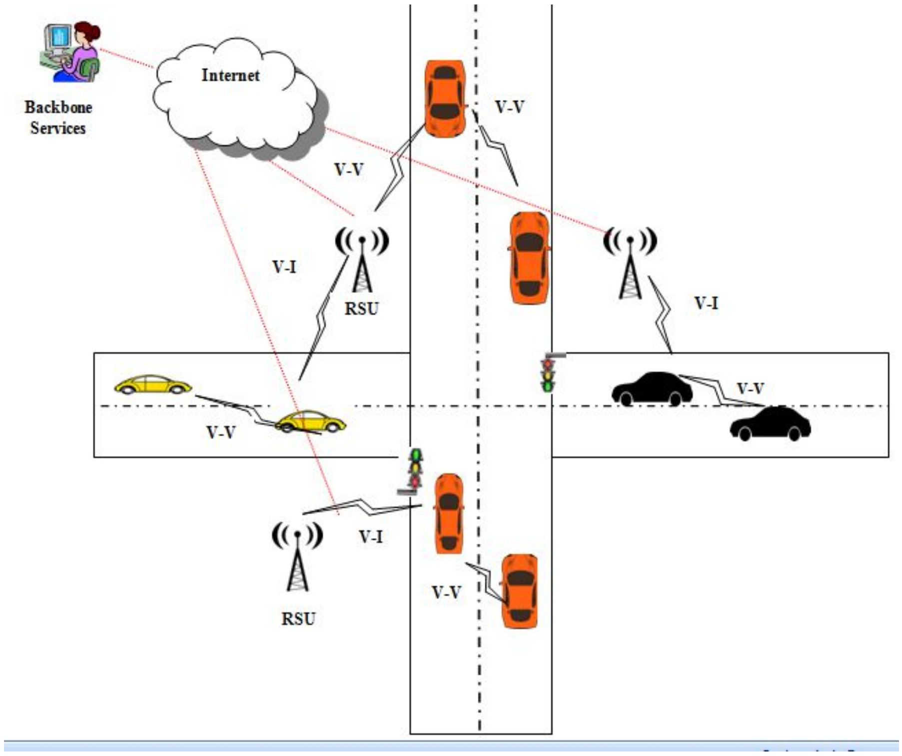


Figure 1.1: Architecture on VANETs

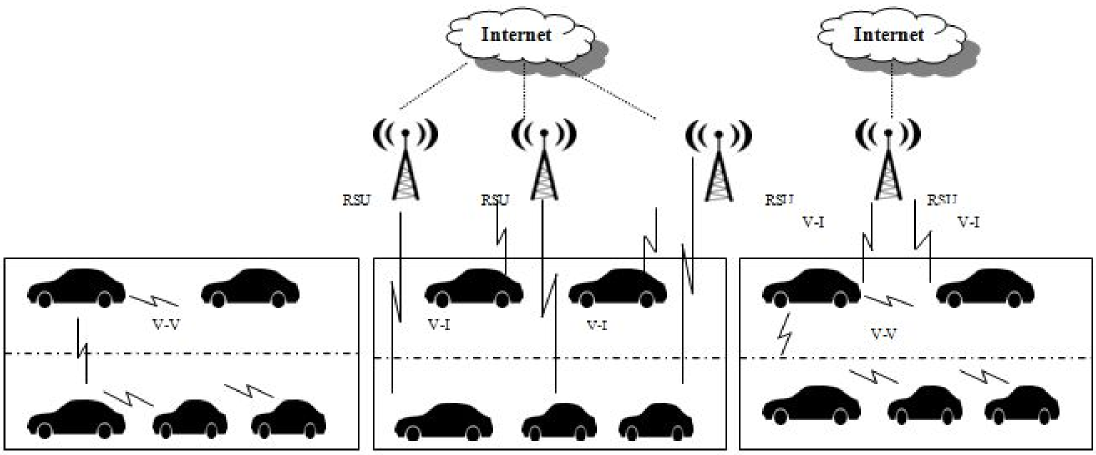
The frequency of communication in VANETs is regulated in US by FCC (Federal Communication Commission) and the band 5.850 to 5.925 GHz called DSRC band is allocated. The use of DSRC band is not subject to license but there are limitations in its use. The available 75 MHz band is divided into 7 channels of 10 MHz wide numbered respectively as 178,172,174,176,180,182,184. The usage characteristics of the channels differ as shown in the Figure 1.2.

HALL Ch 172	SCH Ch 174	SCH Ch 176	CCH Ch 178	SCH Ch 180	SCH Ch 182	Hi-Power Public Safety Ch 184

Figure 1.2: DSRC Channel

The WAVE system uses two hardware components to allow DSRC Communication: On-Board Unit (OBU) and Road Side Unit (RSU). These units obey the rules of WAVE standard to have a low latency communication in order to satisfy the time requirement of VANETs messages. The communication modes enabled by the devices are shown in Figure 1.3.

- Vehicle to Vehicle (V-V) where an OBU transmit messages to other OBU.
- Vehicle to Infrastructure (V-I) making an OBU communicate with the fixed infrastructure like GSM, UMTS or Wi-Max to make the vehicle access backbone network for accessing Internet.
- Hybrid (V-V and V-I) to extend the vehicles coverage area of communication.



a. V-V communication b. V-I communication c. Hybrid communication

Figure 1.3: Communication modes

Characteristics of VANETs:

- High mobility: Vehicles move with different speeds and directions all the time on the road and make the position prediction very difficult. Compared to MANET, VANETs mobility is relatively high.

- Dynamic topology: Due to high mobility VANETs topology changes frequently. Thus it is dynamic and unpredictable. The connections between the vehicles are short and routing paths also change rapidly and security is also difficult to maintain.

- Frequent disconnection: fast movement of vehicles take them go away from the neighbor's radio range more often leading to interrupted connectivity.
- Limited bandwidth: The DSRC standard provides limited bandwidth that is divided into 7 channels. Only 4 channels are used for data packets. The others are used for control and emergency applications.
- Attenuations: In open air environment many noise signals can attenuate making messages useless.
- Limited transmission power: The WAVE standard limits the distance, a data can be transmitted, to 1000m.
- Energy storage and computing: VANETs does not have the limit on storage and computing but processing the real-time information is a big challenge.
- Hard delay constraints: The services in VANETs should have fast response time compared to any other QoS parameter. Delay should be very small for the vehicles in order to take quicker action.
- Geographical type of communication: Compared to any other network VANETs data packet carries geographical information like location area of the source, destination and next neighbor to find the route for packet delivery.
- Mobility modeling and predication: Even though VANETs is dynamic, frequent disconnected network, the mobility pattern can be predicted due to the static road layout. Using the mobility pattern, the future position of each vehicle can be predicted using which many applications are designed.
- Various communications environments: The environment where VANETs operate is of two kinds: Highway with simple straight road where no obstacles like building or trees and city with more complex structure having many obstacles. The city can also be classified as rural and urban.

Security in VANETs

A network of intelligent vehicles can promote safer and faster travel on the road but there is lack in the security of the network. Since each vehicle communicates through messages in a strange environment they rely on many security requirements like availability, authenticity and integrity. Due to unique characteristics of VANETs these security requirements are threatened by many attacks to create a security breach in VANETs as shown in Figure 1.4.

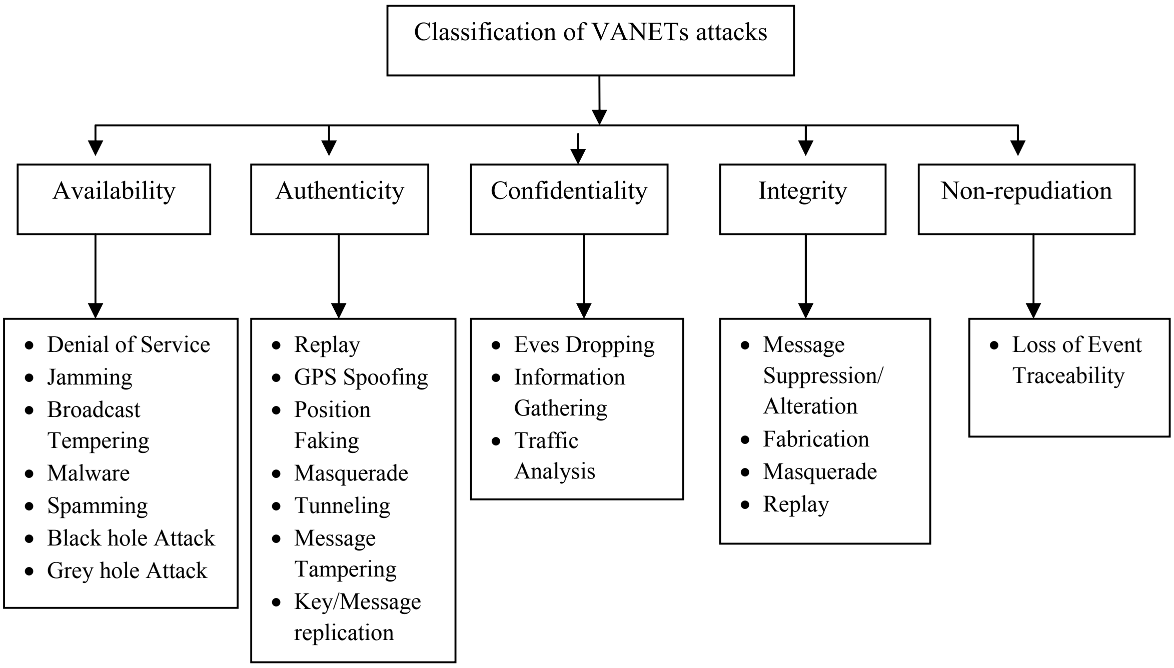


Figure 1.4: VANETs Security threats and attacks

Availability is the top level requirement to be ensured for supporting the functionality of the rest of the security features [5]. The entire network is threatened by attacking the availability of resources in VANETs. Thus the need arises to mitigate the causes that affect the performance of the network and prevent the network from functioning properly. The availability of VANETs services is threatened by various forms of attacks as in [7] given below:

- *Black hole attack:* The Black hole attack is a typical attack on availability of all the ad-hoc network types including VANETs. A Black hole node receives the packet and declines to take part in forwarding data to its neighbor but always declares as part of the network and able to participate [6].

- *Grey hole attack:* Variant of black hole attack is the grey hole attack which varies by dropping only the data packets leading to packet loss.

- *Spamming attack:* To consume the bandwidth and promote collisions the attacker can induce spam messages inside the network which are difficult to control once spread.

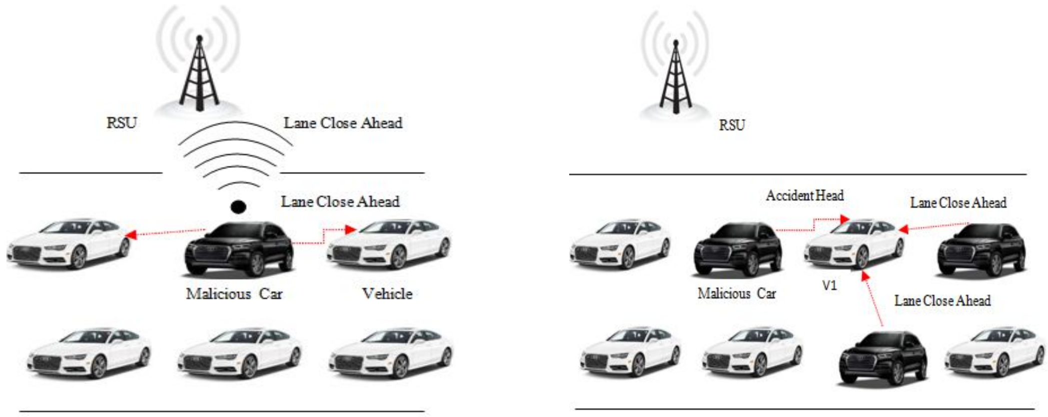
- *Malware attack:* Due to the update of software components of vehicles (OBU) and fixed infrastructure (RSU), the malware (malicious software) trespasses the network to create severe consequences by interrupting the network functionality.

- *Broadcast Tampering attack:* By injecting fake security alert messages, the security of network is affected.

- *Denial of Service attack (DoS):* The Denial of Service attack is the major threat to the availability of the network creating serious consequences. It includes family of attack as follows:

    i. *Jamming attack:* It is a physical layer form of causing DoS attack by jamming the channel with signals and lowering the Signal to Noise Ratio (SNR) of the receiver node.

    ii. *Distributed Denial of Service (DDoS):* The DDoS is a mutant of DoS attack where more than one malicious node is participating in draining the resources of the network. The malicious node achieves DDoS by flooding the network. The Figure 1.5 represents the DoS and DDoS presence in a VANETs environment.



(a) (b)

Figure 1.5: a) Denial of Service attack b) Distributed DoS attack

The threats to availability have to be focused and mitigated to maintain the performance of the network up to the mark.

1.1.2 Impact of DDoS attacks in VANETs

The Distributed Denial of Services (DDoS) is a dangerous attack which takes place in a distributed manner. The attack is performed by more than one attacker from different locations with different time slots [3]. Since multiple attackers perform attack randomly they are very hard to be detected and handled. It a serious form of threat to availability in which the users are denied access to the needed resources of the network.

The DDoS occurs due to following two scenarios:

(i) Malicious behavior of the node which intentionally attacks the victim vehicle to deny the services and make VANETs resources or services unavailable,

(ii) Dense traffic in VANETs makes a vehicle to unintentionally deny the services to neighboring vehicles. In both the cases, the legitimate user is affected from accessing the services of the network.

The network performance is affected due to the following form of DDoS attack [16]:

Computational DDoS attack, the attacker floods the network with numerous messages to drain the computational power of the vehicles. As more attackers perform the attack more processing is done on vehicle preventing it from accessing other needed services.

Memory based DDoS attack, more messages makes the memory of the vehicles to get filled up soon thus intended legitimate messages are dropped.

Signature DDoS attack, any authentication scheme requires verifying the signature before proceeding to process the message. In the case of fake signature, the vehicle keeps on verifying the fake signature.

Neighbor DDoS attack, the most important form of DDoS where neighbors do not cooperate to forward data because of its malicious behavior or dense VANETs. The cooperation among the vehicular nodes is essential to route data packet to the destination. The decrease in the degree of cooperation in vehicles creates a threatening environment where no data is forwarded resulting in a serious impact especially in the applications like traffic monitoring system, collision warning system, etc.

The impact of DDoS in VANETs routing renders the safety of the user at high risk [15, 16]. The safety application includes various services that are to be available to all users at all time in order to provide users with a comfortable and safe drive on the road. The message if delivered out of time, dropped or modified, promotes greater impact on user's lives on the road and traffic on the road becomes a bottleneck. Threats to other security requirements are tolerable compared to availability. Even though all services exist, but are unreachable to users, they are of no use in promoting these requirements and lead to the loss of lives, time, and cost [21].

1.1.3 Co-operation in VANETs

The VANETs has no infrastructure, no centralized control but has dynamic topology and high-speed mobility. The communication solely depends on the V-V communication to transmit the messages. Co-operation between the vehicles plays a key role in facilitating the functioning of the VANETs services [10]. The reliable data dissemination in routing packets necessitates the co-operation as the potential factor from the vehicular node's perspective. The action of packet forwarding tendency is achievable only when co-operation among active nodes is facilitated.

The vehicles in the network can be classified as co-operative and non-cooperative based on the behavior reflected by them. The packet forwarding in the network necessitates the presence of reliable and reachable neighbor node. The reliability and reachability are calculated based on the trust and reference velocity. The neighbor unreachability and unreliability lead to the misbehavior of the node leading to the message loss.

To forward a message the vehicle has to select the neighbor node having good co-operative behavior [11] based on the previous velocity, position, reference velocity, reliability measures. On transmitting if there is no such co-operative node present, the availability of neighbor is affected leading to the DoS. The co-operative node becomes unavailable only in two scenarios: if the neighbor is a malicious node (DoS attack) or heavy traffic, the neighbor unknowingly drops the packet.

The performance of packet forwarding is highly influenced when the vehicular nodes are compromised by DDoS attackers. The malicious behavior of attackers greatly impacts the extent of co-operation maintained among the vehicular nodes [13]. Lack of co-operation leads to catastrophic impacts especially in critical application traffic monitoring systems.

For routing packets in VANETs, high degree of neighbor co-operation is needed to provide an uninterrupted functioning of the network.

1.1.4 Role of Cellular Automata for handling DDoS attack

A Cellular Automata (CA) is discrete, abstract computational system used as model of complexity and as more specific representation of non-linear dynamics. They are regarded as fully distributed systems of computation of complex systems with the local processing of simple components.

The history of Cellular Automata dates back to 1940s and 1950s when John von Neumann, called the father of Cellular Automata, worked to generate a self-replicating machine [20]. In 1970, after the discovery of Conway's Game of Life, a two-dimensional CA, interest in exploring the capabilities of CA increased. Stephen Wolfram, a physicist performed a detailed study of one-dimensional or elementary CA. He published "A New Kind of Science" in 2002 giving the advanced and mathematical proof of CA that can be used in different fields of science from designing hardware of a computer to cryptography.

a. Components

CA is an infinite, regular lattice of simple finite state machines that change their states synchronously, according to a local update rule. The rule specifies the new state of each cell based on the old states of its neighbors. CA can simulate complex systems by the interaction of cells following easy rules [21].

Basic Idea: A cellular automaton is represented by the 4-tuple (*Z, S, N, f)* where:

Z is the finite or infinite lattice or grid of cells.
S is a finite set of cell states or values.
N is the finite number of neighborhood.
f is the local transition function defined by the transition table or the rule.

Grid

A cellular automaton designs any complex system by integrating the relationship and behaviour of simple components called cells. These collections of cells form a grid or lattice which comes in different varieties from square as shown in Figure 1.6, triangular to hexagonal grids.

C1	C2	C3	C4	C5	C6	C7	C8

Figure 1.6: One-dimensional CA

Cell

A Cell is a basic element of CA each with its unique state updated iteratively by following some transition rule. The local update of each cell ends with the self- organization of the entire structure of the grid.

State

The state of a cell represents the value of a cell at time 't'. The number of states of a cell can be from two (binary 1 and binary 0) to any number based on the application and the state is tied with the rule.

Neighborhood

The next state of the cell depends on its neighbor's present and previous states and also its own present and previous states. The number of neighbor to be selected is based on the radius of the grid we set to get updated. CA comes in one-dimensional, two-dimensional and three-dimensional neighborhoods with different radii. To have more neighbors the radius can be increased with the increase in the dimension.

Rules -State Transition Function

The state transition function or local update rule determines the state of each cell in next iteration based on some pattern of the states specified earlier. A set of rules is framed for each pattern of data. The determination of next state of a cell is given in Figure 1.7.

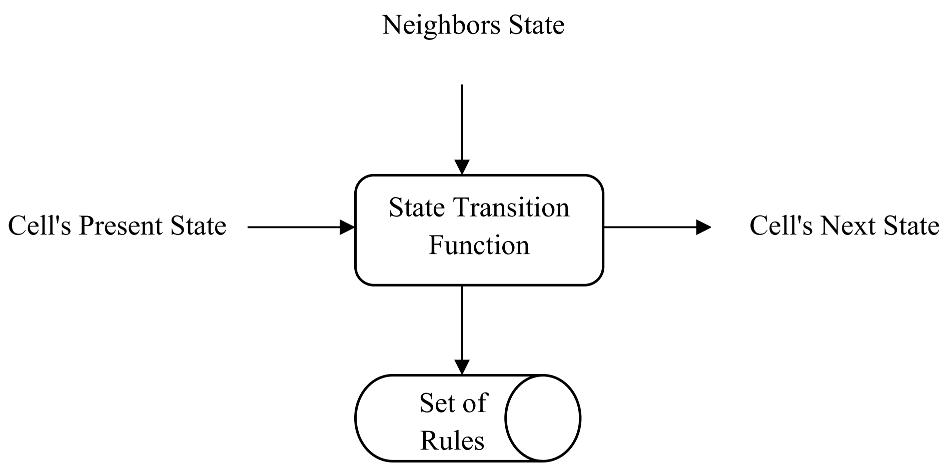


Figure 1.7: Transition Function

Features of Cellular Automata include the following,

- They are simple and easy to be implemented.
- They are able to verify the relevance of physical mechanisms.
- They can include relationships and behaviors which are difficult to be formulated as continuum equations.
- They reflect the intrinsic individuality of cells.
- Discrete dynamical system simulator.
- Simulation results are much more intuitive as they are visually well represented.

Cellular Automata in VANETs

The method of modeling a complex system to find the solution makes Cellular Automata suitable for modeling the VANETs system for identifying the vehicular nodes for routing packet. The use of features like neighborhood and state in CA [20], makes it applicable for identifying the availability of vehicular nodes for ensuring connectivity under DDoS attack. CA in VANETs transforms the information pertaining to the designed cells, cellular spaces and cellular neighbors of each traffic to the vehicles, road, and neighbors respectively.

The vehicle in a cell can be updated by the rules framed by cellular automata in selecting the best neighbor for forwarding the packet. Based on the framed rules, current information pertaining to the node availability is made to be known to the neighboring nodes. The quantification for the availability of each cell not only depends on the current state of the cell but also on the state information of the neighboring nodes.

The state transition rules in cellular automata can be modified by redefining the rules to suit the application needs [21]. For handling DDoS attack, the rules can be modified to find the best neighbor with the highest degree of cooperation. For better connectivity, cellular automata find the best vehicle to route the packet by avoiding the DDoS compromised vehicles.

From the local transition, each cell gets self-organized and picks the best node leading to the global organization of the vehicles. All vehicles work in a distributed, cooperative way by following the rules framed. Thus the best result of cellular automata comes from the quality of the transition rule [23] which can be improvised to achieve the best solution.

1.1.5 Categories of Cellular Automata Model

a. *One-dimensional Neighborhood(1-D)*

Since in 1D there are no shapes, the definition of the neighborhood is usually very simple as shown in Figure 1.8. It is also called Radial neighborhood since the neighborhood in 1D is described by its *radius*, *r*, meaning the number of cells left and right from the central cell that are used for the neighborhood [24]. The output cell is positioned at the center.

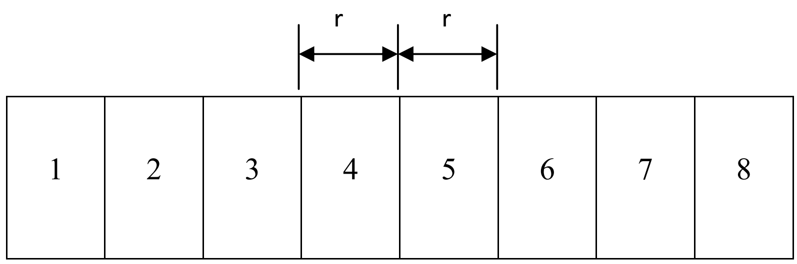
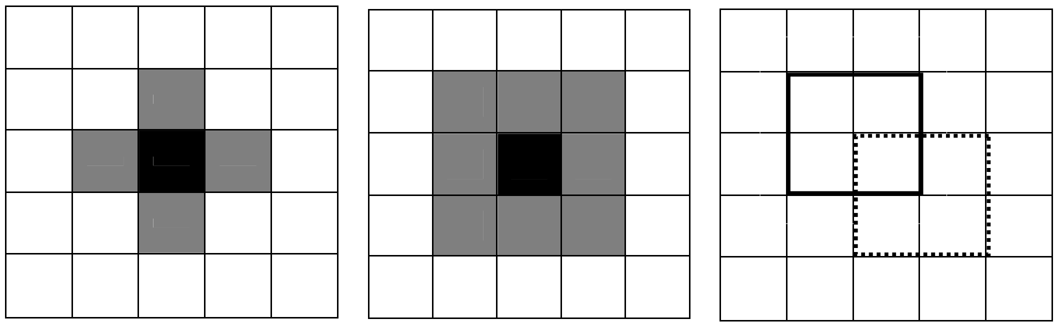


Figure 1.8: 1-D Neighborhood Model

b.*Two-dimensional Neighborhood(2D)*

The 2D CA exhibits the same characteristics of 1D CA with increased number of neighbors and generates good output since all or more surrounding neighbor's states are used. The models of 2D cellular automata are formed by complicated neighborhood with cell space arranged in triangle, square or hexagonal grid.

The commonly used models of CA, Von Neumann model, Moore model and Margolus model as shown in Figure 1.9, follow square model [22]. From the Figure 1.9, the black cell at the center and the grey cell are its neighbors. The Margolus model differs from the other two by the way the cells are handled where a cell block with size 2 x 2 is dealt together and each cell handled separately for rule update respectively.



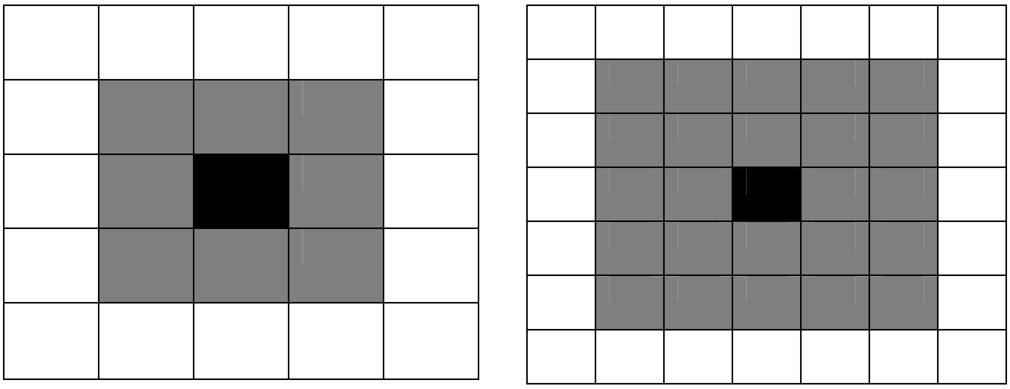
a) Von Neumann Model b) Moore Model c) Margolus Model

Figure 1.9: 2-D Neighborhood Model

Moore Model

In Cellular Automata, the Moore neighborhood is formed by a central cell and eight cells surrounding it making a square lattice. The center cell focuses on all direction cells for deciding the next move of the dynamic system. The neighborhood is named after Edward F. Moore, explorer of cellular automata theory.

The dimension of Moore model can be extended based on the range r as $(2r+1)^2$. The Moore model with its feature of focusing all 8 cells surrounding it makes it suitable for optimal route selection. Since from all directions, the focus is set, it does not fall into local optima and gives a better solution when compared with other models. The Moore model size can be extended by increasing the radius as shown in Figure 1.10.



a) Range, r =1 b) Range, r=2

Figure 1.10: Extended Moore Model

Moore Model is structured as a square grid (usually 3×3 cells) with the output cell in the center. The neighbor cells are described by the directions on the grid as indicated N=*{NW,N,NE,W,C,E,SW,S,SE}*.

Formal definition

Formally the Moore neighborhood is the set of neighbors of coordinates.

N={{-1,-1},{0,-1},{1,-1},{-1,0},{0,0},{+1,0},{-1,+1},{0,+1},{1,+1}}

1.2 Need for the Research

VANET is an emerging technology in which ITS (Intelligent Transportation System) has found its advancements in the recent years due to the tremendous growth in hardware and software utilization in ad-hoc networks. The vulnerabilities in VANETs make it prone to a number of security breaches that degrade its purpose and performance. Further, DDoS attack in VANET is a serious threat to the network as it exploits the resources of the network to a maximum level. In spite of numerous research works carried out in VANETs security, there is a huge research gap existing in the mitigation of DDoS attack in VANETs. Thus there is a need for the research to facilitate maximum degree of security in VANET by ensuring complete availability of all services whenever required. This way of ensuring availability of services opens a new way to enhance the availability of each vehicle to their corresponding neighborhood vehicle and confirms reliable forwarding of the data packets. The need concentrates on the applicability of meta-heuristics in VANETs for estimating the best feasible vehicle in forwarding data that avoids maximum impact from DDoS compromised node. It also explores the need for utilizing cellular automata as the key entity for incorporating meta-heuristics techniques that are proved to be optimal in finding global optimal solution under increased performance. Besides the comprehensive set of solutions that could handle the impacts of DDoS attacks, the role, of meta-heuristic algorithms like tabu search, ant colony algorithm, and artificial bee colony algorithms in mitigating DDoS attacks, needs to be investigated. Hence, this research needs to focus on the design of an efficient and effective meta-heuristic optimization based DDoS mitigation mechanism for facilitating maximum co-operation and network lifetime.

1.3 Motivation of the research

In VANET, reliable routing of packets necessitates maximum co-operation as the essential factor from the vehicular node's perspective. The action of packet forwarding potential is always met only when the co-operation among active mobile nodes is facilitated. The performance of packet forwarding is highly influenced when the vehicular nodes are compromised by DDoS attackers. This malicious intent of attackers greatly impacts the extent of co-operation maintained between the vehicular nodes leading to catastrophic impacts especially in critical applications like traffic monitoring systems. A number of significant approaches were contributed for handling DDoS compromised nodes using heuristic approaches but meta-heuristic stochastic optimization algorithms are identified to be highly suitable for most of the dynamic applications for enhancing feasible solutions that could exist in a specific domain. The rules of defining cellular automata also play a significant role to determine the trustworthy neighborhood nodes that aid in reliable dissemination of data in the network. The meta-heuristic stochastic optimization techniques like tabu search, ant colony algorithm, artificial bee colony algorithms provide an effective way for reliable mitigation of DDoS compromised nodes as they embed maximum degree of exploitation and exploration which leads to improved accuracy in detection. Hence, mitigation mechanisms based on tabu search, ant colony algorithm, and artificial bee colony algorithms that appropriately determine DDoS attacks through maximum degree of exploration and exploitation incorporated in the transition rules of cellular automata need to be formulated.

1.4 Scope of the research

The major potential of VANET in rendering support to the users for driving conveniently on the road, is made possible only through reliable communication among the vehicles. The dynamic nature of VANET makes the vehicle to vehicle communication more effective in terms of multi-hop communication. But the open air medium usage in VANET makes it vulnerable to many serious threats leading to performance degradation in message delivery. There are numerous attacks that can be performed in different layers to pull down the efficiency of VANETs. Among all other security features, availability requirement is the more predominant and has to be present consistently for the conventional working of VANETs services. The research work focuses on reviewing the attacks on availability in network layer and work against the DDoS attack, the most dangerous form of attack, to enhance the availability. The scope of the research is handling DDoS attack in network layer to route packets to the destination using vehicle to vehicle communication with no interference from infrastructure unit. The DDoS attack on neighbor cooperation and reach ability is focused by selecting the best cooperative vehicle for forwarding the data packets. The neighbor reliability, reach ability and trust information are applied to select the neighbor node. The neighborhood is formed by using Moore model of Cellular Automata which applies rule to update the neighbors. Stochastic Meta-heuristics optimization algorithms are applied to frame the rules for Cellular Automata model to find the neighbor. The stochastic algorithms like Ant Colony optimization (ACO), Artificial Bee Colony (ABC) algorithm and Tabu Search algorithms are enhanced to find the optimal neighbor. Thus meta-heuristic techniques inspired mitigation approaches are found to offer better scope for reliable identification of DDoS attacks in VANET.

1.5 Problem Statement and Objectives

1.5.1 Problem Statement

The core aim of this research work is to design and implement four kinds of stochastic optimization techniques through cellular automata for mitigating DDoS attacks in VANETs in order to facilitate maximum degree of co-operation among the vehicular nodes. This aim defines and facilitates the scope in deriving the benefits of stochastic optimization for optimizing global search space and enabling significant degree of exploitation and exploration of the search space. This aim also provides the option of investigating the proposed stochastic optimization-based mitigation schemes using the proven multi-model benchmark functions for understanding the context of their superior performance.

1.5.2 Objectives

The core objectives of this research are:

- To formulate and deploy an enhanced version of ant colony optimization (ACO) algorithm called Cellular Automata-based Improved Ant Colony Optimization Algorithm (MACOA-CA) for updating rules in order to enable dynamic adaptive adjustment of pheromones to mitigate DDoS compromised vehicular nodes.
- To design and implement a Cellular Automata-based Modified Artificial Bee Colony Algorithm (MABCA) using an enhanced version of ABCA that uses two Differential Evolution (DE) based search strategies in the onlooker bee phase and an integrated Chaotic and Opposition-Based Learning in the scout bee phase for optimally selecting a vehicular node for replacing DDoS compromised vehicular node.
- To propose and incorporate an Improved Stochastic Optimization algorithm based Cellular Automata that integrates Grenade explosion technique and Cauchy Operator with Artificial Bee Colony Algorithm (CA-IABCA) for mitigating DDoS attacks using the benefits of grenade explosion in the onlooker bee phase and Cauchy operator in the scout bee's phase.

- To contribute an Improved Tabu Search based Cellular Automaton (CA-MTSA) Inspired Algorithm for quantifying the availability of each vehicular node attributed towards effective data forwarding based on its past velocity, past reliability factor, local optimal state, global optimal state and neighbor's best state for mitigation DDoS compromised nodes.
- To measure the performance of the proposed stochastic optimization algorithm by the evaluation parameters like prediction variance, average prediction variance, Packet Delivery Ratio and End to End delay by varying number of nodes and prediction interval and finally to investigate the superiority of the proposed algorithms by employing multi-modal bench mark functions.

1.6 Methodology and Approach

The problem of finding the reliable neighbor in a large, dynamic, instable topology network for routing the data is similar to finding the optimized path in a large network. Thus the selection of optimal neighbor is considered as the optimization problem which can be solved by applying various approximate solutions. The widely used approximate techniques are Meta heuristics and stochastic optimization algorithms [26, 27]. Among the available algorithms, ACO, ABC, and TS are found to be efficient in finding the global best solution [28] based on some enhancement made in standard algorithms to suit the applications.

The approach for finding the best neighbor using the above-stated algorithms requires the following information of the vehicle. The vehicle’s reference velocity, reliability and trust factor are measured by applying any one of the Meta heuristics algorithms. The fitness values and their probability of being selected are calculated and finally, the vehicle with higher probability is selected as the next forwarding node.

The neighborhood plays a key role in VANETs to route packets. Thus in this research work, the cellular automata Moore model is applied. The benefits of Moore model allow the vehicles to select limited neighbors of closer reference velocity. The cellular automata combined with Meta heuristics enhance the efficiency of both the techniques. The transition rule of CA is optimized by proposed stochastic algorithms and these standard algorithms are improved in terms of parallel execution.

The proposed algorithms are evaluated by simulation and analytical methods. The simulation is evaluated by measuring the prediction variance, average prediction variance, packet delivery ration, and end-to-end delay metrics and compared with the existing algorithms. The analytical testing was done by testing the proposed algorithms with standard benchmark functions to test the rate of convergence, precision, robustness and general performance in increased dimensions.

1.7 Organization of the Thesis

The outline of this thesis is organized as follows:

Chapter 1 introduces the background of VANETs, and then illustrates the impact of DDoS with brief introduction about the Cellular Automata, motivations, problem statements, research objectives and main contributions of this thesis.

Chapter 2 presents the literature survey. In this chapter, we review all attacks on availability of VANETs and then give a detailed view of DDoS attack and mitigation works carried out by other researchers. This chapter also gives the literature survey of different versions of Ant Colony Optimization, Artificial Bee Colony algorithm and Tabu search algorithms to solve the optimization problems.

Chapter 3 describes the methodology of the proposed Modified Ant Colony Optimization Algorithm inspired Cellular Automata (MACOA-CA). The proposed algorithm uses dynamic pheromone updating and evaporation strategies for mitigating DDoS attack in VANETs by using the global optimal solution found by IACOA. The chapter gives the detailed algorithm steps and flowchart with the simulation and experimental analysis.

Chapter 4 explains the technique of the proposed Cellular Automata based Improved ABC algorithm (CA-IABCA) by applying Grenade Explosion and Cauchy operator. The algorithm increases stochastic nature to avoid DDoS attack in VANETs for finding global optimal solution with the elaborate algorithm steps and flowchart with the simulation and experimental analysis. This chapter also describes the methodology of the proposed Cellular Automata based Modified ABC algorithm(CA-MABCA) for handling DDoS attack in VANETs by involving Differential Evolution and chaotic system to find the global optimal solution. The proposed algorithm is explained using the flow chart and algorithm with the simulation and experimental analysis.

Chapter 5 explains the design of proposed scheme of Cellular Automata Inspired Modified Tabu Search Algorithm to select the best neighbor based on reliable factor. The vehicle movement is traced in Cellular Automata Moore model for selecting the best one. The proposed algorithm is explained using the flow chart and algorithm with the simulation and experimental analysis.

In Chapter 6 the proposed algorithms are compared with benchmark multimodal functions to explore its efficiency.

Finally, the Chapter 7 concludes the dissertation and offers some directions for future works.

1.8 Summary

In this chapter brief introduction about the VANETs along with its standard, communication modes, components, architecture, routing protocols and security issues are discussed. Next section focus on explaining the security breaches at the network layer and categorized different types of attacks practiced. The DDoS attack, a severe form of attack, was then briefed with its impact in VANETs services. The applicability of cellular automata in handling DDoS is also explored along with different categories of CA models. In the next succeeding sections, need and scope of the research work along with motivation has been stated. Finally, the methodology and approaches applied for handling the problem was discussed.

Chapter 2

Literature Survey

The Distributed Denial of Service (DDoS) is a major form of attack in VANETs caused by injecting fake messages to prevent the legitimate users from getting the VANETs services. The cooperation among the neighboring nodes is necessary for forwarding data packets to reach the destination on time. The selection of the reliable neighbor vehicle for data forwarding in large, dynamic and high threat prone VANETs is a NP-hard problem. Meta-heuristics optimization techniques are suitable for handling such NP-hard problems to come up with a feasible solution. The Cellular Automata Neighborhood models follow the neighborhood strategy to solve a problem. This chapter gives the survey of ACO, ABC and TS meta-heuristics stochastic optimization techniques applied to find the best feasible solution. The chapter also elaborates the work from the literature regarding the Cellular Automata models for mitigating DDoS attack in VANETs.

2.1 Cellular Automata-based DDoS Mitigation Techniques

Cellular Automata is a discrete dynamical system of grid in which each cell is in predetermined states. The neighborhood is formed for each cell based on the transition rules framed for the application. Each cell updates its cell based on the transition function in each discrete time step in a synchronous manner.

Cellular Automata is applied to model the mobility of vehicles [29] for UV-CAST in VANETs to disseminate messages. The mobility of vehicles is framed by interconnections including traffic lights, cycle duration for traffic dynamics and vehicle interspaces. The vehicles following the motion pattern of the CA model gives congestion free traffic and warning messages are disseminated properly with reduced message delivery latency [30]. Cellular Automata is used to form a car society [31] to cluster the interest users based on Zone of Interest (ZOI) for mobicast communication.

The dynamic urban growth is modeled using cellular automata model in [32] whose transition rules are improved by particle swarm optimization. The similarity of self-organization and bottom up feature of PSO (Particle Swarm Optimization) and CA made the hybridization of these two techniques possible. The PSO parameters and objective functions modeled the urban growth which is driven by CA as transition function. The CA Moore model is best suited for searching the best neighbor since the model focuses all the surrounding neighbor cells to change the state of center cell.

The Tourist navigation system is modeled using cellular automata to avoid congestion in the tour path. The authors proposed an adaptive recommendation mechanism in [33] where each group is modeled as a cell of cellular automata and the transition rules are framed to decrease the wait time and congestion and increase scalability.

In [34] the mobile routing is optimized by applying hexagonal cellular grid to find the shortest path to the destination along and also satisfy with the QoS like delay, bandwidth and energy of the MANET.

A hybrid integrated approach called CANPSO [35] was proposed for improving the complexity in identifying the best neighbor for confident packet delivery. This approach was the first innovation that introduced the concept of combining Cellular Automata idea with meta-heuristic algorithms like PSO, Tabu search, and stochastic optimization techniques.

Cellular Automata based Particle Swarm Optimization (CAPSO) [36] is an optimization algorithm that considers each and every individual node of the entire topology as a typical grid of two-dimensional grid cellular automata. In CAPSO, the update rule of each and every individual cell depends on its current state and the present status of the neighbor estimated based on the status of the optimal cell. The update rules are based on re-estimation techniques that possess the capability of determining the best neighbors of the network topology.

Improved Probabilistic Cellular automata based Particle Swarm Optimization (IPCPSO) [37] was proposed for judging and handling the degree of prematurity and deviation that exists between the possible feasible updates of the vehicular nodes. IPCPSO also employs the concept of group interactive policy that avoids the limitation of local minimum.

Mishra, S et al. [38] proposed a Learning Automata based DDoS detection based on the modifications of the existing Optimized Link State Routing protocol. Authors weaved the concept of learning automata and reliable routing assistance for effective and less overhead induce mechanism for detection. Authors proved that the proposed cellular automata assisted scheme is more predominant than the learning supervising schemes of the literature.

Aghababa, M, et al. [39] proved that cellular automata are mainly designed for specific functions and they do not possess a standard programming paradigm. The update rules of cellular automata are formulated in an ad hoc manner or propounded through genetic algorithms like searching approaches. They identified that the proposed evolution rules formulated through genetic algorithm when integrated with cellular automata performs in an effective and efficient way.

2.2 Stochastic Optimizations for DDoS Mitigation

The optimization problem solving technique with randomness introduced to maximize or minimize the objective function is referred to as Stochastic Optimization. The benefit of stochastic optimization technique is that it enhances the global search ability with fast convergence rate.

In [40] the stochastic optimization algorithms such as Genetic Algorithm, Simulated Annealing and Iterative Improvement algorithms are applied to optimize the flow-jamming attack, a form of DoS attack, in wireless sensor networks. The authors redefined the mathematical model for the flow-jamming attack to optimize the attack.

To maximize the coverage area of the network [41], a new stochastic algorithm such as ABC is adopted to optimize the dynamic deployment of stationary and mobile sensor nodes. The result proved that ABC is superior in deploying the nodes in sensor network when compared to others. To defend against DDoS in server [42] the behavior matrix is constructed to depict the user's behavior and a puzzle controller constructs a covariance matrix from the user behavior matrix. The entropy measure of the covariance matrix is compared with the threshold to detect DDoS.

Another stochastic optimization technique called Particle Swarm Optimization (SOA-PSO) [43] was propounded for resolving the issue of DDoS attacks. It is also prone to get trapped into local minimum similar to that of other stochastic optimization approaches of Ant Colony Optimization and Genetic algorithms. SOA-PSO was proposed to improve the search domain and enhance the exploration speed. Authors through simulation proved that the convergence value of SOA-PSO is effective and quantifiable. SOA-PSO is analyzed with multimodal functions like Rosenbrock and Rastrigin and it was found that it optimizes the solution at a faster rate.

2.3 Categories of Stochastic optimizations for DDoS Mitigation

The optimization problems can be solved by using the approximate methods to find the near feasible solution with insufficient resources, data and knowledge. The approximate methods with the randomness injected allow the method to escape from local optimum and approach a global optimum. The randomization included in heuristics approaches leads to stochastic optimization algorithms among which few are listed below

- Swarm algorithms
    - Artificial Bee Colony Optimization
    - Ant Colony Optimization
    - Particle Swarm Optimization
    - Cuckoo Search
    - Flocking and Schooling in birds and fishes
- Evolutionary algorithms
    - Genetic Algorithms
    - Evolution Strategies

2.4 Significance of Meta-Heuristics over Heuristic

Heuristic is a set of rules applied to solve a problem with the knowledge discovered in earlier phase using a heuristic function not guaranteed to be optimal or perfect but produce satisfactory solution in less time [44]. The process carried out by heuristic approach in constructing the solution for optimization problems is depicted in Figure 2.1.

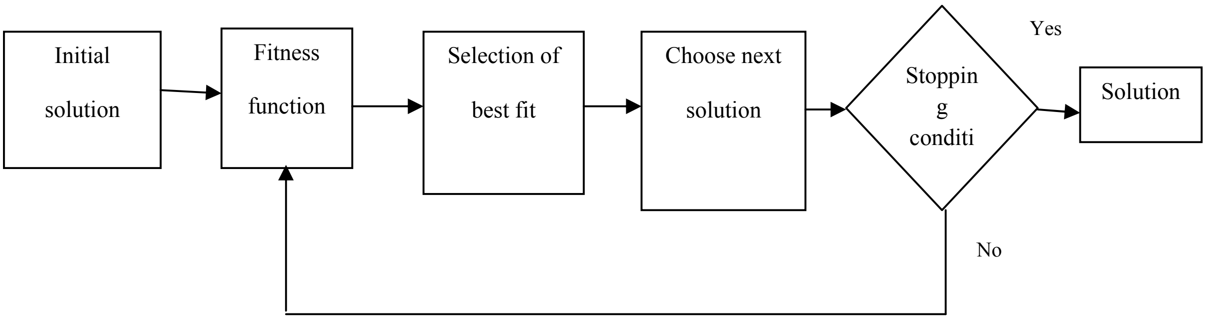


Figure 2.1: Basic Heuristic algorithm workflow

The problem with heuristics approach is

i. *Local optima problem*:

The heuristics technique makes use of greedy approaches to solve any problem. So often they get stuck in local optima problem and fails to produce global optimal solution.

ii. *Specific Heuristics:*

The optimization solution produced by heuristics is specific and problem dependent taking more factors of the problem specified to them giving variations in performance for different applications.

iii. *Deterministic:*

The heuristic function flow is such that the next iteration path can be easily predicted.

iv. *No global solution:*

The heuristic approaches are able to exploit only the local best solution found so far and unable to explore the whole global search space.

Meta-heuristic is an advancement of heuristic approach which is used to guide and modify the operations of heuristic and to improve the quality of the solution produced. It is also called master iterative process which gives good direction for the heuristic algorithm [45] applied in the inner iteration and leads the solution in the right path. Meta-heuristics allows tackling large-size problem instances by delivering satisfactory solution in reasonable time. There is no guarantee to find global optimal solutions or even bounded solutions.

In designing a meta-heuristic, two contradictory criteria must be taken into account: exploration of the search space (diversification) and exploitation of the best solutions found (intensification) [46]. It is an approximate way of problem solving adopting randomness in heuristics to improve the solution for obtaining global best solution [47]. The meta-heuristic differs from heuristics mainly in improving the solution produced by heuristic methods. The random functions are applied to explore new search areas for intensifying the search in the new search space [48].

The flow of any meta-heuristics algorithm follows the steps as shown in Figure 2.2 to improve the solution by searching the entire global search space and produce global best solution for large complex optimization problem [49].

Properties of Meta-heuristics algorithms

- Meta-heuristics are strategies that “guide” the search process.
- The goal is to efficiently explore the search space in order to find near optimal solutions.
- Techniques which constitute meta-heuristic algorithms range from simple local search procedures to complex learning processes.
- Meta-heuristic algorithms are approximate and usually non-deterministic.
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
- The basic concepts of meta-heuristics permit an abstract level description.
- Meta-heuristics are not problem-specific.

- Meta-heuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.
- Today's more advanced meta-heuristics use search experience (embodied in some form of memory) to guide the search.

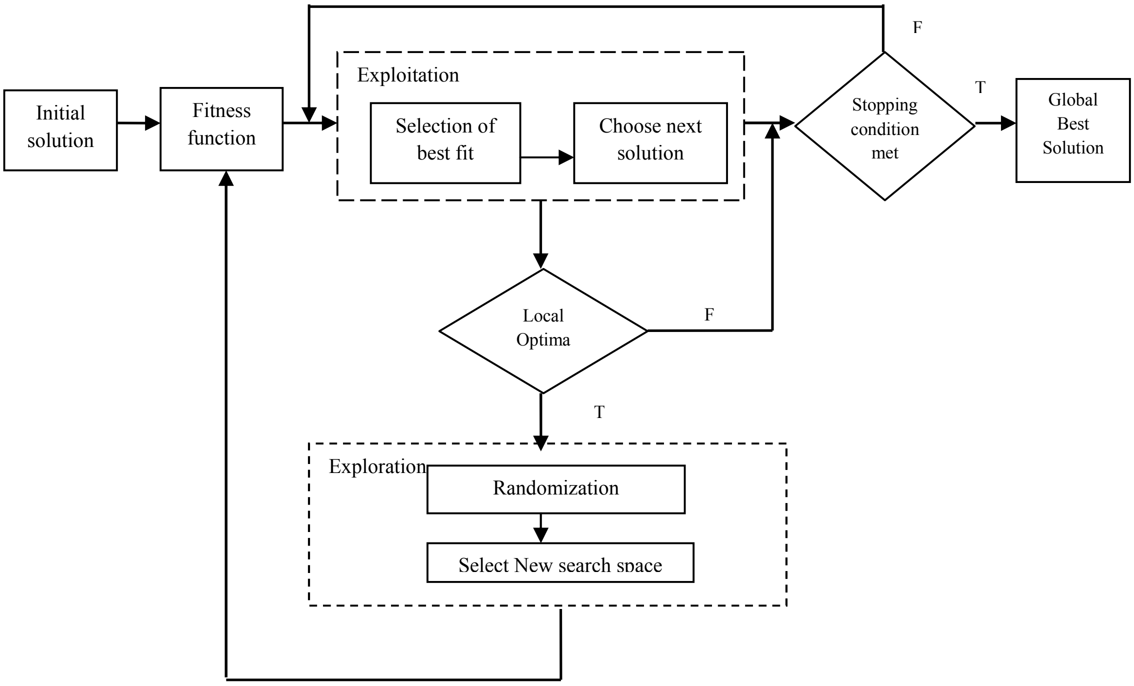


Figure 2.2: Basic Meta-Heuristic algorithm workflow

Meta-heuristics can be applied to solve the following types of problems which cannot be solved using heuristic approaches

1. P class problems with very large instances.
2. P class with hard real time constrains.
3. NP class with moderate or difficult structures of the input instances.
4. Optimization problems with time consuming objective functions or constraints.
5. No analytical models of optimization problems that cannot be solved in an exhaustive manner.
6. Problems with uncertainty and robust optimization.

Advantages of Meta-heuristic optimization

i. *Solve large complex problems*: Meta-heuristics has the capability to solve large complex problems with limited information, resources and constrained time limit.

ii. *Obtain global best solution*: The exploitation and exploration criterion of meta-heuristic makes it possible to find the global solution.

iii. *Applicable for General problem-independent problems*: Meta-heuristics is applied to solve any problem since it does not focus on the features of any specific problem.

iv. *Non deterministic*: Due to randomness in the methodology applied meta-heuristics, is stochastic in nature and solves problems with uncertainty.

2.5 Meta-Heuristic optimization techniques for DDoS Mitigation

As discussed in section 2.4 the optimization problems can be solved by heuristics and meta-heuristics algorithm by searching the search space thoroughly to find the feasible near optimal solution. Several of meta-heuristics based search techniques are discovered by the researchers which are classified as shown in Figure 2.3.

Initially, Chen. H et al. [50] proposed a scheme that addressed the issue of solving a significant kind of DDoS attack called Low Rate DDoS attack using an ant colony framework called DDIACS. Authors used three phases that include information rules framed through heuristics, multi-agent algorithm for DDoS detection and forward search process. Authors proposed DDIACS in a reactive manner such that it could handle and comply with the recently emerging software defined network. DDIACS is capable of effectively monitoring and managing the network traffic and infrastructure. This framework also possesses the benefits of fast convergence and flexibility but fails in dealing with congestion.

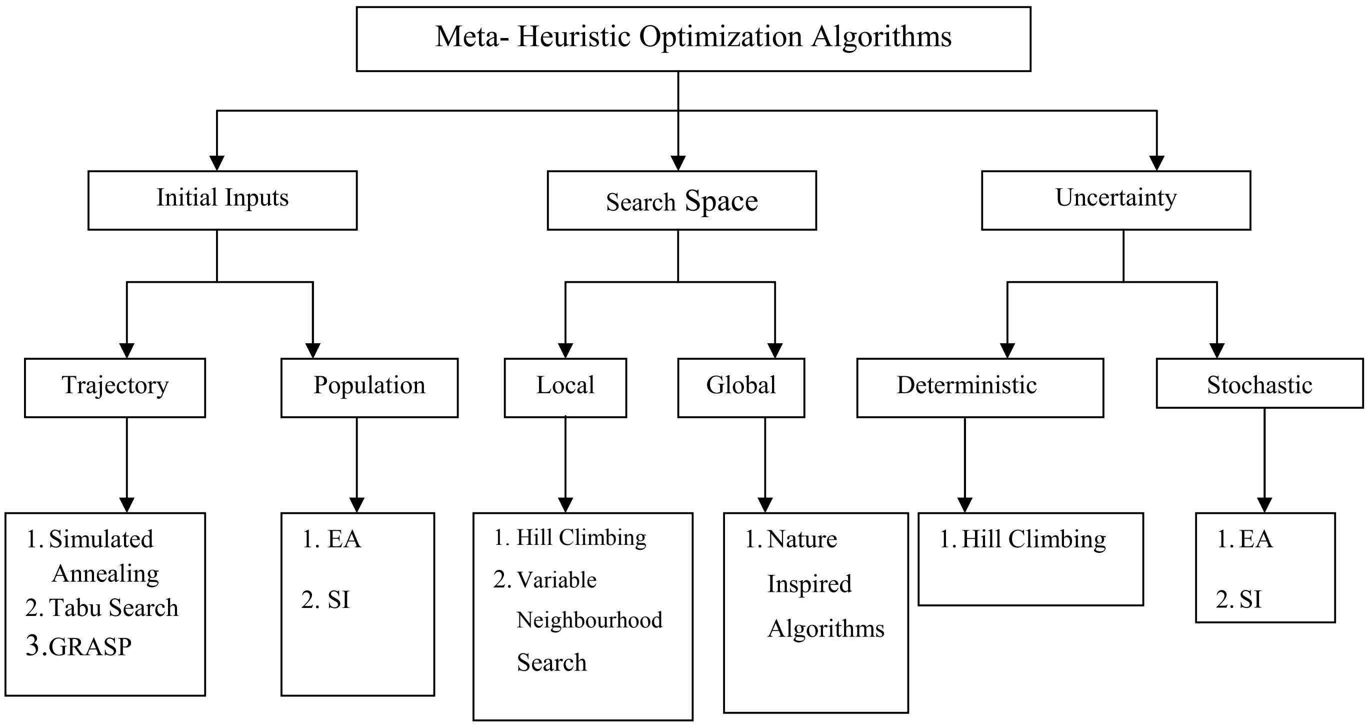


Figure 2.3: Classification of Meta-heuristics algorithms

Vangili, A et al. [51] propounded a bio-inspired approach like Ant Colony Optimization for modifying the operations of AODV routing protocol for detecting malicious behavior in ad hoc networks. This novel scheme incorporates the computation of pheromone value through forward ratio in each and every node.

2.6 Artificial Bee colony inspired Cellular Automata-based DDoS Mitigation

The Artificial Bee Colony optimization algorithm is a population based stochastic algorithm that uses behavior of honey bees for solving optimization problems. In ABC system, artificial bees fly around in a multidimensional search space. Some bees (employed and onlooker bees) choose food sources depending on their own experience and that of their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one. Thus, ABC system combines local search methods carried out by employed and onlooker bees with global search methods managed by onlookers and scouts attempting to balance exploration and exploitation process.

The colony of artificial bees consists of three kinds of bees: employed bees, onlooker bees and scouts. The first half of the colony consists of the artificial employed bees and the second includes the onlookers.

The complete working flow of the standard ABC optimization algorithm is shown in Figure 2.4.

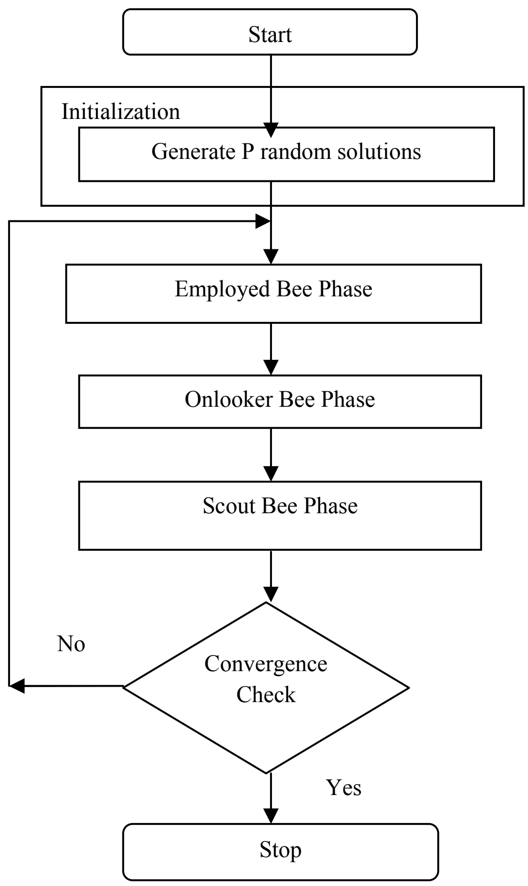


Figure 2.4: Artificial Bee Colony Optimization algorithm

• *Employed bee phase*: each employed bee exploits a single food source and advertises its quality and position to the onlookers, by dancing in the nearby hive.

• *Onlooker bee phase*: the onlookers tend to choose the best food sources to further exploit, based on information communicated by the employed bees through their dances. Therefore, good food sources attract more onlooker bees compared to bad ones.

• *Scout bee phase*: When a food source is considered exhausted, it will be abandoned and its employed bee will be converted to a scout that will randomly chose a new food source to replace the old one.

2.6.1 Improvement of Artificial Bee colony

The authors in [52] proposed an improved ABC algorithm to improve exploration performance by introducing the Rate of Change (RoC) concept using slope to modify the limit in the scout bee phase to improve the exploration. The RoC in the performance graph is tested over benchmark functions and colony size parameter and produced good improvement in the algorithm exploration.

The ABC algorithm performance degrades because they are poor at exploitation and in [53] the authors proposed a technique to handle the problem. The standard ABC is improved by modifying the onlooker bee phase to concentrate the search based on previous best solution. The scout bee phase is improved by computing the distance between the best and worst solution, to avoid generating worst solutions.

The exploitation of the solution from a search area is improved in [54] using Grenade explosion in onlooker bee phase where the searching depth is increased. To balance the exploitation with exploration the authors applied Cauchy operator in scout bee phase. The experiment results proved that performance increased by applying the modifications.

In [55] the exploitation is improved by applying Differential Evolution (DE) in onlooker bee phase and the employee bee phase equation is improved to balance the exploration. Other techniques [56] applied the meta-heuristic Simulated Annealing to improve exploration and self-adaptive perturbation rate to have good exploitation.

The optimal route for data delivery in VANETs [57] is achieved by adopting ABC algorithm for sending data packets using clustering. The author uses two types of packets, scout packet to discover route and forager to forward data. The QoS constrained multicast routing problem is handled in [58] for efficient and reliable communication in multicast routing. The routing problem is transformed to SMT problem and proposed MABC algorithm applies micro-population along with variation in formula for onlooker and employee bee phase based on QoS parameters reducing the computation time.

The drawback of standard ABC is slow convergence which is handled by [59] modifying the initializing phase to generate random solutions and the search equation of the scout bee phase is modified by applying mutation operator. In [60] the authors also proposed new phases in ABC algorithm before employee phase to select the best local solution. The search equation of the onlooker and employee bee phase is modified to have good global optimal solution and convergence.

The initial population plays a major role in convergence of the algorithm, so the authors in [61] focused on initialization phase. The initial population selection, if made random, achieves good convergence. The randomness is achieved by applying opposition-based learning and chaotic systems. The mutation operator of DE is used in the search equation to have good exploration. Same technique is followed in [62] using DE/best/1 version of mutation operator along with the parameter tuning to reduce computation time and enhance convergence speed.

In [63] the initial population is arranged in an orthogonal array for better scattering of feasible solutions and new search rule is introduced with DE/rand/2 and De/best/2 to enhance exploration and convergence respectively. In [64] initial population is improved by chaotic systems and opposition based learning and search equation is modified using DE/rand/1 and DE/best/2 along with tuning the probability parameter.

The balance between the level of exploration and exploitation must be maintained to have better solution. The work in [65] focuses on Cognitive learning factor in onlooker bee and employed bee phase. The proposed technique also focused on control parameter, limit and target food number to balance the exploitation and exploration level of the standard ABC algorithm. The authors in [66] proposed two algorithms I-ABC and

PS-ABC by modifying the search process in onlooker and employed bee phase to increase the efficiency of the ABC algorithm.

Through test results the authors found that by improving the bee phases the efficiency can be improved. In [67] chaotic systems and opposition based learning applied for initialization process and adaptive tent chaotic maps along with tournament selection are introduced for searching. In [68] ABC is applied in image processing to cluster two binary vectors based on similarity measure and the difference between the best and worst vectors is selected to avoid generating poor solutions.

2.6.2 Significance of Cauchy Operator in Exploration

To avoid getting into local optima and stagnation problem these algorithms must have good exploration level. To enhance the exploration, randomness is introduced in these algorithms which plays the major role of exploration. Many researchers applied the continuous probability distribution function as random number generators and achieved good results.

The stochastic meta-heuristics optimization algorithms are based on the randomness in their process which can be derived from the distribution statistics. The distribution comes in two types: continuous and discrete distribution. The continuous distribution is good for random number generations. There are different types of continuous probability distribution such as normal, exponential, binomial, Cauchy and Rayleigh.

Cauchy or Lorentz distribution

The Cauchy distribution is a continuous probability distribution defined for the range $-\infty < x < \infty$, given by equation 2.1. The Cauchy distribution has no mean, standard deviation or higher moments with mode and median equal. It is the distribution of the ratio of two independent normally distributed Gaussian random variables.

$$f(x) = \frac{1}{\pi} \cdot \frac{1}{1+x^2} \qquad (2.1)$$

The Cauchy distribution is defined by two parameters: a, location of the center (peak) parameter and b, scaling parameter. The fat-tail symmetric Cauchy distribution is shown in Figure 2.5.

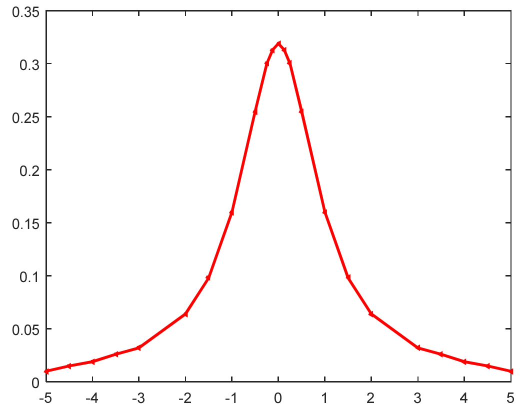


Figure 2.5: Standard Cauchy Distribution curve, Cauchy (0,1)

Comparison of Normal and Cauchy distribution

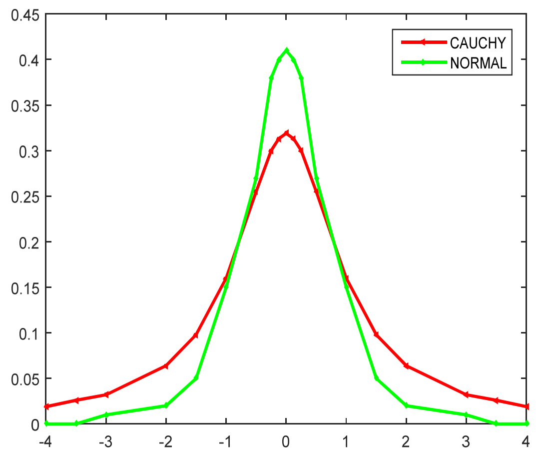


Figure 2.6: Distribution curve of Normal and Cauchy Operator

From the Figure 2.6 the distribution curve of the Cauchy distribution reveals that the curve does not reach zero at any moment and has a wider search space than the Gaussian distribution which reaches zero on both the ends after -3 and 3.

- The tail of the Cauchy distribution is fat making the variance to be infinite for exploring larger search space.
- The tail of the Gaussian distribution ends with zero making the variance to be finite resulting in small search space.

Advantages of Cauchy distribution

- Bell shaped symmetric curve as Gaussian curve but it is the ration of two Gaussian curves having longer jumps.
- Flat-tailed at both ends suits it to the real risky models.
- Coverage of large search space leading to global solution.
- Fast convergence to best feasible solution.

In Cauchy distribution, because of the fat tail, the random number generated is unexpected and are scattered around the entire search space when compared to the Gaussian distribution. So, the Cauchy distribution can be applied as random number generator to explore wider search space to find the best solution.

Many research works have been carried out in improving the evolution strategy and it is found that having good distribution of searching the problem space leads to better solution. The proposed works in [69], [70] and [71] enhanced the performance of standard Evolution strategy with Gaussian distribution for exploring new solution spaces. At that time of their work Gaussian mutation is the only choice to produce different off-springs of the current best solution.

In 1987, Szu and Hartley [75] first applied Cauchy distribution as mutation operator to develop fast simulated annealing with self-adaptation and successively in [72-74] many new mutation operators with Cauchy distribution came into origin. All the work revealed that Cauchy mutation has higher probability of longer jumps. The evolution strategy with Cauchy mutation can escape from local optima problem very quickly when compared to Gaussian mutation, having smaller jumps with predictable probability [76].

The authors in [77] proposed Cauchy mutation operator, an efficient search operator for enhancing the global search. The property of the Cauchy operator with wider search space and higher probability of longer jumps enables the optimization algorithm to find the best solution.

The global and local search efficiency is improved by incorporating two mutation operators for each criterion [78]. To explore the new region of search, Cauchy operator is applied to ensure search in the farther region and prevent local optima problem in early stage. After finding the new search space the optimal solution is searched within that region using Gaussian operator to intensify the search in the local region to obtain the better solution.

The performance of the PSO searching capability is enhanced by adopting Bayesian technique that uses the probability density function over the weight space and calculate the optimal inertia weight vector. The Bayesian uses the Gaussian distribution to exploit the local space, but suffers from local optimum problem. Since Cauchy has increased probability of longer jumps the authors used Cauchy operators in [79] to enhance the exploration.

To improve the result of biogeography optimization problem the mutation strategy by Chaos and Cauchy distribution is applied in [80]. In the initial stage chaos search is applied for random search in large space to avoid local optimum problem, and in later stages uses Cauchy operator to obtain good convergence precision.

2.7 Ant colony inspired Cellular Automata-based DDoS Mitigation

ACO (Ant Colony Optimization), popular swarm intelligence with stochastic nature meta-heuristic algorithm, takes the inspiration from the foraging behavior of the ant for solving optimization problems. The ants secrete pheromone using which they exhibit foraging behavior. It was first introduced by Marco Dorigo in 1992 to solve the travelling salesman problem. The benefits of ACO made it very popular in research field and has experienced huge growth, standing as an important nature-inspired stochastic meta-heuristic for NP-hard problems [81]. The artificial ACO makes use of forward ant and backward ant agents which are used to find the path. The node searching for destination uses forward ant agent to find the destination, and once found, backward ant agent is used to traverse the path again from destination.

Pheromone

The pheromone is a chemical substance secreted by the ants to have many followers on the same path. The path with higher pheromone content has the higher probability of being selected for next iteration. The pheromone content gets evaporated as time increases, making the path useless. This makes the evaporation rate the important factor in exploitation feature of ACO. If more pheromone is updated on the path, exploration becomes very poor and the algorithm falls into convergence at a faster rate leaving the best solutions. Thus the pheromone updation and evaporation rate are the main criteria that highly influence the exploration and exploitation of the ACO which when improved results in high performance.

The characteristics of ACO include robust, positive feedback, distributed computing and easy fusing with other algorithms making ACO simpler and efficient in searching optimal solutions. It can also be integrated with other optimization algorithms like PSO, GA to improve its global searching capability. ACO can also be improved by modifying the pheromone updating procedure, which plays the key role in ACO. The steps in ACO are depicted in Figure 2.7 with the following phases,

Initialization phase: Initialize the control parameters like pheromone value, evaporation rate.

Selecting initial solution phase: Randomly or strategically selects the initial solutions from the available solutions.

Pheromone update rule phase: The updation rule for pheromone is framed to exploit the path selected.

Evaporation Rule phase: The pheromone evaporation rule depends on the frequency of ants traversing the path.

Last two phases are repeated for each iteration, and at end of each iteration the probability of selecting the path is calculated to follow the path for next iteration.

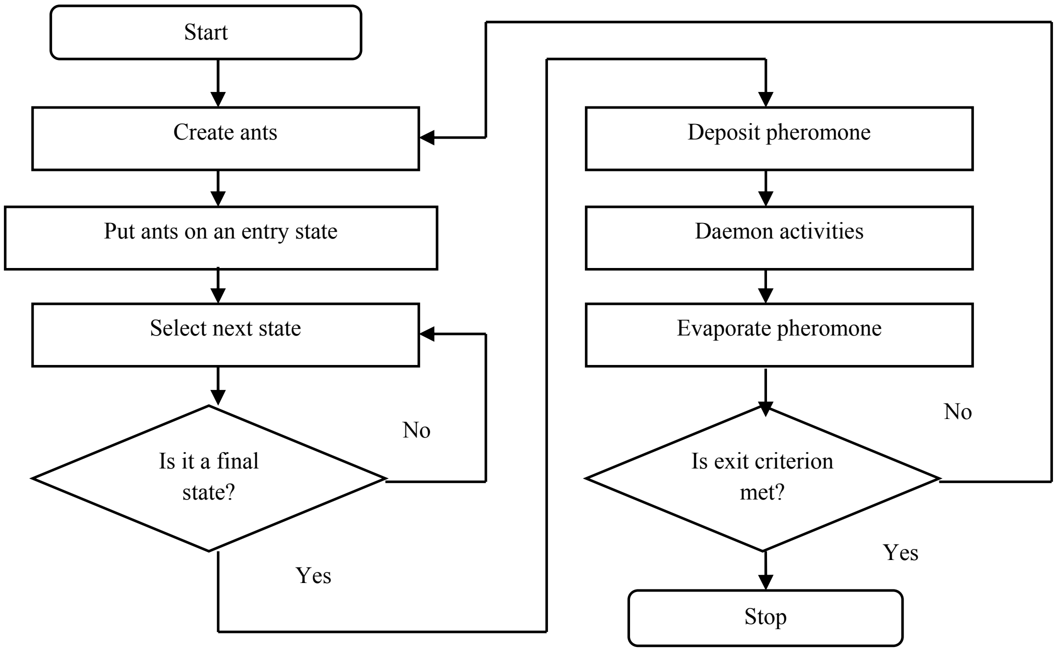


Figure 2.7: Ant Colony Algorithm

2.7.1 Enhancement of Ant colony optimization search

The problem of stagnation in ant colony is dealt in [82] by improving the standard ACO algorithm in five stages, Dynamic movement probability rule, Improved updating rules of pheromone, Adaptive adjustment of pheromone value, Dynamic Evaporation strategy and Boundary Mutation strategy. The Dynamic evaporation strategy helps in avoiding the algorithm to converge quickly and mutation helps in good exploration. In LSACA [83] the authors improved ACO by modifying the transition rule with pheromone density and heuristic information. The probability movement of ants follows the formula with information inspiration factor and expect information factor. LSACA uses 2-Opt technique to improve the solutions generated, in order to reduce the time.

To increase the efficiency and global optimal solution finding capability the chaotic signals are introduced into the dynamic movement of ants in [84]. The chaotic signal is generated using Tent Chaotic Map, a one-dimensional chaotic map which produces more auto-correlated, large number of sequences in a faster rate.

In [85] the pheromone evaporation rate is varied to improve the global searching ability and to prevent from converging to local optimum early. The pheromone deposit for every iteration is restricted based on max-min strategy to avoid stagnation in ACO. For scheduling the jog in parallel machines, in [86] ACO is improved in pheromone update rule with elite strategy and max-min ant systems to have good exploration. The objective function is set to minimize the weighted value of make-span and early due date.

Mobile Agent Routing problem is solved in [87] by using IACO (Improved ACO) to find the correct path to the destination. The ACO is improved by introducing genetic operators in global pheromone updating rule. The genetic operator, mutation, is applied in the global pheromone updating rule to escape from local optima problem, (i.e.), to have good exploration. In [88] Acoustic (sound) indicator is used to improve the ACO algorithm exploration feature. The sound strength is to decide on exploitation or exploration by measuring the relatedness of ants against a threshold. The relatedness between ants is found from difference between the sound created by ants, given in equation 2.2.

$$Relatedness = \frac{distance\ mean\text{-}number\ of\ cluster}{SD\ of\ Distance} \tag{2.2}$$

The pheromone value of the highly undesirable links is lowered, so that they are avoided from processing during finding the solution. Mostly cross links are avoided during the algorithm in the assumption that they will not lead to an optimum solution. So the [89] made use of this concept to escape from stagnation problem. The Pheromone evaluation function is based on average pheromone value of all ants in edges.

In [90] the author proposed multiple interacting ant colonies whose pheromone value is updated by the exploration factor, which is based on the parameter q_0 whose value is different for each colony of ants for different levels of exploration. Because of multiple ant colonies, stagnation problem is reduced with good exploration. In [91] the author reduces Stagnation by using dynamic candidate strategy list and adaptive heuristic parameter updation. The dynamic candidate list strategy is a sorted data structure based on increasing distance to store the closed cities to be visited. Entropy is the heuristic information adapted in this work to reduce stagnation.

To improve the global optimal solution, the concept of back tracing and diversification is adopted in [92]. Initially the agents are allowed to move and find the local and global optimal solution for 'N' number of iterations. After 'N' iteration if the global solution remains the same the back-tracing is applied by re-setting the initial position of agents to the end of current global best solution. Diversification is performed by reinitializing the pheromone value of all the edges to improve the global best solution of ACO.

The initial pheromone value also plays an important role in finding best solution. To solve TSP, the authors in [93] improved ACO in terms of, Initial pheromone value, pheromone updating rule and evaporation strategy. The edges are assigned pheromone value based on their length, so that shortest length has more pheromone compared to larger one. The pheromone transition rules are based on max-min strategy and length of the global optimum path. To get the best path from the tour it is necessary to enhance the selection probability of the path. The weight factor [94] is added to the selection probability based on the number of neighbors around and supervisory mechanism is included to refine the quality of the solution produced in the previous stage.

The multi-population strategy is applied in [95] where ants are divided into scout, search and worker ant. Scout ant is used to find the optimal path, search ant helps in exploiting the path and worker ant is used to analyze the found path and update the pheromone value. The chaotic and min-max strategy is applied to improve the global optimum value.

An improved ACO algorithm in [96] used quantum vectors for pheromone representation, dynamic pheromone updating strategy using quantum to control evaporation factor, quantum rotation gate and quantum mutation for diversification to enhance the search ability of traditional ACO algorithm.

To enhance search ability and to increase convergence speed, [97] includes non-uniform distribution of initial pheromone, heuristic strategy with direction information. Coverage and updating rule by Rowlett wheel method and evaporation coefficient is adjusted.

In [98] MACO improves the performance of ACO in TSP problem by enlarging the searching space to escape from local optimum. To have a good convergence rate the mutation technique is introduced to the ACO algorithm with the same computational complexity. The mutation is applied at the end of each iteration of ACO algorithm and finds the next solution space by exchanging any of the previous best solutions.

Sabet et al. [99] presented an enhancement in ant colony optimization process which is referred as PMACO algorithms. This algorithm improves the number of pheromone in the critical paths which in turn enhances the exploitation of the optimal solution. Experimental result proves that the PMACO algorithms are more efficient than all other enhancements in ant colony optimization algorithm.

2.7.2 Significance of mutation strategies in exploration

Generally, mutation refers to the act of producing diversity in the population of evolutionary algorithms. There are numerous mutation operators to produce the diversity in the population. This mutation can be applied in stochastic meta-heuristic optimization algorithms to improve the exploration feature of those algorithms. The performance of the stochastic algorithms is based on the exploitation and exploration measure of the algorithms in searching the global solution.

The excessive exploration makes the algorithm to jump to entire search space with higher probability and makes the convergence to be slow and excessive exploitation makes the algorithm to suffer from local optimum problem [100]. Thus there must be good trade-off between the exploration and exploitation features in any meta-heuristics algorithms. Much research work has been carried out focusing on this problem and it is found that the mutation operation applied to any standard algorithm gives better performance with same computational complexity.

Oliveto et al [101] modified the mutation strategy to have a good balance between the exploration and exploitation by introducing rank based fitness of the population and mutating each bit by the probability of the mutation. This increased the search space and resulted in fast convergence to the best solution.

Xiang, Wan-li et al [102] solved the pre-convergence problem of Differential Evolution by combining different mutation strategies of DE/current/1/bin and DE/pbest/1/bin to accelerate standard DE.

A new mutation operator is applied [103] in ACO algorithm to prevent the premature convergence and local optimum problem by checking the diversity value. If the value is less than the threshold (0.5) then the mutation operator is applied by randomly selecting an ant with low probability. The selected ant is mutated by exchanging any two bits of ant and the new fitness value is calculated.

2.8 Tabu Search inspired Cellular Automata-based DDoS Mitigation

Tabu search is meta-heuristic optimization technique which adopts adaptive memory programming [104] to solve optimization problem by maximizing or minimizing the cost function of application. It is a neighborhood search method which employs "intelligent" search and flexible memory technique which starts finding solution from a single point and moves towards the best solution [105]. If it is stuck with the local optimal location, it moves backward to select the aspiration criteria for exploring new regions of solution space.

The working flow of the Tabu search optimization algorithm is depicted in the Figure 2.8.

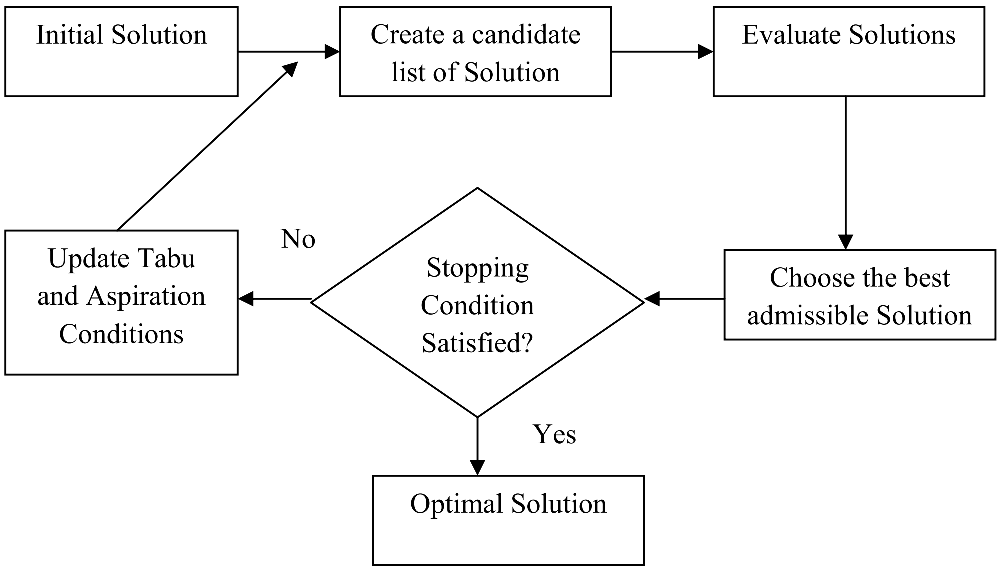


Figure 2.8: Tabu Search algorithm

Tabu search is based on introducing flexible memory structures in conjunction with strategic restrictions and aspiration levels as a means for exploiting search spaces. Memory structures [106],

- Short term -list of solutions recently considered
- Intermediate term- Intensification rules to search local space
- Long term- diversification rules to search new region

Tabu Search is applied to find the least cost multicast routing in MANET [107] by optimizing the QoS parameters bandwidth and end-to-end delay constraints. The addressing issue of the multicast routing is maintained in short term memory and long term memory is used to improve the efficiency of the algorithm. The algorithm aims to find the low cost multicasting tree with bandwidth-delay constraints. In [108] TSRA, TS based algorithm with a new move and neighborhood search method which integrate energy consumption and hop counts into routing choice. Tabu search is used to optimize the routing quality, energy consumption, and cost of routing. It is needed to achieve the tradeoff between route hops, location of nodes and energy consumption.

To determine an optimal path for the nodes in MANET Tabu search algorithm proposed in [109] carries out two neighborhood generating operations in order to determine an optimal path and minimize neighbor search time. VAST from [110] also uses Tabu Search to find an optimal path from source and destination in dense MANET. Both algorithms when compared to other techniques give minimum routing cost and less algorithm execution time. The paper [111] proposes a Tabu Search based optimization to use scarce radio network effectively by designing an effective channel assignment algorithm to reduce channel conflict.

In large VANETs environment routing problem is an NP hard, [112] Combined Nearest Neighbor Search with Tabu Search to give optimized solution of finding route to destination. The solution works in two stages, Stage 1: Nearest Neighbor Search is used to construct initial routes and Stage 2: Tabu Search is utilized to optimize the intra-route and the inter-route.

The major problem in VANETs is stable routing and their tendency to get trapped in local optimum due to dynamic topology changes. To maintain route in such environment RMRPTS: a Reliable Multi-level Routing Protocol [113] based on clustering with Tabu Search in VANETs is proposed. The protocol works by establishing Reliable routing between Cluster members using Fuzzy approach and Cluster heads to destination using Tabu search. In [114] and [115] the solutions are provided for vehicle routing problem to search neighbor based on Tabu Search. The proposal gave a new neighborhood generation procedure which considers the scattering pattern of the vehicles.

Wireless sensor Network with energy constrained nodes focus on energy saving during routing of messages. In [116] authors used Tabu search to optimize the routing quality, energy consumption, and cost of routing. The objective function is based on the tradeoff between route hops, location of nodes and energy consumption.

For enabling smoother driving on the road the vehicles on the road should not get into congestion considered as local optimum. To optimize the selection of combination of parameters of VANETs to control congestion in the network, [117] used Tabu Search algorithm UOTabu (Uni-Objective Tabu Search). The proposed algorithm uses only one objective function, delay.

Similarly, to escape from the local optimum, meta-heuristics optimization algorithm is applied in [118] for optimizing the selection of combination of parameters of VANETs to control congestion in the network, by applying Tabu Search algorithm MOTabu (Multi-Objective Tabu Search). The MOTabu uses two objective functions, delay and jitter. To optimize the geographic routing protocol in VANETs it uses Tabu Search to search path and Simulated Annealing to decrease the randomness in forwarding.

The geographic routing protocol of VANETs uses the position information of the vehicles to route the messages. To optimize the geographic routing protocol in VANET the researchers mixed the Tabu search technique with simulated Annealing in [119]. The Tabu search is used to discover the path by optimizing the objective function and Simulated Annealing to decrease the randomness in forwarding.

2.9 Extract of the Literature Survey

The cooperation among neighbors plays a major role in routing of VANET messages which are prone to many security breaches like black-hole attack, grey-hole attack, spamware, DoS, DDoS, etc,. These attacks make use of network resources inefficiently and prevent the legitimate users from accessing the resources. From the literature, DDoS is identified as more hazardous form of attack which must be prevented from happening. To enhance the performance of VANETs services the availability of neighbor should be maintained to the maximum degree.

From the literature, the parallel synchronous update of cells of CA model and strong dependency between the neighboring cells can predict the next state of the system accurately. This cooperation among the cells gives the adaptability of Cellular Automata in finding the next cooperative neighbor from the VANETs to forward message to the destination successfully.

The performance of cellular automata is dependent on the transition rules used to update the cell state. The transition rule can be optimized to enhance the effectiveness of CA model. From the survey it is inferred that integrating CA model with any meta-heuristics improves the searching behavior of the optimization algorithms.

Initially the optimization problems are solved by employing the heuristic approaches but it suffers from local optimum problem leading to pre-mature convergence to a local solution. The local optimum problem is solved by introducing a heuristic over the existing heuristics. From the survey it is inferred that the meta-heuristics optimization technique is applied to solve NP hard problems. The uncertainty plays a key role in solving the optimization problems; any heuristics with randomness introduced give stochastic meta-heuristic optimization algorithm.

The meta-heuristic stochastic optimization algorithms are utilized for the following reasons i) It is suitable for resolving any issues that could be derived in a finite dimensional space for identifying an optimal solution, ii) They are experimentally proved and confirmed to be highly suitable for approximation of solution than the heuristic stochastic optimization algorithms in most real time complex environments, iii) They possess a maximum search potential that make them highly suitable for applicability in VANETs and iv) They are proved to exhibit an higher level of precision when enhanced and integrated for maximizing the exploration extent.

Various meta-heuristics stochastic optimization algorithm used to solve large combinatorial problems are Genetic Algorithm, Evolutionary programming, Ant Colony Optimization, Artificial Bee Colony, Simulated Annealing, Tabu Search, Differential Evolution , etc,.

From review it is found that the stochastic optimization algorithms should have a good balance between the exploration and exploitation such that they do not suffer from stagnation, local optimum problem, and pre-mature convergence. The problem of ABC algorithm is that it is poor in exploitation and frequently converge to bad solution. To enhance the algorithm, the randomness is introduced in the standard ABC algorithm to reach best solution.

For random number generation it is inferred from the literature that the probability distribution functions can be applied. The main two distributions are mostly applied in literature for generating random numbers; Gaussian distribution generates Gaussian numbers or Gaussian operator, Cauchy distribution generating Cauchy numbers or Cauchy operators. From the distribution curve it is inferred that Cauchy operator is suitable for exploring new search space since it has wider search space, higher probability of longer jumps.

Similarly, from literature it is grasped that to prevent ACO from stagnation problem the exploration phase is improved by applying mutation operator, making the pheromone updation and evaporation rule dynamic with probability estimation. The mutation technique like exchange mutation is applied with other heuristic approach integrated. Finally, it is concluded that the exploration can be improved by using probability distribution operators and mutation operators.

To have a balance between the exploration and exploitation the different optimization techniques are integrated with ABC. From the literature it is inferred that the improvement in different bee phases can be applied to enhance the efficiency of the algorithm. For local optimization onlooker bee phase can be integrated with different optimization techniques like Grenade Explosion Method, Differential Evolution and ACO. For global optimization scout bee phase can be enhanced using random number generation systems like Chaotic system with opposition based learning, Cauchy operator etc., to achieve near optimal solution.

Tabu search, a trajectory based optimization meta-heuristics handles local optimum by aspiration criteria resulting in increased computation time. From the survey it is revealed that, integrating cellular automata and Tabu search increases the global searching capability of the Tabu search in reasonable time with good prediction accuracy.

Finally, the literature survey revealed that integrating the cellular automata with the improved stochastic optimization algorithms gives good prediction of the reliable neighbor for forwarding packets in VANETs. The DDoS can be prevented by selecting the best neighbor from the above mentioned technique.

2.10 Summary

This chapter presents and discusses different approaches for handling DDoS attack in VANETs. The detailed review of Cellular Automata, Stochastic Optimization algorithms are initially focused and in next sections the Artificial Bee colony algorithm, Ant Colony algorithm and Tabu search meta-heuristic stochastic optimization are explained in brief with their approaches to handle DDoS by incorporating Cellular Automata. These sections also focus on different improvements applied to the standard algorithms to increase their efficiency and performance by enhancing the exploration and exploitation phase.

CHAPTER 3

MODIFIED ANT COLONY OPTIMIZATION ALGORITHM INSPIRED CELLULAR AUTOMATA FOR MITIGATING DDoS ATTACKS

The Ant Colony Optimization algorithm is a meta-heuristic stochastic optimization technique inspired by collective behavior of decentralized, self- organized systems of ants using a unique feature called pheromone traits. ACO helps in finding the best local optimal solution since it is good in exploitation. The standard ACO algorithm on improvement in pheromone updating rule and its intensity constant, explores the global space efficiently to find the global best optimal solution with fast convergence rate. To further improve the rate of finding the optimal node, the improved ACO algorithm can be integrated with Cellular Automata Neighborhood model. The CA model promotes the efficient strategy for selecting the neighbor nodes in less time with low complexity.

3.1 Improved Ant Colony Optimization Algorithms for DDoS Mitigation

In Vehicular Ad hoc NETworks (VANETs), reliable data dissemination between vehicular nodes necessitates the maximum degree of collaboration as they play a significant role in ensuring the core objective of communication. But the malicious action of vehicular nodes may disrupt the established degree of co-operation as they lead to poor performance in spite of high resource utilization. The malicious activity of vehicular nodes like DDoS attack must be detected and resolved in a potential way by identifying optimal nodes and optimal paths.

To identify the optimal path with optimal nodes ACO has been used in many domains like WSN [120], MANET [121] and [122] with improved performance. The only problem found while applying ACO is that after some duration all nodes follow the same path leading to traffic in bandwidth [123]. This stagnation problem is reduced by many research works by modifying the rule for pheromone updation and evaporation based on length of the path, pheromone decay rate and stability of the path. Modeling the neighborhood using Cellular Automata increases the speed of convergence and eliminates stagnation problem by always selecting best neighbor for every iteration. To prevent the search from falling into local optimum point and to enhance the convergence speed, global searching potential, an enhanced version of ant colony optimization (ACO) algorithm called Modified Ant Colony Optimization Algorithm Inspired Cellular Automata (MACOA-CA) is propounded.

The Modified Ant Colony Optimization Algorithm Inspired Cellular Automata is proposed for eliminating the concept of stagnation that exists in the traditional Ant Colony-based Optimization Algorithm (ACOA). MACOA-CA used for mitigation assures an effective and efficient global search space for identifying and replacing the DDoS compromised node with optimally elected vehicular node. The traditional CA-ACO algorithm for DDoS mitigation is improved in the following dimensions viz.,

i) The movements of ants are modified based on dynamic movement probability rule,

ii) The pheromone updating rules are improved based on pheromone intensity constant,

iii) A pheromone adaptive adjustment strategy is incorporated for modifying the non-uniform distribution of pheromone to uniform distribution of pheromone and

iv) Dynamic evaporation factor strategy is used for increasing the search potential that in turn enhances the rate of convergence to a considerable level.

3.1.1 Improved Movement Rules of Ants

Naturally, Ants have the potential in searching and analyzing the quality of their food sources through optimal shortest paths that are identified by their intelligence in an iterative manner. Inspired by the intelligence of ants, Improved Ant Colony Optimization (IACO) Algorithm which is an enhanced version of Ant Colony Optimization (ACO) propounded by Marco Dorigo is used in MACOA-CA for searching the optimal vehicular nodes. MACOA-CA could aid in selecting the shortest path and alternative vehicular nodes for resolving issues that arise due to the influence of DDoS attacks during data dissemination. Since the principle of stagnation is the primitive limitation of ACO optimization process, MACOA-CA uses dynamic movement probability rule. The rule integrates the benefits of random and deterministic selection for overcoming the limitation of ACO and for improving the global selection strategy. This improvement in MACOA-CA is facilitated by formulating dynamic movement probability rule in which the paths that are maximum and minimum visited by the ant agents are used for modification of pheromones. This modification is mainly due to the evolutionary changing process of ACO.

Thus the transition rules for each search with probability 'P_{ij}^{k}' is computed based on number of ant agents 'm' , present iteration' I_p ', maximum heuristic function '$\eta_{\max}$ ' and total number of iterations used for performing global search '$Q_c(i, j)$' through equation 3.1 and 3.2 as

$$P_{ij}^{k} = \frac{R_{ij}(t)^{\alpha} * \eta_{ij}(t)^{\beta} * x_{ij}(t)}{\sum_{k \in per(k)} R_{ik}(t)^{\alpha} * \eta_{ik}(t)^{\beta} * x(t)} \qquad (3.1)$$

Where $$x_{ij(t)} = \frac{m * I_p}{m * I_p + \delta * Q_c(i,j) * (\eta(i,j)/\eta_{\max})} \qquad (3.2)$$

If the number of ant agents in MACOA-CA are continuously increased then the value of 'x_{ij}' is periodically decreased. In this context, the iteration of search approaches a suboptimal solution even when the value of pheromone is gradually improved. When the optimal nodes and the paths are identified, the exploration may lead to premature convergence due to the use of excessive amount of pheromone. But, MACOA-CA tackles it by depressing the level of pheromone based on the degree of exploration required.

The cellular automata model space used in MACOA-CA is facilitated with four points such as source vehicle point, destination vehicle point, intermediate router node points and free space that follows Moore model. The distance between the source vehicle point and destination vehicle point is considered 'L' and the intermediate router nodes can move around a width of 'L/2'. Thus the space is represented by equation 3.3.

$$S = (x, y) / x \in \{0,.., X_{max}\}, y \in \{0,..., Y_{max}\} \qquad (3.3)$$

The position of each vehicle pertains to a point (x, y), when S (x, y) = 1, the collision of vehicles is possible and S (x, y) = 0 represents the collision free space of the cellular automata.

The cellular automata model employed in MACOA-CA consists of cells and cellular spaces for discretizing time and space in the search space. The time and space are discretized for analyzing and describing the dynamic behavior of vehicular nodes in the 2D space as shown in Figure 3.1a and Figure 3.1b. Each and every cell employed in the lattice grid space exhibits finite number of discrete states and the behavioral states are updated based on the newly innovated local space rule. This complex dynamically changing behavioral state process identification is modeled into a discrete interactive process. The Cellular Automata Moore model used in MACOA-CA is represented using a 4-tuple $C_A = (L_d, S_c, N_c, R_c)$

Where

C_A : Moore model based cellular automata.

L_d : Cellular space with 'd' positive dimension used in the Moore model (d=2 in MACOA-CA).

S_c : Possible state space of cellular automata (0, 1).

N_c : Neighbors of each individual cellular cell defined based on Moore model represented through S=$\{n_1, n_2, ..., n_r, ... n_n\}$ .where 'S' and 'r' denotes the spatial vector that incorporates 'n' feasible cellular states and direction of the ant colony respectively. In MACOA-CA, $r \in [1,8]$ and $s_r \in [0,1]$ which infers that the transition will decide to choose any direction, else if $s_r = 1$, the ant colony optimization is not possible.

R_c : Rule for cellular transition or cellular state transformation function.

The Moore cellular automata neighborhood model is used in the proposed work since each focuses on all the surrounding neighbors in eight directions (North, North-East, East, South, South-East, North-West, South- West and West). The MACOA algorithm finds the probability of selecting a node as next forwarding node among the eight surrounding neighbors by applying equation 3.1.

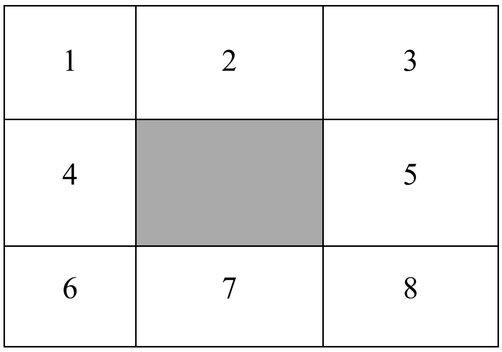


Figure 3.1a: Moore Model for MACOA-CA

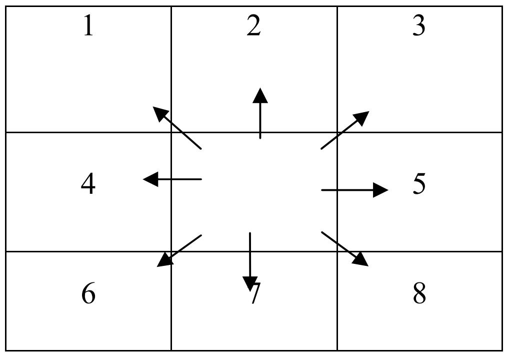


Figure 3.1b: Direction of transfer in MACOA-CA

3.1.2 Enhanced Pheromone updation rules of MACOA-CA

The pheromone rules for MACOA-CA are enhanced by updating the residual pheromone after each iterative search as the heuristic factor of search should not be integrated with the estimated residual pheromone data elucidated. Hence the pheromone rules for MACOA-CA are improved using equation 3.4, 3.5, 3.6 and 3.7.

$$R_{ij}(t+1) = \rho R_{ij}(t) + \Delta R_{ij} + \Delta R_{ij}^{\ c} \qquad (3.4)$$

$$\Delta R_{ij}^{\ c} = \sum_{k=1}^{m} \Delta R_{ij}^{\ k} \qquad (3.5)$$

$\Delta R_{ij}^{k} = \frac{Q}{L_k}$ (The path (i, j) traversed by each ant agent in specific iterations) (3.6)

$\Delta R_{ij}^{\ c} = \frac{\delta Q}{L_c}$ (The path (i, j) is the identified optimal solution) (3.7)

Where, 'Q'-Pheromone intensity constant

'$R_{ij}(t)$' - Pheromone value at time 't'

'$R_{ij}(t+1)$' - Pheromone value at time't+1'

'ρ' - Evaporation factor rate

The pheromone updation for iteration is constant and it depends on three parameters as follows,

a. Pheromone evaporation or decay rate.
b. Previous pheromone deposit value.
c. Optimal pheromone deposit value up to i^{th} iteration.

These parameters help ACO to improve its efficiency by exploring new route to the destination. Variance in pheromone deposit value increases the probability of other alternate good routes to be selected resulting in no stagnation in local optimum location. This increases possibility of more or all feasible routes to be selected for forwarding the messages to the destination in reasonable time in both dense and sparse environments.

In MACOA-CA with the cellular automata rules optimized by the improved ACO pheromone updation, each vehicle's neighbor information such as trust and reference velocity are updated using the equation 3.4. Since the VANETs environment is dynamic the routes change from time to time so we apply the improved ACO to predict the next best neighbor for packet forwarding.

3.1.3 Pheromone Adaptive Adjustment Strategy for MACOA-CA

In the traditional ACO algorithm, fixed level of pheromone is used for updating the pheromone-based search strategy. This kind of strategy ignores several characteristics of distribution that pertains to the identified solution of each iteration and hence it is susceptible to the process of stagnation and slow convergence. The stagnation makes ACO to fall into a local optima in the premature stage of searching that prevents the algorithm from identifying the optimal vehicular nodes for handling DDoS attacks. Therefore, a Pheromone Adaptive Adjustment Strategy for MACOA-CA is incorporated to convert the non-uniform distribution to a relative uniform pheromone distribution that resolves the trade-off that exists between the deviation in search expansion and the exploration of an optimal solution for attaining the local optimum solution.

In this Pheromone Adaptive Adjustment Strategy of MACOA-CA, the influential function referred to as pheromone intensity constant 'ρ' is replaced by a real dynamic ranging variable function '$V(t)$' based on adjusting pheromone represented through equation 3.8 as

$$\Delta R_{ij(t)}^{(k)} = \frac{V(t)}{L_k} \qquad (3.8)$$

The aforementioned real dynamic ranging variable function '$V(t)$' defined in MACOA-CA is portrayed from equation 3.9 to equation 3.11 as follows

$$V(t) = V_1, t \leq Th_1 \qquad (3.9)$$

$$V(t) = V_2, t \leq Th_2 \qquad (3.10)$$

$$V(t) = V_3, t \leq Th_3 \qquad (3.11)$$

Where, V_1, V_2 and V_3 refer to three levels of tolerance pertaining to dynamic pheromone intensity. Th_1, Th_2, Th_3 represent the number of vehicular nodes in a particular cell.

The replacement of real dynamic ranging variable function is mainly to sustain the balance that exists between exploitation and exploration keys of dynamic searching of ant agents. The real dynamic ranging variable aids in maintaining the evocation function to a constant level under the influence of pheromone evaporation. It also checks whether the optimal quality solution is constant over a period of time under searching and it also seems to prevent the search to suddenly fall into a critical point of convergence. Then the Pheromone Adaptive Adjustment Strategy plays a vital role in decreasing the level of information used for searching the critical point convergence. Once the critical point convergence is found, the number of optimal paths is discriminated from worst paths based on the positive feedback scheme used by the classical ACO. But in MACOA-CA, considerable amount of negative feedback pheromone is used in the search for minimizing the deviation in pheromone of each and every optimal path solution identified. This Pheromone Adaptive Adjustment Strategy step also aids in expanding the possibility of global search.

The pheromone updation is made adaptive based on the dynamic situation in the network. The pheromone updation value is varied based on the number of vehicles in the surroundings, when the traffic is more or dense the updation value is V_1, during medium traffic the updation value is V_2 and if traffic is less the updation value is V_3. Thus the updation value is made adaptive and adjustable to the traffic in the network to enhance the exploration level of the algorithm in finding the best neighbor.

3.1.4 MACOA-CA Dynamic Evaporation Factor Strategy

In MACOA-CA**,** the evaporation factor ' ρ ' of the pheromone cannot be a constant as it directly represents the convergence speed and global searching ability of the algorithm. Generally, the Evaporation Factor of ACO in the least unexplored region and paths converges to 0. This property of evaporation factor convergence greatly minimizes the global searching potential of the implemented algorithm. In reverse, if pheromone used is high, it also affects the global searching potential of ACO.

The idea of initializing the pheromone's value remains a critical issue that needs to be dynamically resolved as it is the core mechanism of controlling and synchronizing the rate of release and evaporation. Dynamic Evaporation Factor Strategy in MACOA-CA is necessary for setting the value of pheromone for improving the global exploring skill of the deployed algorithm. The utilized Dynamic Evaporation Factor Strategy in MACOA-CA is capable of improving the global search potential and further induces the rate of convergence to a significant level. This strategy uses the idea of incorporating maximum value to the dynamic evaporation factor at the initial state for improving the rate of searching potential. Even when the evaporation factor is initially high, they gradually decay and start to converge into an optimal solution. To study the decay rate of evaporation factor, three decay models, scale decay model, line decay model and curve decay model, can be used. In MACOA-CA, Dynamic Evaporation Factor is estimated based on curve decay model through equation 3.12 as

$$\rho(t) = \frac{T(R_{\max} - R_{\min})t}{T-1} + \frac{T(R_{\min} - R_{\max})}{T-1} \tag{3.12}$$

where 'ρ', 'T', 't', '$R_{\max}$', '$R_{\min}$' refers to the evaporation rate of pheromone, maximum number of iterations used for identifying optimal nodes and optimal paths, minimum number of iterations used for identifying optimal nodes and optimal paths, upper and lower threshold of pheromone value respectively.

In MACOA-CA, curve delay model is mainly used for investigating the decay rate of evaporation factor as it is the predominant model that can discriminate the deviation that exists between the evaporation and release rate of pheromone in a significant way.

3.1.5 Boundary Symmetric Mutation Scheme of MACOA-CA

Statistical theory infers that most of the distribution will tend to be normal or will meet normal distribution based on the increasing number of vehicular nodes used in the cellular automata model. The co-ordinates of each vehicular node are initially sorted based on the co-ordinates of vehicular nodes in the cellular automata model. When the number of nodes in the cellular co-ordinate is small, the global search is carried based upon the idea of centrotaxis. In CA model the selected optimal nodes for handling DDoS will be concentrated in the center and the optimal nodes for routing are also elected with respect to the same phenomenon.

In MACOA-CA approach, initially the ant agents are made to explore the possibilities starting from the boundary towards the center of the employed Moore-based cellular automata. After each and every incremental time, the search is performed from the center towards the boundary. Thus ant agents in MACOA-CA are made to obey the trajectory model of boundary-center-boundary during the exploration of identifying the optimal nodes and optimal paths for mitigating DDoS attacks.

The trajectory model is used to overcome the perplexing paths that might arise during the exploration in the center area and further, a number of overlapping paths with critical limitations may arise in the boundary paths. The mutation strategy adopted here is to select the node which occurs in the intersection of the path from boundary to center and center to boundary. The intersecting node is identified as being in close related velocity with the other nodes in the network. Furthermore, effective mutation strategy is integrated with MACOA-CA for estimating the quantification of mutation degree along the boundary paths. This boundary symmetric mutation strategy not only enhances the mutation efficiency but also helps to achieve better quality exploration results.

In MACOA-CA algorithm, the idea of probabilistic quartile is used to mutate nearly one-third from the initial and end of the paths explored. Thus mutation in the boundary happens only in the interior part of the limits and does not happen exterior to the exploration area. Hence, Cellular Automata-based Improved Ant Colony-based Optimization Algorithm combines improved dynamic transition rules of ant agents, enhanced update rules of pheromone, pheromone's adjustment strategy of pheromone and dynamic evaporation factor strategy with boundary symmetric mutation for speeding the rate of search.

3.1.6 Flow Chart of proposed MACOA-CA

The steps of MACOA-CA used in improving the search quality for identifying optimal nodes for handling DDoS attacks and finding optimal quality path solutions are depicted in Figure 3.2.

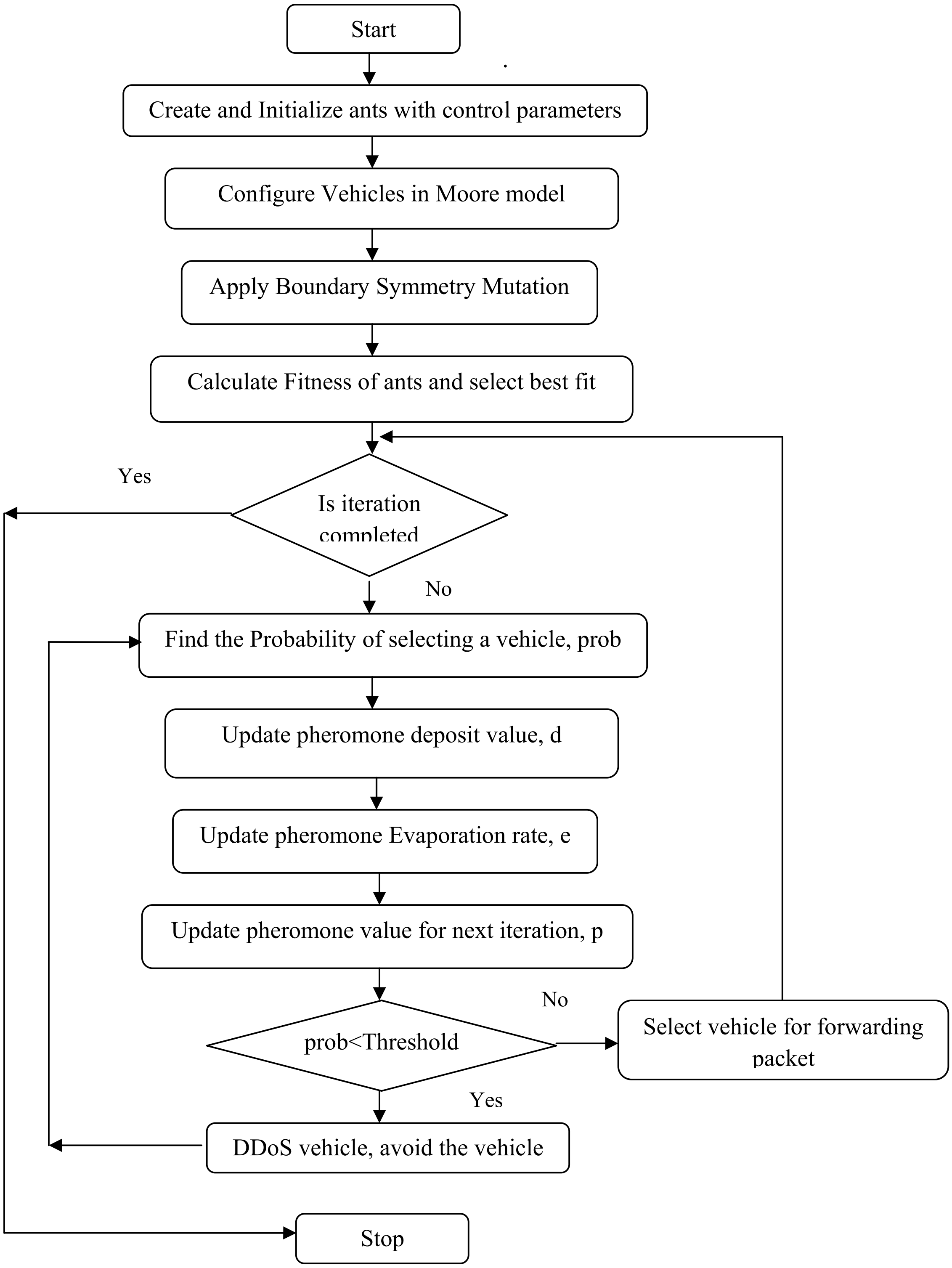


Figure 3.2: Flow Chart for MACOA-CA algorithm

Algorithm 3.1: **MACOA-CA (Modified Ant Colony Optimization Algorithm Inspired Cellular Automata)**

Step 1: Set the parameters.

Step 2: Find the fitness function of each vehicular nodes.

Step 3: Deploy the vehicles and ants in Moore model and update their position.

Step 4: When the cache list is not empty, select the vehicular nodes by the selection probability.

$$P_{ij}^{k} = \frac{R_{ij}(t)^{\alpha} * \eta_{ij}(t)^{\beta} * x_{ij}(t)}{\sum_{k \in per(k)} R_{ik}(t)^{\alpha} * \eta_{ik}(t)^{\beta} * x(t)}$$

Step 5: After the ant agents have been identified the optimal paths and optimal node, the path length is calculated and updated in the cache list.

Step 6: The current optimal node and path are saved and the global optimal path are updated in each iteration.

Step 7: Update the pheromone according to the updating rules of pheromone.

$$R_{ij}(t+1) = \rho R_{ij}(t) + \Delta R_{ij} + \Delta R_{ij^{c}}$$

Step 8: Set the iteration control by periodic increments of 1 based on pheromone adaptive adjustment, Dynamic evaporation factor and Boundary symmetric mutation scheme.

Step 9: The iteration control is processed until t<T_{max} go to step (4).

Step 10: Otherwise, MACOA-CA is terminated and the identified optimal node and path are used for mitigation DDoS attack.

3.2 Simulation Experiments and Results

The performance investigation of MACOA-CA with CA-ACOA, CA-GA and CA-PSO is evaluated based on ns-2 simulator and SUMO traffic simulator is used for generating vehicular mobility traces. Initially, evaluation parameters such as prediction variance (meters), prediction variance (seconds) and average prediction variance are used for investigating the performance of MACOA-CA. In this performance analysis, the experiments are carried out either by varying the number of nodes or by varying the prediction interval. The comparative analysis of MACOA-CA is based on the experiments discussed below. In the first three experiments, the performance of MACOA-CA with CA-ACOA, CA-GA and CA-PSO is analyzed through evaluation factors such as variance (meters), prediction variance (seconds) and average prediction variance obtained by varying the prediction interval and vehicular nodes.

The simulation setup used for comparative performance analysis of MACOA-CA with CA-ACOA, CA-GA and CA-PSO are detailed in Table 3.1.

Table 3.1: Simulation setup for evaluating MACOA-CA

Parameters	Value
Number of vehicular nodes	100, 200, 300
Range of transmission	600m
Threshold speed	40-60 m/sec
Acceleration of vehicular node	1.2 m/s^2
Retardation of vehicular node	6.5 m/s^2
Simulation time	600s
Prediction interval	10-140 s
MAC protocol	IEEE 802.11p
Refresh interval time	50s

Experiment 1-Performance analysis of MACOA-CA based on prediction variance (meters) by varying prediction interval

Prediction variance in meters

The performance metric prediction variance is the difference between the predicted position of a vehicle and subsequent actual position that the vehicle reaches at any instant.

In experiment 1, the prediction variance is measured in meters by varying the prediction interval by 10, 20, and up to 100 seconds. The predicted position of the neighbor is measured against each interval and plotted. In experiment 1, the performance investigation of MACOA-CA over existing cellular automata based DDoS mitigation approaches like CA-ACOA, CA-GA and CA-PSO are investigated.

Figure 3.3 portrays the performance of MACOA-CA in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 100 vehicular nodes. Results make it clear that the prediction variance for MACOA-CA, CA-ACOA, CA-GA and CA-PSO increases phenomenally when prediction interval is varied from 10 to 100 seconds. But, MACOA-CA is able to dynamically decrease the variation especially when the prediction interval increases. MACOA-CA is found to exhibit a decrease in prediction variance to a maximum level of 16% than the compared baseline approaches.

Figure 3.4 portrays the performance of MACOA-CA in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 200 vehicular nodes. From the graph it is clear that the prediction variance of MACOA-CA also increases gradually as CA-ACOA, CA-GA and CA-PSO increases when prediction interval is varied from 10 to 100 seconds under the influence of 200 nodes. But, MACOA-CA shows a small percentage increase in variance when compared to other existing algorithms. MACOA-CA is found to exhibit a decrease in prediction variance to a maximum level of 19% than the compared baseline approaches.

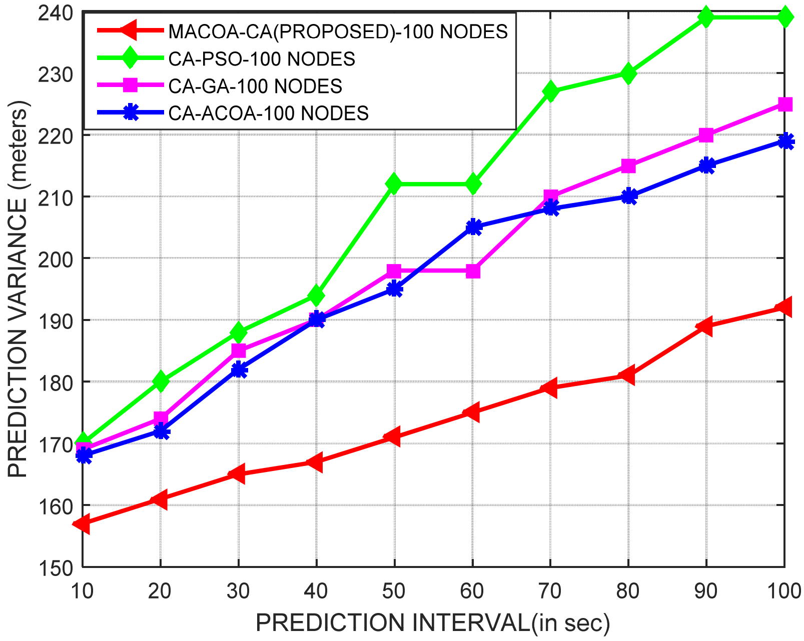


Figure 3.3 -Experiment 1-Performance of MACOA-CA-Prediction variance (meters)-100 nodes

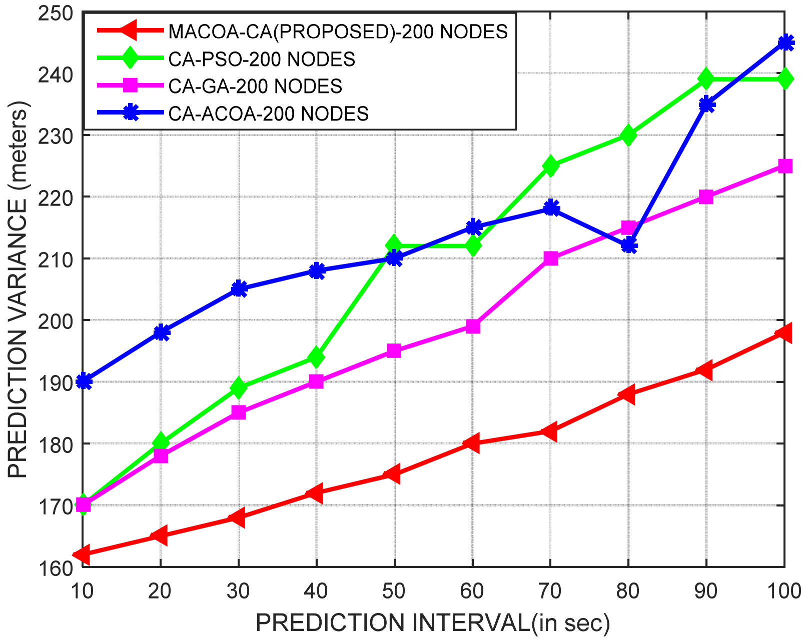


Figure 3.4 -Experiment 1-Performance of MACOA-CA -Prediction variance (meters)-200 nodes

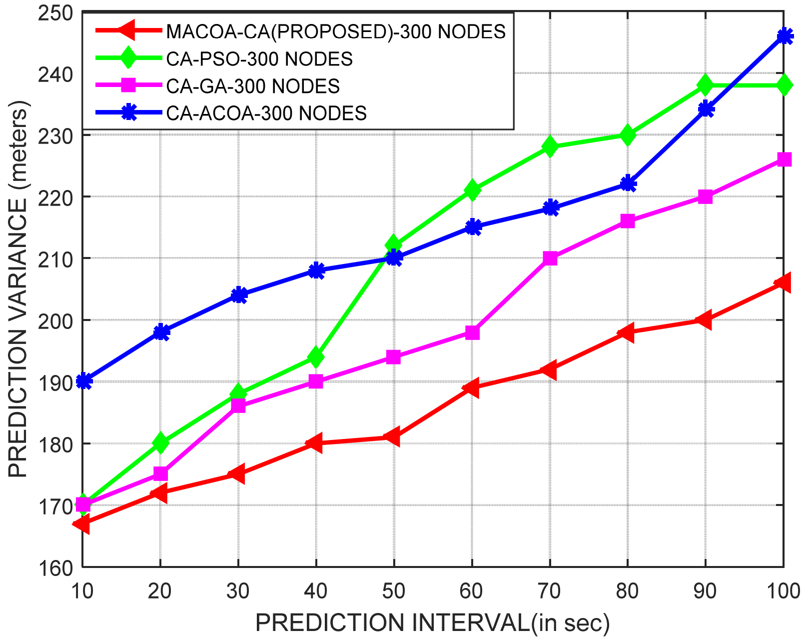


Figure 3.5 - Experiment 1-Performance of MACOA-CA- Prediction variance (meters)-300 nodes

The performance of MACOA-CA in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 300 vehicular nodes is portrayed in Figure 3.5. For 300 vehicles the prediction variance has a remarkable increase as the prediction interval increases similar to the other algorithms like CA-ACOA, CA-GA and CA-PSO. But, MACOA-CA has a meagre increase in prediction variance under 300 nodes compared to existing algorithms. From the graph it is inferred that MACOA-CA is found to exhibit a decrease in prediction variance to a maximum level of 23% than the compared existing approaches.

Experiment 2- Performance analysis of MACOA-CA based on prediction variance by varying number of nodes

Prediction Variance in seconds

The second performance metric prediction variance in seconds is the prediction accuracy of a node in predicting its position for every unit time interval.

In experiment 2, the performance analysis of MACOA-CA, CA-ACOA, CA-GA and CA-PSO is carried out in terms of prediction variance per second by varying the number of vehicular nodes from 100 to 300 based on varying prediction interval.

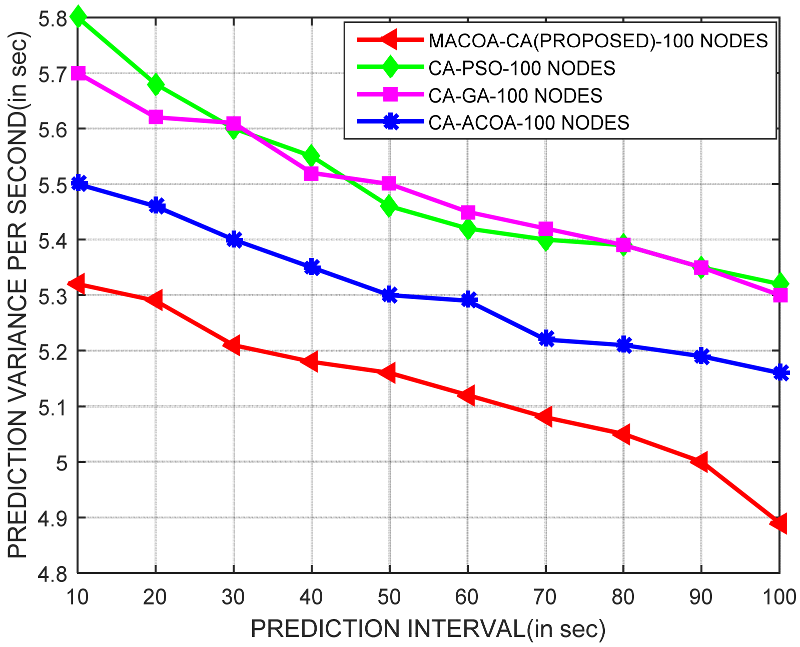


Figure 3.6- Experiment 2-Performance of MACOA-CA-Prediction variance per second-100 nodes

Figure 3.6 describes the performance of MACOA-CA, CA-ACOA, CA-GA and CA-PSO based on prediction variance in seconds evaluated by varying the prediction interval. It is found that the prediction variance per second for MACOA-CA, CA-ACOA, CA-GA and CA-PSO is considerably decreasing when the prediction interval is increased. But MACOA-CA is able to increase the dimension of searching and prevents it from being trapped into local minimum. Hence MACOA-CA decreases the prediction variance in seconds than CA-ACOA, CA-GA and CA-PSO to a considerable level of 21%, 23% and 26% under the influence of 100 nodes.

The graph in Figure 3.7 depicts the performance of MACOA-CA, CA-ACOA, CA-GA and CA-PSO based on prediction variance in seconds evaluated by varying the prediction interval with 200 nodes. It is found that the prediction variance per second at start up stage is more for MACOA-CA, CA-ACOA, CA-GA and CA-PSO but is considerably decreases as the interval time increases. The MACOA-CA decreases the prediction variance in seconds than CA-ACOA by 15%, CA-GA by 17% and CA-PSO by 20% .

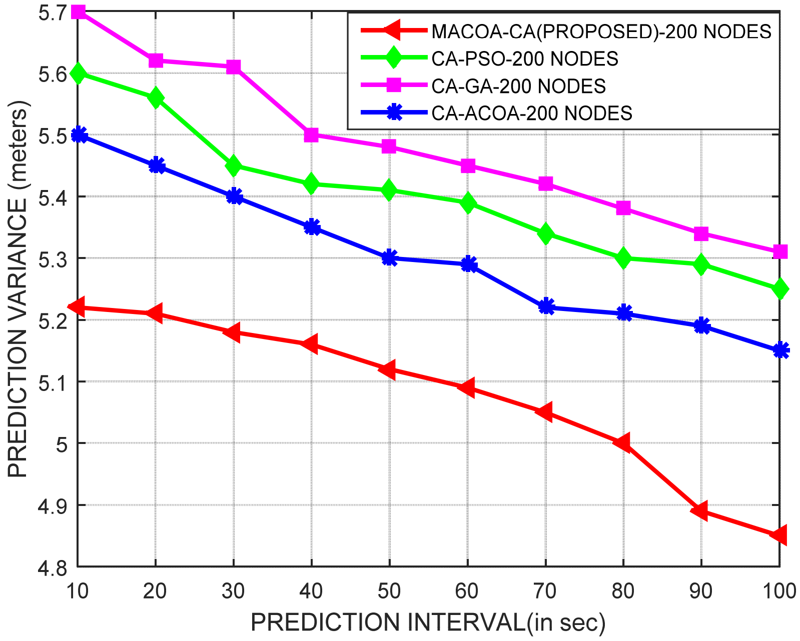


Figure 3.7 - Experiment 2-Performance of MACOA-CA-Prediction variance per second-200 nodes

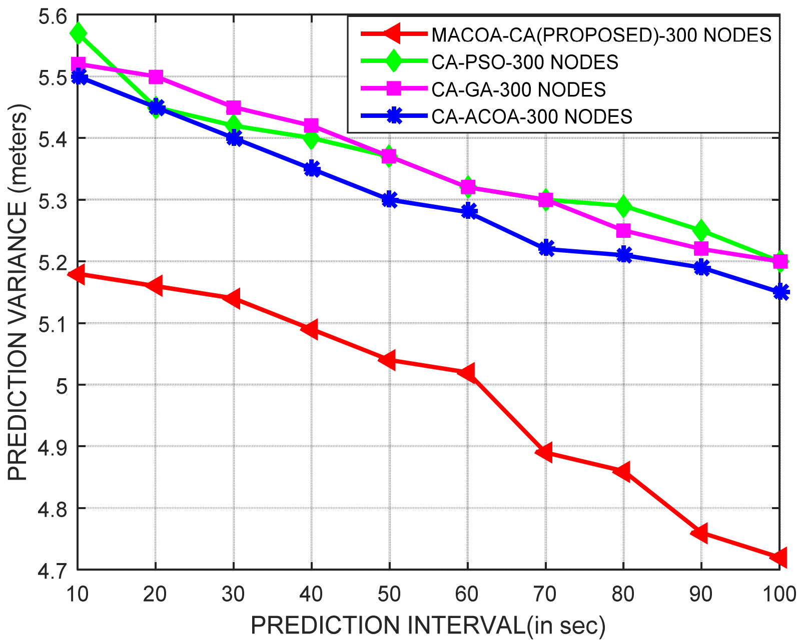


Figure 3.8: Experiment 2-Performance of MACOA-CA- Prediction variance per second-300 nodes

The prediction variance in seconds for 300 nodes is shown in Figure 3.8 for MACOA-CA, CA-ACOA, CA-GA and CA-PSO by varying the prediction interval. When compared to the baseline approaches MACOA-CA has lower prediction variance in seconds due to its balanced exploration and exploitation level. For 300 nodes, MACOA-CA decreases the prediction variance in seconds than CA-ACOA, CA-GA and CA-PSO to a remarkable level of 11%, 14% and 18% respectively.

Experiment 3- Performance analysis of MACOA-CA based on prediction variance by varying number of nodes

Average Prediction Variance

The third performance metric average prediction variance is defined as the prediction accuracy of the vehicle over a 'n' number of consecutive prediction. It is calculated by measuring the prediction variance of the vehicles of some n prediction and taking its average.

In experiment 3, the performance of MACOA-CA, CA-ACOA, CA-GA and CA-PSO is investigated in terms of prediction variance (meters) by varying the number of vehicular nodes.

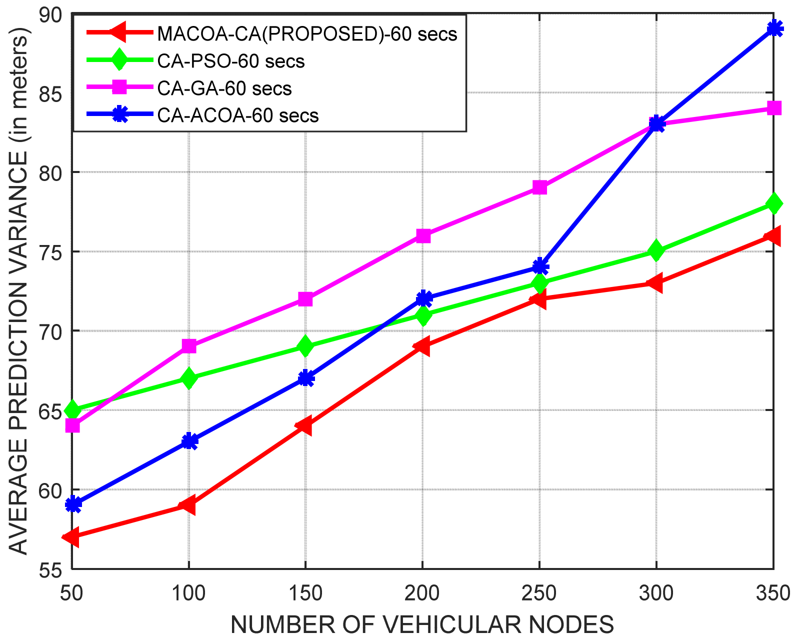


Figure 3.9 - Experiment 3-Performance of MACOA-CA- Average Prediction variance-60 seconds

Figure 3.9 portrays the relationship that infers the optimal average prediction accuracy facilitated by MACOA-CA, CA-ACOA, CA-GA and CA-PSO when vehicular nodes are significantly varied. This investigation is initially with an accuracy level of 60sec as it is considered the minimum optimal value for accuracy prediction in MACOA-CA, CA-ACOA, CA-GA and CA-PSO mechanism. The average prediction variance of MACOA-CA shows a meagre variation of approximately 56-60m with varying number of vehicular nodes, whereas CA-ACOA, CA-ACOA, CA-GA exhibits a deviation of 69-74m, 80-84m and 91-97m respectively.

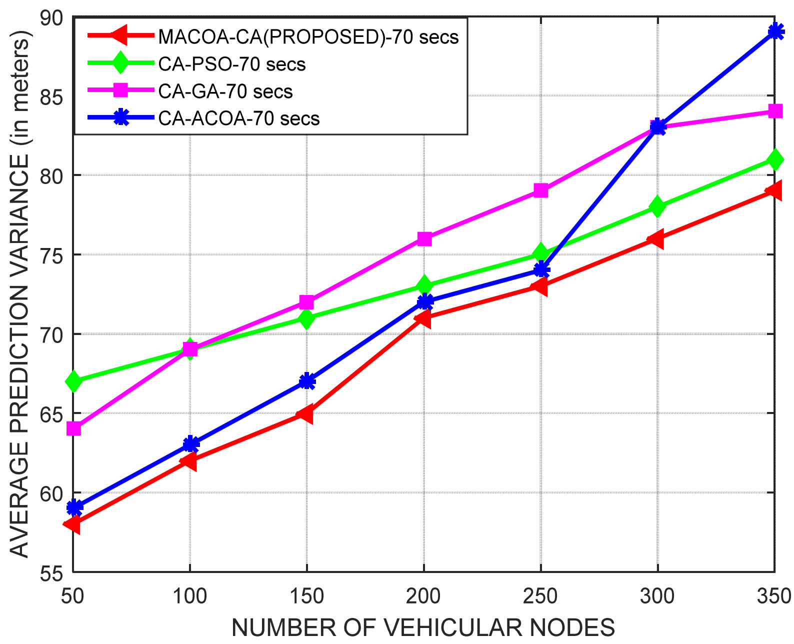


Figure 3.10 - Experiment 3-Performance of MACOA-CA- Average Prediction variance-70seconds

Similarly, the average prediction variance is measured for 70 seconds by varying the number nodes and the result is depicted in Figure 3.10. The graph shows that as the number of nodes increases the average prediction interval increases for MACOA-CA, CA-ACOA, CA-GA and CA-PSO. But, MACOA-CA due to its better optimization level shows small percentage increase compared to other existing works. The average prediction variance of MACOA-CA shows a meagre variation of approximately 52-55m with varying number of vehicular nodes, whereas CA-ACOA, CA-GA and CA-PSO exhibits a deviation of 59-64m, 67-73m and 77-80m respectively.

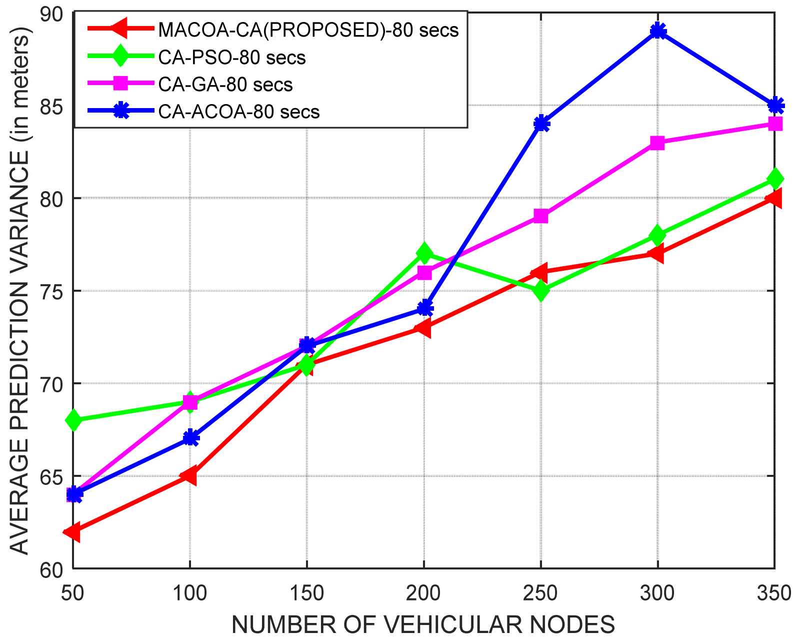


Figure 3.11 - Experiment 3-Performance of MACOA-CA- Average Prediction variance - 80 seconds

In addition, Figure 3.11 also portrays the relationship that infers the optimal average prediction accuracy facilitated by MACOA-CA, CA-ACOA, CA-GA and CA-PSO when vehicular nodes are significantly varied under an accuracy level of 80sec. For 80 seconds the average prediction variance of MACOA-CA has a gradual increase with minimum percentage level. The average prediction variance of MACOA-CA shows a meagre variation of approximately 50-53m with varying number of vehicular nodes, whereas CA-ACOA, CA-GA and CA-PSO exhibits a deviation of 57-64m, 68-73m and 78-89m respectively.

The result analysis of the experiment conducted for the performance metrics prediction variance (seconds), prediction variance (meters) and average prediction variance is listed below.

Table 3.2 Performance comparison of Prediction variance (measured in meters) proposed MACOA-CA algorithm with existing algorithms

Algorithm	No. of vehicles		
	100	200	300
CA-ACOA vs MACOA-CA	13	20	29
CA-GA vs MACOA-CA	15	11	24
CA-PSO vs MACOA-CA	20	21	32

Table 3.2 gives the comparison of prediction variance of proposed algorithm with other baseline approaches with the decrease in percentage. For 100 vehicles the decrease in prediction variance is between 13-20 % when compared to the CA-ACOA, CA-GA, CA-PSO algorithms. For 200 vehicles the decrease in prediction variance is between 11-21% when compared to the CA-ACOA, CA-GA, CA-PSO algorithms. For 300 vehicles the decrease in prediction variance is between 24-32 % when compared to the CA-ACOA, CA-GA, CA-PSO algorithms. Due to dynamism and adaptive adjustment strategy MACOA-CA algorithm results in better prediction of the neighboring vehicles for forwarding the packet to the destination with no increase in complexity and running time.

Table 3.3 Performance comparison of Prediction variance (measured in seconds) proposed MACOA-CA algorithm with existing algorithms

Algorithm	No. of vehicles		
	100	200	300
CA-ACOA vs MACOA-CA	4	5	6
CA-GA vs MACOA-CA	7	9	6
CA-PSO vs MACOA-CA	8	7	6.5

Table 3.3 gives the comparison of prediction variance in seconds of the proposed algorithm with other baseline approaches with the decrease in percentage. For 100 vehicles the decrease in prediction variance is between 4-8 % when compared to the CA-ACOA, CA-GA, CA-PSO algorithms. For 200 vehicles the decrease in prediction variance is between 5-7 % when compared to the CA-ACOA, CA-GA, CA-PSO algorithms. For 300 vehicles the decrease in prediction variance is between 6-6.5 % when compared to the CA-ACOA, CA-GA, CA-PSO algorithms. Due to dynamism and adaptive adjustment strategy MACOA-CA algorithm results in better prediction of the neighboring vehicles for forwarding the packet to the destination with no increase in complexity and running time.

Table 3.4 gives the comparison of average prediction variance of the proposed algorithm with other baseline approaches. The experiment was conducted by varying the number of nodes from 50 to 350 and measuring the average prediction variance for each group.

Table 3.4 Performance comparison of Average Prediction variance (measured in meters) proposed MACOA-CA algorithm with existing algorithms

Algorithm	**Accuracy interval level(in meters)**		
	60s	**70s**	**80s**
CA-ACOA	58-89	59-89	68-85
CA-GA	64-84	64-84	63-84
CA-PSO	65-77	68-61	63-81
IACOA(Proposed Work)	**57-76**	**58-78**	**62-80**

The result from the table 3.4 reveals that for prediction interval of 60s the proposed algorithm has average prediction variance from 57-76m which is less when compared to all other existing algorithms. Similarly, the average prediction variance is measured by varying the number of vehicles with different prediction intervals of 70s and 80s and is found that the proposed algorithm has less average prediction variance among all existing algorithms.

3.3 Summary

In this chapter MACOA-CA with improved dynamic evaporation rule and adaptive pheromone update rule is proposed to achieve better exploration and exploitation for removing the stagnation problem. The proposed work successfully mitigates the DDoS by selecting the reliable node from the CA Moore model using improved ACO algorithm for forwarding the packet to the destination. The simulation clearly shows that the prediction variance and delay is decreased with improved PDR by applying the MACOA-CA algorithm for data routing. The result analysis also reveals that the proposed algorithm MACOA-CA has better prediction over the existing algorithms with no increase in complexity and running time.

CHAPTER 4

CELLULAR AUTOMATA BASED IMPROVED ARTIFICIAL BEE COLONY A0LGORITHM (CA-IABCA)

The Artificial Bee Colony algorithm based on the intelligent foraging behavior of the honey bee swarm with its good exploration, helps in finding the optimal solutions for optimization problem within a reasonable time. The neighbor vehicle with good degree of cooperation can be obtained by applying ABC algorithm with better exploitation and exploration level. The performance of the algorithm can further be improved by adopting good randomness in scout bee phase and introducing other meta-heuristics algorithms in onlooker bee phase. The improved ABC algorithm can be integrated with simple, abstract CA neighborhood model for reducing the complexity of the algorithm.

4.1 Grenade Explosion-Based Artificial Bee Colony Algorithm

The optimization techniques are considered to be the most efficient way of solving the problems with high dimensionality, lack of resource and no analytical model to solve them in reasonable time [124]. The artificial bee colony algorithm starts with the random set of population agent to explore the problem space for finding the best optimal solution. Then in next succeeding iterations part of the near optimal solutions are relocated based on some logical function to hope that they move towards the global optimal point [125].

The grenade, a small bomb designed for short range use explodes shrapnel when thrown by hand. The shrapnel is capable of destructing the object it is hitting, and the region with high damage is selected for next grenade explosion since it is assumed that the region has worthy objects. This property of grenade explosion is applied in GEM optimization technique to find the global solution [126]. The GEM has the property to converge to the exact location of the global optimum [127] in reasonable time. Thus GEM has the property of fast convergence, reliable and efficient in finding the global optimum solution compared to other optimization algorithms.

4.1.1 2D-Space cellular model configuration for CA-IABCA

The cellular automata model space used in CA-IABCA is facilitated with four points such as source vehicle point, destination vehicle point, intermediate router node points and free space that follows Moore model. The distance between the source vehicle point and destination vehicle point is considered as 'L' and the intermediate router nodes can move around a width of L/2.Thus the space is represented using equation 4.1.

$$S = \{(x, y) / x \in \{0,...,X_{\max}\}, y \in \{0,....,Y_{\max}\} \quad (4.1)$$

The position of each vehicle pertains to a point (x, y), when S(x, y) =1, the collision of vehicles are possible and if S(x, y) =0 represents the collision free space of the cellular automata.

4.1.2 Evolution rules employed for CA-IABCA

The cellular automata model employed in CA-IABCA consists of cells and cellular spaces for discretizing time and space in the search space. The time and space are discretized for analyzing and describing the dynamic behavior of vehicular nodes in the 2D space as shown in Figure 4.1a and Figure 4.1b. Each and every cell employed in the lattice grid space exhibits finite number of discrete states and the behavioral states are updated based on the newly innovated local space rule. This complex dynamically changing complex behavioral state process identification is modeled into a discrete interactive process.

The Cellular Automata Moore model used in CA-IABCA is represented using a 4-tuple $C_A = (L_d, S_c, N_c, R_c)$,

Where,

C_A: Moore model based cellular automata.

L_d : Cellular space with 'd' positive dimension used in the Moore model (d=2 in CA-IABCA).

S_c: Possible state space of cellular automata (0, 1).

N_c: Neighbors of each individual cellular cell defined based on Moore model represented through S= $\{n_1, n_2, \ldots\ldots n_r, \ldots n_n\}$.where 'S' and 'r' denotes the spatial vector that incorporates 'n' feasible cellular states and direction of the artificial bee colony respectively. In CA-IABCA, $r \in [1,8]$ and $s_r \in [0,1]$ which infers that the transition will decide to choose any direction, else if $s_r = 1$, the artificial bee colony optimization is not possible.

R_c: Rule for cellular transition or cellular state transformation function Primitive Artificial Bee Colony Algorithm is employed for estimating the transition probability.

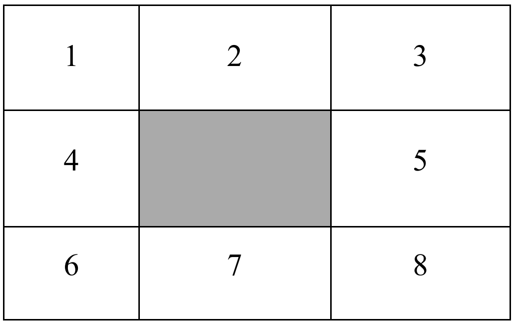


Figure 4.1a: Moore Model for CA-IABCA

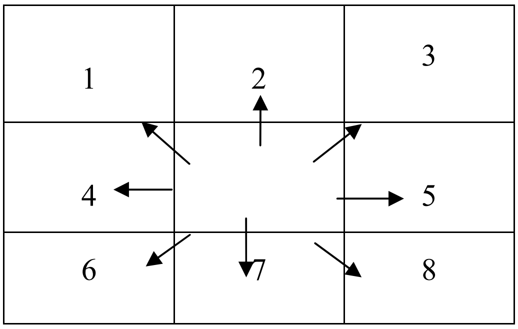


Figure 4.1b: Direction of transfer in CA-IABCA

4.1.3 CA-IABCA–Cellular Automata based Improved Artificial Bee Colony Algorithm with Grenade Explosion and Cauchy Operator

In general, a honey bee colony is efficient and effective enough in identifying the high quality food sources through natural perspectives. Therefore, CA-IABCA optimizes the problem of mitigating DDoS compromised vehicular nodes through the incorporation of concept derived from the intelligent honey bee foraging behavior. Similar to the role of three types of bees in ABCA, the proposed CA-IABCA detects DDoS attacks by accomplishing three phases such as employed bee phase, onlooker bee phase and scout bee phase. In the employed bee phase, the position of vehicular nodes in the cellular automata with its information on trustworthiness is gathered and shared. Then in the onlooker bee phase, the vehicular nodes with optimal probability are identified based on the information collected in the employed bee phase through the estimation of fitness value and further the search dimension is exploited. Finally, in the scout bee phase, the vehicular nodes which are found trustworthy are clustered near the intermediate routers. The trustworthy nodes are elected as the forwarding node when the intermediate nodes are identified as DDoS compromised. This clustering of nodes is performed for a predetermined number of iterations and threshold.

Initially, 'n' number of solution vectors that represents the set of vehicular nodes for mitigation is identified randomly. Each solution vector ON_i is a D-dimensional solution vector and $ON_i = \{ON_{i1}, ON_{i2}, ..., ON_{iD}\}$ represents the 'i^{th}' solution of the mitigation problem. The solution vectors of the problem get updated iteratively based on the search processes carried out by the employer bee phase, onlooker bee phase and scout bee phase with a Maximum Iteration Number (MIN). Each solution chooses the vehicular node for mitigation only when it satisfies the fitness function defined through equations 4.2 and 4.3.

$$fit(ON_i) = \frac{1}{1 + f(ON_i)}, f(ON_i) \geq 0 \qquad (4.2)$$

$$fit(ON_i) = 1 + abs.f(ON_i), f(ON_i), < 0 \qquad (4.3)$$

Where $f(ON_i)$ is the objective function value of solution ON_i.

Further, the probability of selecting a solution by the onlooker bee is presented by equation 4.4.

$$P_{ON(i)} = \frac{fit(ON_i)}{\sum_{n=1}^{TS} fit(ON_n)} \qquad (4.4)$$

After selecting a solution, the neighborhood nodes of the chosen vehicular nodes are explored by dynamically changing a parameter (In CA-IABCA, Reference velocity is used). Then the candidate solution of identified solution ON_{ij} is determined through equation 4.5.

$$N_{ij} = ON_{ij} + \psi_{ij}(ON_{ij} - ON_{kj}) \qquad (4.5)$$

Where 'k' is randomly chosen based on the number of identified solutions. Indices 'i' and 'j' represent the randomly chosen dimension that ranges between -1 and +1. TS and ψ_{ij} represent the total number of search solution and weight-age factor of fitness respectively. After the estimation of candidate solution, greedy selection mechanism is applied to determine whether to estimate a new candidate solution or to retain the old candidate solution.

If the optimal solution is ON_i, the scout bee phase produces a new solution from the identified optimal solution through equation 4.6.

$$ON_{ij} = ON_{\min j} + rand\,(0,1)(ON_{\max j} - ON_{\min j}) \qquad (4.6)$$

Where $ON_{\min j}$ and $ON_{\max j}$ are the minimum and maximum thresholds of the candidate solution ON_{ij} respectively.

4.1.4 Grenade Explosion based Onlooker Bees inspired Exploitation Mechanism

In ABCA, initially the onlooker bee elects an optimal vehicular node in the cellular automata model for mitigation based on the estimation of fitness value based probability P (ON_{ij}) for each node. This estimation of probability for each node (ON_{ij}) iteratively exploits the neighboring cells of cellular space L_d for determining the most optimal vehicular node at the end of each and every iteration through exploitation factor $E_{p(ij)}$. This $E_{p(ij)}$ is identified from ON_{ij} by changing a single parameter that satisfies the condition $E_{p(ij)} \neq ON_{ij}.E_{p(ij)}$. This modification in $E_{p(ij)}$ from ON_{ij} is achieved from the randomly selected position of the newly chosen expected optimal vehicular node ON_{kj}.The parameter 'j' is considered as the significant deviation parameter since it identifies the deviation between the actual position of optimal vehicular node and randomly chosen position of optimal vehicular node achieved through ABCA. Furthermore, the choice of parameter 'j' cannot always ensure ABCA in identifying high reliable positions of optimal vehicular nodes for mitigation and even they are prone to the problem of slow convergence and caught into the problem of local optimum in searching operation. This limitation of deviation parameter 'j' motivates the necessity of using Grenade Explosion Mechanism (GEM) proposed.

The GEM utilizes a class of traditional benchmark functions and few randomly generated multimodal functions for detecting the optimal regions that have the maximum probability of holding most optimal vehicular nodes of mitigation. They also possess the capacity of converging the search solution to a global minimum. Hence GEM is incorporated into the onlooker bee phase of CA-IABCA.

In this improved onlooker bee phase of CA-IABCA, GEM aids in selecting an effective search dimension rather than a random dimension chosen by the traditional ABCA. GEM helps in moving towards the identification of an optimal position of vehicular nodes that has the maximum probability of being elected as the key node for mitigating DDoS compromised nodes. After the estimation of search dimension ($SP_{(G_n)}$), the fitness of each solution is determined by generating a number of grenades (G_n) that are initialized at random locations with $SP_{(G_n)} \notin [-1,1]^n$. In CA-IABCA, grenade refers to the subdivided area of the search dimension with a specific radius of exploration. Generate a position area $X_k^{'}$ for identifying optimal vehicular nodes around the k^th^ grenade through equation 4.7.

$$X_k^{'} = \{X_k + sign(UD_{(rm)})(UD_{rm})^f L_{ge}\} \qquad (4.7)$$

Where UD_{rm} and L_{ge} refer to the uniformly distributed random number and length of influence of each grenade k=1, 2,….., G_n with a constant f= $\max\{1, n.(\frac{\log \frac{R_s}{L_{ge}}}{\log R_{\exp}})\}$.

Evaluate the distance between $X_k^{'}$ and radius of exploration $R_{\exp}$ and if $X_k^{'}$ is found at least $R_{\exp}$ distance apart from the position of grenades, $X_k^{'}$ is selected. Then compute the fitness value of the vehicular nodes within the grenade region existing around the selected $X_k^{'}$. Further reduce $R_{\exp}$ for increasing the potential of global investigation and length of grenade L_{ge} influence through equations 4.8 and 4.9.

$$R_{\exp} = \frac{R_{\exp-initial}}{(R_{usd})^{\frac{Iter-no}{Num-Iter}}} \quad (4.8)$$ and

$$L_{ge} = (L_{ge-initial})^{m}(R_{\exp})^{1-m} \quad (4.9)$$

Where, ‘m’ is iteratively varied from a higher threshold value to a lower threshold value. Hence global exploration of optimal vehicular nodes is completely achieved in the onlooker bee’s phase of CA-IABCA.

4.1.5 Cauchy operator based Scout Bees Exploration Phase

Similar to the use of GEM in onlooker bee phase, Cauchy Operator (CO) is mainly used for ensuring the possibility of the search to be executed in the global region and to prevent the search being trapped into local optimum. Further, the probability of generating random number which is deviating from the origin is greater in Cauchy than the Gaussian distribution. Cauchy distribution ensures wider search space than the Gaussian distribution employed through random numbers [128]. Thus Cauchy operator is employed in the scout bee phase of CA-IABCA using equation 4.10.

$$X_k^{'} = X_k^{'}\,Cauchy\,(0,1) \quad (4.10)$$

Where Cauchy (0,1) refers to the standard Cauchy distribution with center ‘0’ and scaling parameter 1 as defined through equation 4.11.

$$Cauchy\,(0,1) = \frac{1}{\pi(1+ON_{ij}^{2})} \quad (4.11)$$

In the next section, the flow chart (Figure 4.2) and steps involved in the implementation of CA-IABCA are portrayed.

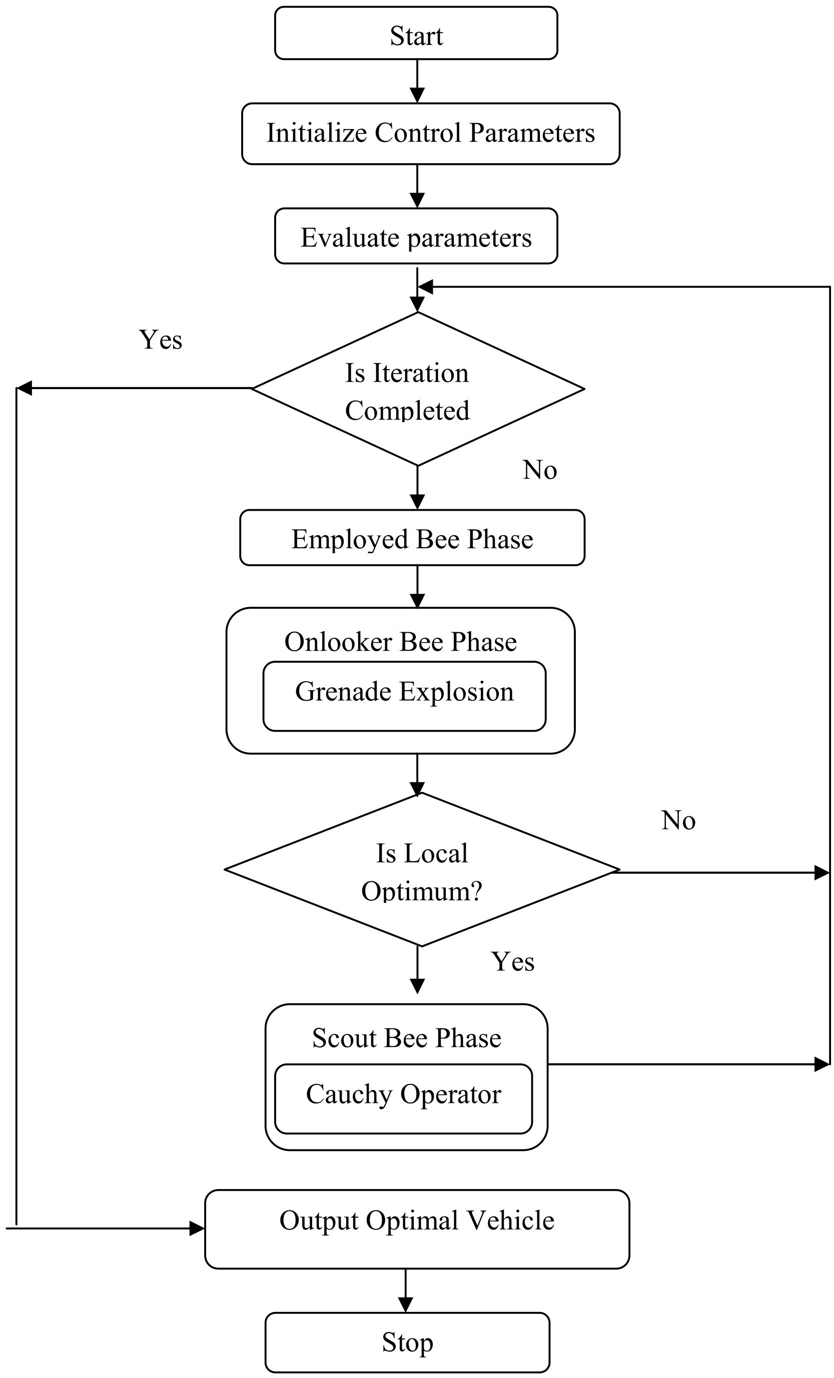


Figure 4.2: Flow chart of CA-IABCA

Algorithm 4.1: CA-IABCA (Cellular Automata based Improved Artificial Bee Colony Algorithm)

Step 1. Initialize the control parameters value: D, Threshold limit, MIN.

Step 2. Initialize the number of vehicular nodes and the position of vehicles with Fitness value probability using.

$$P_{ON(i)} = \frac{fit(ON_i)}{\sum_{n=1}^{TS} fit(ON_n)}$$

Step 3. Estimate the fitness function of each solution using.

$$fit(ON_i) = \frac{1}{1 + f(ON_i)}, f(ON_i) \geq 0$$

Step 4. Set Iteration=1.

Step 5. Repeat until packet is forwarded.

Step 6. Compute the probabilistic value of each feasible solution.

Step 7. Identify and evaluate the computed probabilistic value of each feasible solution using each iteration of onlooker bee phase in all search dimensions and determine the exploitation factor using.

$$R_{\exp} = \frac{R_{\exp-initial}}{(R_{usd})^{\frac{Iter-no}{Num-Iter}}}$$

Step 8. Identify the optimal candidate solution using $L_{ge} = (L_{ge-initial})^m (R_{\exp})^{1-m}$ and apply the process of greedy selection during the onlooker bees phase.

Step 9. If exists, identify and replace the old feasible solution using newly generated solution using $X_k^{'} = X_k^{'} Cauchy \ (0,1)$ through Cauchy operator in the scout bee phase.

Step 10. Else retain the old feasible solution as optimal solution.

Step 11. Update the best optimal solution that has been achieved in this iteration.

Step 12. iteration = iteration + 1.

Step 13. Until iteration = MIN.

4.2 CA-IABCA-Simulation Experiments and Results Analysis

The comparative performance investigation of CA-IABCA with SOA-ABCA, SOA-ACO and SOA-PSO is evaluated using ns-2 simulator. SUMO traffic simulator is used for generating vehicular mobility traces. Initially, evaluation parameters such as prediction variance (meters), prediction variance (seconds) and average prediction variance are used for investigating the performance of CA-IABCA. In this performance analysis, the experiments are carried out either by varying the number of nodes or by varying the prediction interval. The comparative analysis of CA-IABCA is based on five experiments discussed below. In the first three experiments, the performance of CA-IABCA with SOA-ABCA, SOA-ACO and SOA-PSO is analyzed through prediction variance (meters), prediction variance (seconds) and average prediction variance obtained by varying the prediction interval and vehicular nodes.

The simulation setup used for comparative performance analysis of CA-IABCA with SOA-ABCA, SOA-ACO and SOA-PSO are detailed in Table 4.1.

Table 4.1: Simulation setup for evaluating CA-IABCA

Parameters	Value
Number of vehicular nodes	100,200,300
Range of transmission	500m
Threshold speed	20-60 m/sec
Acceleration of vehicular node	1.2 m/s^2
Retardation of vehicular node	6.5 m/s^2
Simulation time	600s
Prediction interval	10-140 s
MAC protocol	IEEE 802.11p
Refresh interval time	30s

Experiment 1-Performance analysis of CA-IABCA based on prediction variance (meters) by varying prediction interval

In experiment 1, the comparative performance of CA-IABCA over the existing cellular automata based DDoS mitigation approaches like SOA-ABCA, SOA-ACO and SOA-PSO are investigated.

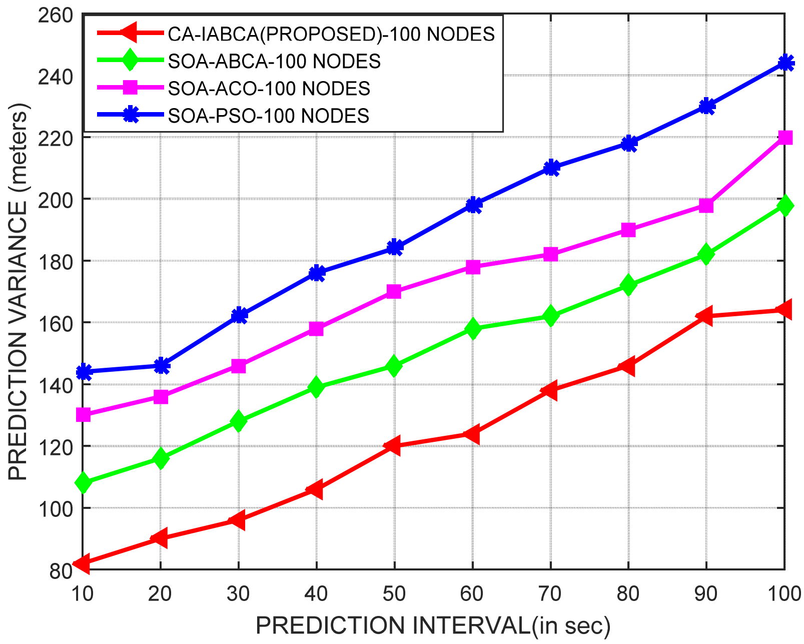


Figure 4.3 -Experiment 1-Performance of CA-IABCA-Prediction variance (meters)-100 nodes

Figure 4.3 portrays the performance of CA-IABCA in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 100 vehicular nodes. The results make it clear that the prediction variance for CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO increases phenomenally when prediction interval is increased 10 seconds for each iteration up to 100 seconds. But, CA-IABCA is able to dynamically decrease the variation especially when the prediction interval increases. CA-IABCA is found to exhibit a decrease in prediction variance to a maximum level of 10% than the compared baseline approaches.

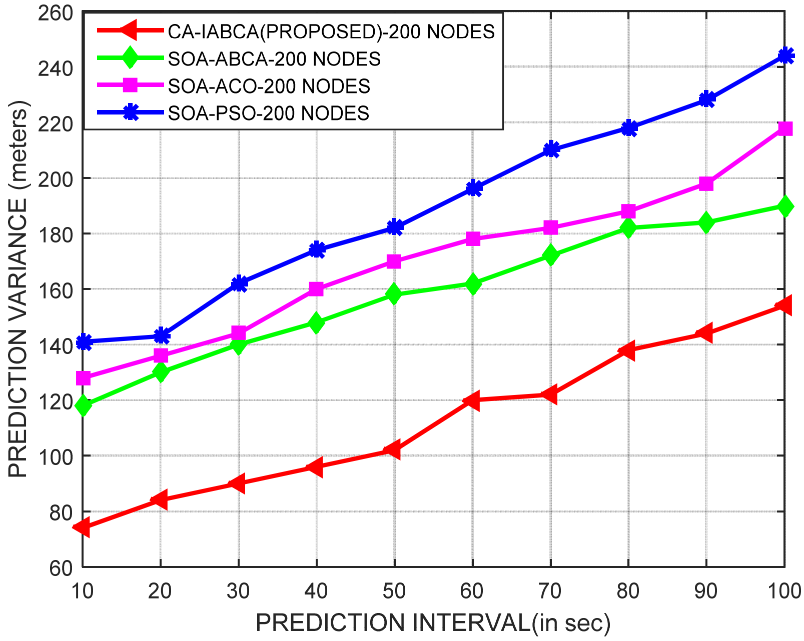


Figure 4.4 -Experiment 1-Performance of CA-IABCA -Prediction variance (meters)-200 nodes

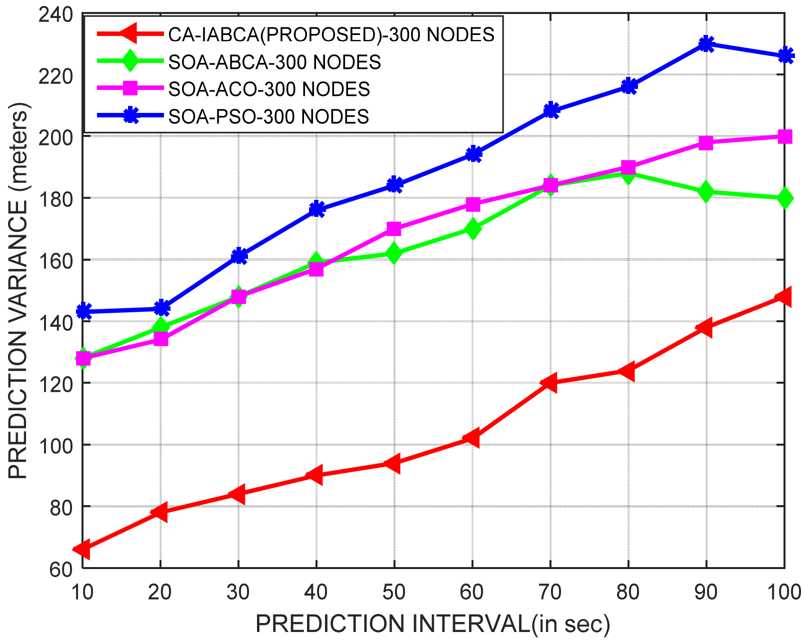


Figure 4.5 - Experiment 1-Performance of CA-IABCA-Prediction variance (meters)-300 nodes

The prediction variance for 200 nodes is depicted in Figure 4.4 measured by varying the prediction interval in each iteration. For each iteration the results show that the prediction variance for CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO increases remarkably when prediction interval is varied from 10 to 100 seconds under the influence of 200 nodes. But, CA-IABCA is able to dynamically decrease the variation in less percentage even when the number of nodes are increased by 200 especially when the prediction interval increases. CA-IABCA is found to exhibit a decrease in prediction variance to a maximum of 13% than the compared baseline approaches.

For 300 nodes the prediction variance graph is shown in Figure 4.5 portraying the performance of CA-IABCA by varying the prediction interval (seconds). The prediction variance for CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO increases considerably in each iteration when prediction interval is varied from 10 to 100 seconds under the influence of 300 vehicular nodes. The CA-IABCA gives the better performance in terms of prediction variance against the existing algorithms by having minimum increase percentage. CA-IABCA is found to exhibit a decrease in prediction variance to a maximum of 15% than the compared baseline approaches.

Experiment 2- Performance analysis of CA-IABCA based on prediction variance by varying number of nodes

In experiment 2, the performance analysis of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is carried out in terms of prediction variance per second by varying the number of vehicular nodes from 100 to 300 based on varying prediction interval.

Figure 4.6 describes the performance of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO based on prediction variance in seconds evaluated by varying the prediction interval. It is found that the prediction variance per second for CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is considerably decreased when the prediction interval is increased. But CA-IABCA is able to increase the dimension of searching and prevents it from being trapped into local minimum. Hence CA-IABCA decreases the prediction variance in seconds than SOA-ABCA, SOA-ACO and SOA-PSO to a considerable level of 13%, 16% and 19% under the influence of 100 nodes.

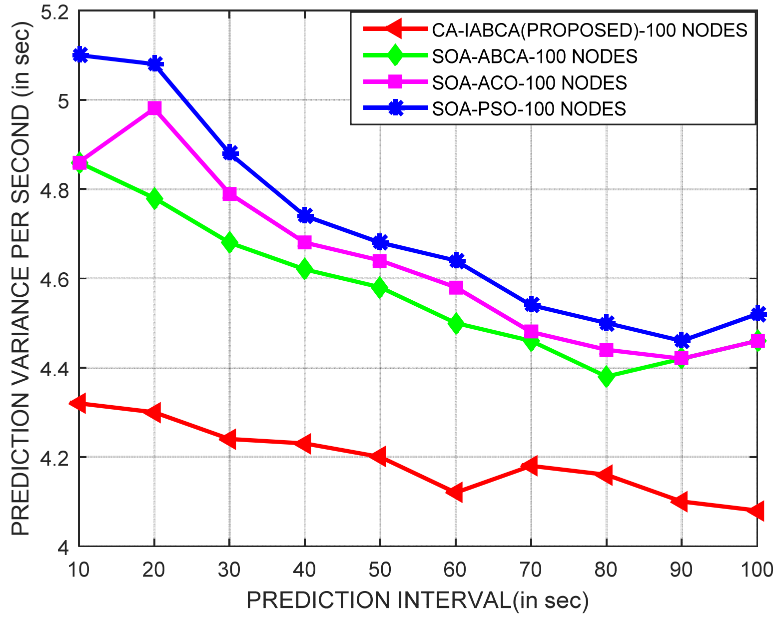


Figure 4.6 - Experiment 2-Performance of CA-IABCA-Prediction variance per second-100 nodes

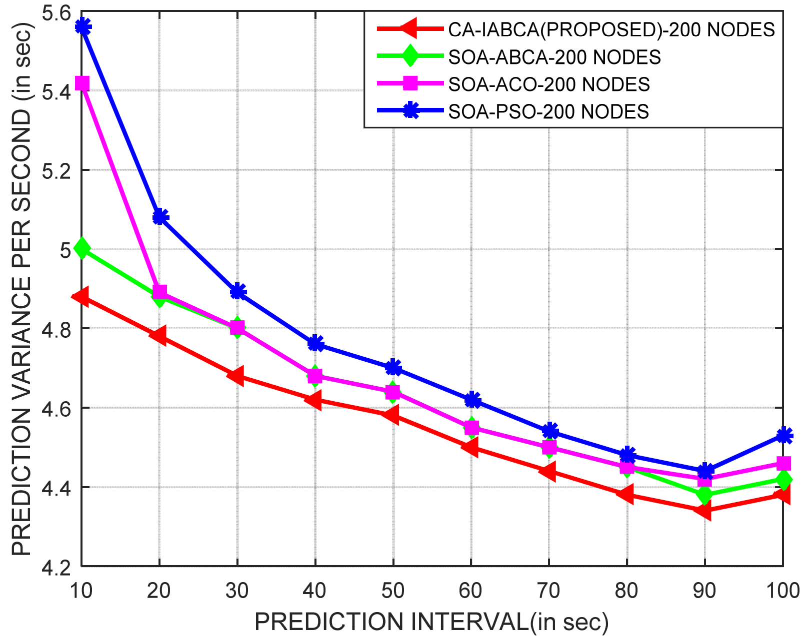


Figure 4.7 - Experiment 2-Performance of CA-IABCA-Prediction variance per second-200 nodes

Figure 4.7 describes the performance of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO based on prediction variance in seconds evaluated by varying the prediction interval with 200 nodes. It is found that the prediction variance per second for CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is decreases phenomenally decreasing not only with respect to prediction interval but also based on number of nodes. The performance of the CA-IABCA decreases in prediction variance in seconds than SOA-ABCA by 12%, SOA-ACO by 14% and SOA-PSO by 17% due to its good convergence.

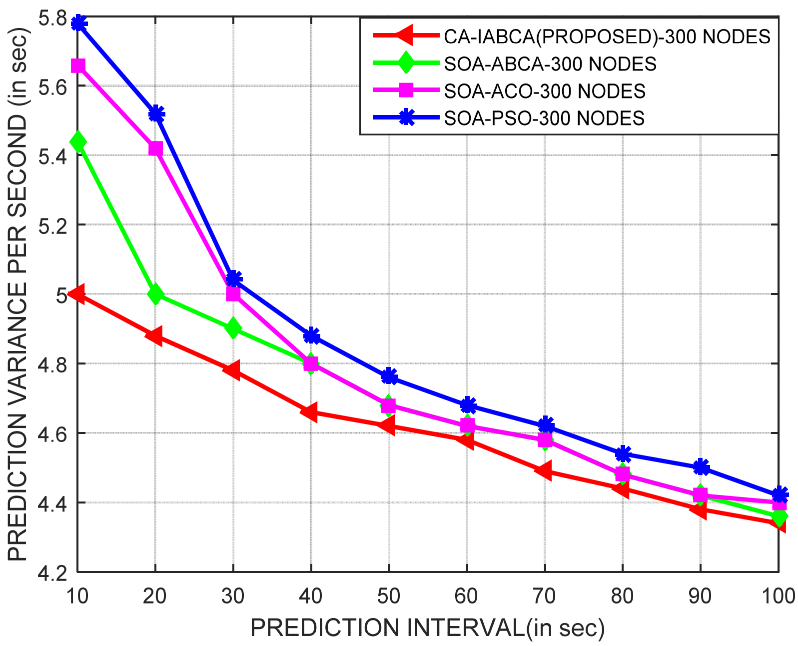


Figure 4.8 - Experiment 2-Performance of CA-IABCA-Prediction variance per second-300 nodes

Similarly, for 300 vehicular nodes the prediction variance is measured and shown in graph as in Figure 4.8 varying the prediction interval by 10s for each iteration. Each iteration shows that there is a gradual decrease in prediction variance as the interval increases. It is found that the prediction variance per second for CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is highly decreased when the number of nodes is increased to 300. But CA-IABCA decreases the prediction variance in seconds than SOA-ABCA by 8%, SOA-ACO by 10% and SOA-PSO by 13%.

Experiment 3- Performance analysis of CA-IABCA based on prediction variance by varying number of nodes

In experiment 3, the performance of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is investigated in terms of prediction variance (meters) by varying the number of vehicular nodes from 100 to 300 with fixed accuracy interval of 80s, 90s and 100s respectively.

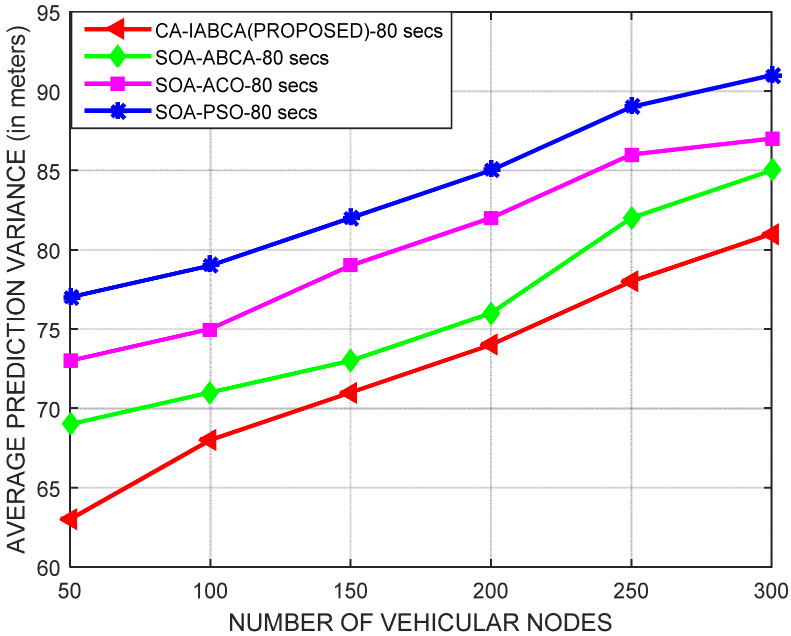


Figure 4.9 - Experiment 3-Performance of CA-IABCA-Average Prediction variance-80seconds

Figure 4.9 portrays the relationship that infers the optimal average prediction accuracy facilitated by CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO when vehicular nodes are significantly varied. This investigation is initially with an accuracy level of 80sec as it is considered as the minimum optimal value for accuracy prediction in CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO mechanism.

The average prediction variance of CA-IABCA shows a meager variation of approximately 52-57m with varying number of vehicular nodes, whereas SOA-ABCA, SOA-ABCA, SOA-ACO exhibits a deviation of 65-73m, 79-82m and 93-98m respectively.

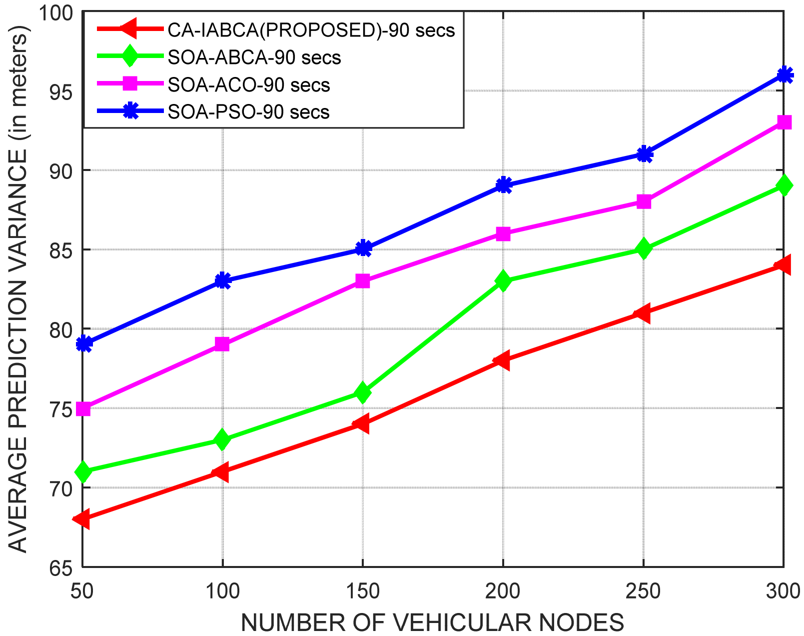


Figure 4.10 - Experiment 3-Performance of CA-IABCA-Average Prediction variance-90seconds

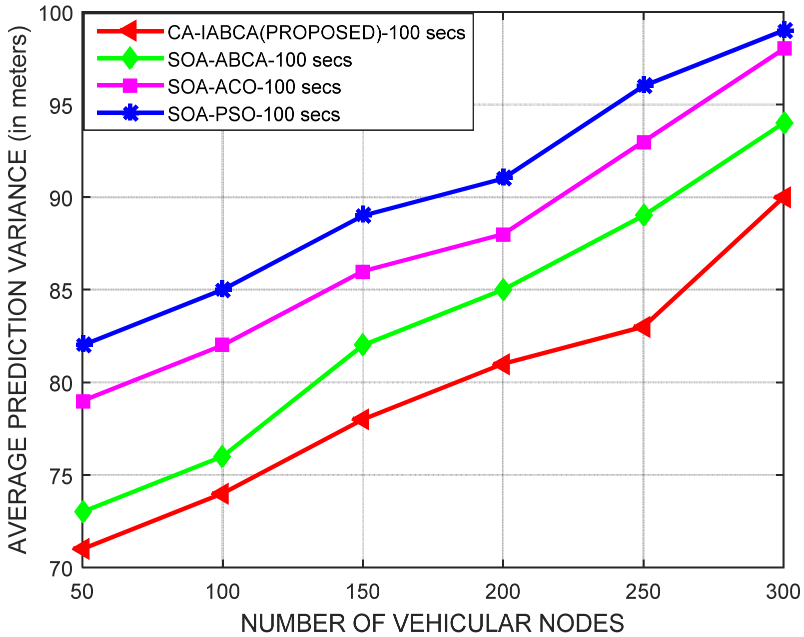


Figure 4.11 - Experiment 3-Performance of CA-IABCA--Average Prediction variance-100 seconds

Similarly, Figure 4.10 also portrays the relationship that infers the optimal average prediction accuracy facilitated by CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO when vehicular nodes are significantly varied with an accuracy level of 90 sec as it is considered as the minimum optimal value for accuracy prediction. The average prediction variance of CA-IABCA shows a meager variation of approximately 56-60m with varying number of vehicular nodes, whereas SOA-ABCA, SOA-ABCA, SOA-ACO exhibits a deviation of 68-76m, 82-87m and 98-103m respectively.

In addition, Figure 4.11 also portrays the relationship that infers the optimal average prediction accuracy facilitated by CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO when vehicular nodes are significantly varied under an accuracy level of 100sec. The average prediction variance of CA-IABCA shows a meagre variation of approximately 63-67m with varying number of vehicular nodes, whereas SOA-ABCA, SOA-ABCA, SOA-ACO exhibits a deviation of 73-78m, 86-90m and 96-104m respectively.

The result analysis of the experiment conducted with the performance metrics prediction variance (seconds), prediction variance (meters) and average prediction variance is given in the following table 4.2. The table 4.2 compares the proposed algorithm with each of the existing algorithms. Table 4.2 gives the comparison of prediction variance of proposed algorithm with other baseline approaches with the decrease in percentage.

Table 4.2 Performance comparison of Average Prediction variance (measured in meters) of proposed CA-IABCA algorithm with existing algorithms with decrease in percentage

Algorithm	**No. of vehicles**		
	100	**200**	**300**
SOA-ABCA vs CA-IABCA	19	35	50
SOA-ACO vs CA-IABCA	35	42	50
SOA-PSO vs CA-IABCA	41	48	55

For 100 vehicles the decrease in prediction variance is between 19-41 % when compared to the SOA-ABCA, SOA-ACO and SOA-PSO algorithms. For 200 vehicles the decrease in prediction variance is between 35-48 % when compared to the

SOA-ABCA, SOA-ACO and SOA-PSO algorithms. For 300 vehicles the decrease in prediction variance is between 50-55 % when compared to the SOA-ABCA, SOA-ACO and SOA-PSO algorithms. Integrating the features of Cellular Automata and improved ABC algorithm with Grenade Explosion and Cauchy Operator results in improved global search solution of finding the best neighbor to forward data.

Table 4.3 Performance comparison of Prediction variance (measured in seconds) of proposed CA-IABCA algorithm with existing algorithms with decrease in percentage

Algorithm	No. of vehicles		
	100	200	300
SOA-ABCA vs CA-IABCA	13	12	8
SOA-ACO vs CA-IABCA	16	14	10
SOA-PSO vs CA-IABCA	19	17	13

Table 4.3 gives the comparison of delays incurred by proposed algorithm with other baseline approaches with the decrease in percentage. For 100 vehicles the decrease in prediction variance is between 13-19 % when compared to the SOA-ABCA, SOA-ACO and SOA-PSO algorithms. For 200 vehicles the decrease in prediction variance is between 12-14 % when compared to the SOA-ABCA, SOA-ACO and SOA-PSO algorithms. For 300 vehicles the decrease in prediction variance is between 8-13 % when compared to the SOA-ABCA, SOA-ACO and SOA-PSO algorithms. From the analysis it is revealed that CA-IABCA decreases in prediction delay when compared to other existing algorithms.

Table 4.4 Performance comparison of Average Prediction variance (measured in meters) of proposed CA-IABCA algorithm with existing algorithms with decrease in meters

Algorithm	Accuracy interval level		
	80s	90s	100s
SOA-ABCA	65-73	68-76	73-78
SOA-ACO	79-82	82-87	86-90
SOA-PSO	93-98	98-103	96-104
CA-IABCA (Proposed Work)	**52-57**	**56-60**	**63-67**

Table 4.4 gives the comparison of average prediction variance of proposed algorithm with other baseline approaches. The experiment conducted by varying the number of nodes from 50 to 300 and measuring the average prediction variance for each group. The result from the Table 4.4 reveals that for prediction interval of 80s the proposed algorithm has average prediction variance from 52-57m which is less when compared to all other existing algorithms. Similarly, the average prediction variance is measured by varying the number of vehicles with different prediction intervals of 90s and 100s which resulted in average prediction variance of 56-60m and 63-67m respectively and is found that the proposed algorithm has less average prediction variance among all existing algorithms.

4. 3 Modified Artificial Bee Colony Algorithm using Differential Evolution

In the virtue of nature, the honey bee colony is potential enough in determining the optimal food sources from the feasible number of food sources available to them. This potential of honey bee colonies foraging behavior is made used in Artificial Bee Colony Algorithm (ABCA) for optimizing any real time situations. It searches and estimates optimal search solution from the existing list of feasible candidate solutions [129]. Modified Artificial Bee Colony Algorithm (MABCA) is an enhanced version of ABCA in which two significant modifications are carried out on the initial population estimation step and adaptable search strategies implementation phase. MABCA is used for identifying an optimal vehicular node for mitigation when a specific vehicular node under data dissemination is found to be DDoS compromised node. Similar to ABCA, MABCA uses three kinds of bee phases, employer bee phase, onlooker bee phase and scout bee phase for establishing optimal candidate solutions in the event of DDoS attacks of vehicular node. In the employer bee phase, location, reputation, relative velocity and mobility rate of vehicular nodes are collected and updated from the two-dimensional cellular automata. After data gathering, a series of evolutionary operations such as mutation, crossover and selection are employed with two different ABCA differential evolution-based mutation strategies. These strategies are incorporated in the mutation step of onlooker bee phase for computing the exploitation probability. Then, an effective initialization scheme that integrates opposition-based learning and chaotic systems are used in the scout bee phase. The initialization scheme is used for discovering the optimal vehicular node to play the role of a forwarding node until a predetermined number of iterations and threshold when the intermediate node is identified as DDoS compromised.

In MABCA, the number of cells in two dimensional cellular grid constitutes the number of employer bee agents and thus there is only one employer bee agent for each two dimensional cellular automata. The number of vehicular nodes in each cell represents the candidate solution of CA-MABCA problem and choosing an optimal node from the available number of candidate solutions is computed through the fitness function. In CA-MABCA, the onlooker bee agent identifies a candidate solution of optimal vehicular node based on the probabilistic value 'PV_i'related to the cells of the cellular automata using equation 4.12.

$$PV_i = \frac{Fitness_i}{\sum_{i=1}^{ON} Fitness_i} \qquad (4.12)$$

Where '$Fitness_i$' and 'ON_i' represents the fitness function of each candidate solution and optimal nodes that could be elected for mitigating DDoS compromised nodes. To determine the candidate solution $CS_i = \{CS_{i1}, CS_{i2}, ..., CS_{iD}\}$ from the old candidate solution $OCS_i = \{OCS_{i1}, OCS_{i2},...., OCS_{iD}\}$ of each iteration from the D-dimensional solution vector using equation 4.13.

$$CS_{ij} = OCS_{ij} + \varpi_{ij}(OCS_{ij} - OCS_{kj}) \qquad (4.13)$$

Where 'k', 'i' and 'j' are randomly chosen based on the number of quantified candidate solution and randomly chosen dimension that lies in the range of -1 and +1. CA-MABCA uses the technique of greedy selection to decide whether new candidate solution has been computed or the existing candidate solution has to be retained.

If the existing candidate solution is retained, the scout bee agent produces a new optimal solution from the computed optimal candidate solution using equation 4.14.

$$OCS_{ij} = OCS_{\min j} + rand(0,1)(OCS_{\max j} - OCS_{\min j}) \quad (4.14)$$

Where $OCS_{\min\ j}$ and $OCS_{\max\ j}$ are the minimum and maximum thresholds of the candidate solutions. The candidate solution vectors get iteratively updated through the incorporation of employee bee phase, onlooker bee phase and scout bee phase agent within the Lowest Minimum Threshold Time (LMIN_THRESHOLD) [130]. The candidate solution is said to be a feasible solution under election only when it satisfies the fitness function given by equation 4.15 and 4.16.

$$Fitness(ON) = \frac{1}{1 + sf(ON)}, sf(ON) \geq 0 \qquad (4.15)$$

$$Fitness(ON) = 1 + abs.sf(ON), sf(ON), \geq 0 \qquad (4.16)$$

Where *s f(ON)* is the objective function value of solution 'ON'.

4.3.1 2D-Space cellular model configuration for CA-MABCA

The cellular automata model space used in CA-IABCA is facilitated with four points such as source vehicle point, destination vehicle point, intermediate router node points and free space that follows Moore model. The distance between the source vehicle point and destination vehicle point is considered as 'L' and the intermediate router nodes can move around a width of 'L/2'.Thus the space 'S' is given by the equation 4.17.

$$S = \{(x, y) / x \in \{0,..., X_{\max}\}, y \in \{0,...., Y_{\max}\} \qquad (4.17)$$

The position of each vehicle pertains to a point (x, y), when S(x, y) =1, the collision of vehicles is possible and S(x,y) =0 represents the collision free space of the cellular automata.

4.3.2 Evolution rules employed for CA-MABCA

The cellular automata model employed in CA-MABCA consists of cells and cellular spaces for discretizing time and space in the search space. The time and space are discretized for analyzing and describing the dynamic behavior of vehicular nodes in the 2D space as shown in Figure 4.12a and Figure 4.12b. Each and every cell employed in the lattice grid space exhibits finite number of discrete states and the behavioral states are updated based on the newly innovated local space rule. This complex dynamically changing complex behavioral state process identification is modeled into a discrete interactive process.

The Cellular Automata Moore model used in CA-MABCA is represented using a 4-tuple $C_A = (L_d, S_c, N_c, R_c)$, where,

C_A: Moore model based cellular automata.

L_d : Cellular space with 'd' positive dimension used in the Moore model (d=2 in CA-MABCA).

S_c: Possible state space of cellular automata (0, 1).

N_c: Neighbors of each individual cellular cell defined based on Moore model represented through S=$\{n_1, n_2, ...n_r, ...n_n\}$.where 'S' and 'r' denote the spatial vector that incorporates 'n' feasible cellular states and direction of the artificial bee colony respectively. In CA-MABCA, $r \in [1,8]$ and $s_r \in [0,1]$ which infers that the transition will decide to choose any direction, else if $s_r = 1$, the artificial bee colony optimization is not possible.

R_c: Rule for cellular transition or cellular state transformation function Primitive Artificial Bee Colony Algorithm is employed for estimating the transition probability.

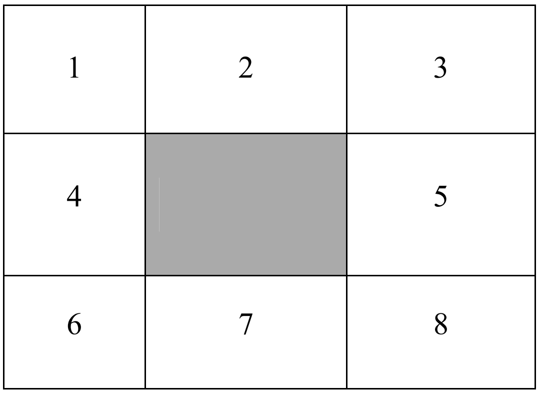


Figure 4.12a : Moore Model for CA-MABCA

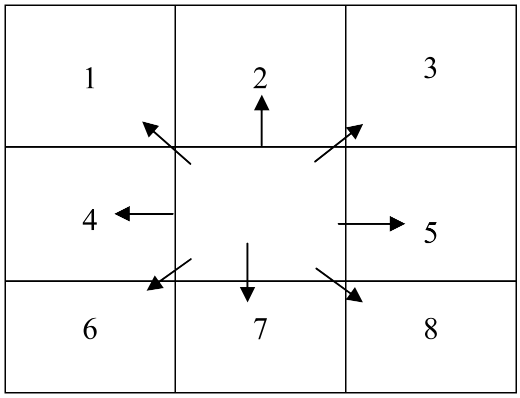


Figure 4.12b : Direction of transfer in CA-MABCA

4.3.3 Differential Evolution based Onlooker Bees inspired Exploitation Mechanism

In CA-MABCA, a series of operation such as mutation, crossover and selection are performed in the Onlooker Bee Exploitation Bee phase. For accomplishing this bee phase, Differential Evolution (DE) techniques are found to be highly applicable and suitable. In this approach, two variants of DE [131] used in this CA-MABCA are given by equations 4.18 and 4.19.

$$CS_{ij} = OCS_{best,j} + \varpi_{ij}(OCS_{n1,j} - OCS_{n2,j}) \quad (4.18) \text{ and}$$

$$CS_{ij} = OCS_{best,j} + \varpi_{ij}(OCS_{n1,j} - OCS_{n2,j}) + \varpi_{ij}(OCS_{n3,j} - OCS_{n4,j}) \quad (4.19)$$

Where 'n1','n2', 'n3' and 'n4' are randomly chosen mutual exclusive integers that range from 1,2,…,ON and is entirely distinct from the base indices 'i '. 'OCS_{best}' presents the best optimal candidate solution determined from the feasible candidate solutions of the search domain and ' $\varpi_{i,j}$ ' is the randomly generated integers that ranges from -1 to +1. It is also inferred that the improvised search solution represented through equation 4.18 and 4.19 is capable for enhancing the exploitation degree of CA-MABCA to a significant level.

4.3.4 Integrated Chaotic and opposition-based learning inspired Scout Bees Exploration Phase

In the Scout bee phase, population initialization plays a significant role as they influence the quality and convergence rate of the final optimal solution that is being identified. Random initialization is used mostly in this phase if the information about population initialization referred to as candidate solutions is not completely available. In CA-MABCA, Integrated Chaotic and Opposition-based Learning inspired Scout Bees Exploration Phase is incorporated as it possesses the properties of irregularity, ergodicity and randomity for generating candidate solutions in addition to the features of random initialization. This Integrated Chaotic and opposition-based learning process uses sinusoidal iteration as expressed in equation 4.20.

$$CS(I)_{k+1} = \sin(\pi CS(I)_k), CS(I)_k \in (0,1) \qquad (4.20)$$

Where 'k' and 'K' represent the iteration counter and maximum count of preset chaotic iterations (K=250 for CA-MABCA). Further the optimal solutions from the existing candidate solutions can be rapidly achieved through the use of a parameter called Population Diversity (PD) quantified using equation 4.21.

$$PD = \frac{1}{ON}\sum_{I=1}^{ON}\sqrt{\frac{1}{D}\sum_{j=1}^{D}(OCS_{i.j} - OCS_j)} \qquad (4.21)$$

Thus the degree of exploration is assured in the scout bee phase of CA-MABCA. The algorithms step that form the foundation of implementing CA-MABCA are detailed Figure 4.13.

Algorithm 4.2: MABCA (Modified Artificial Bee Colony Algorithm)

Step 1: *Initialize the Vehicles in 2D Moore Model CA along with its reference velocity, reliability factor.*

Step 2: *Compute the fitness function based on reliability and reference mobility.*

Step 3: *Loop 1.*

Step 4: *Select the initial solution using the probability function.*

$$PV_i = \frac{Fitness_i}{\sum_{i=1}^{ON} Fitness_i}$$

Step 5: *Apply two variants of Differential Evolution to select the candidate solution in Onlooker bee phase.*

$$CS_{ij} = OCS_{best,j} + \varpi_{ij}(OCS_{n1,j} - OCS_{n2,j})$$

$$CS_{ij} = OCS_{best,j} + \varpi_{ij}(OCS_{n1,j} - OCS_{n2,j}) + \varpi_{ij}(OCS_{n3,j} - OCS_{n4,j})$$

Step 6: *Apply chaotic and opposition-based machine learning to replace the old solution if the new one exists in scout bee phase by.*

$$CS(I)_{k+1} = \sin(\pi CS(I)_k), CS(I)_k \in (0,1)$$

***Step** 7*: *Update the solutions for each round.*

Step 8: *End.*

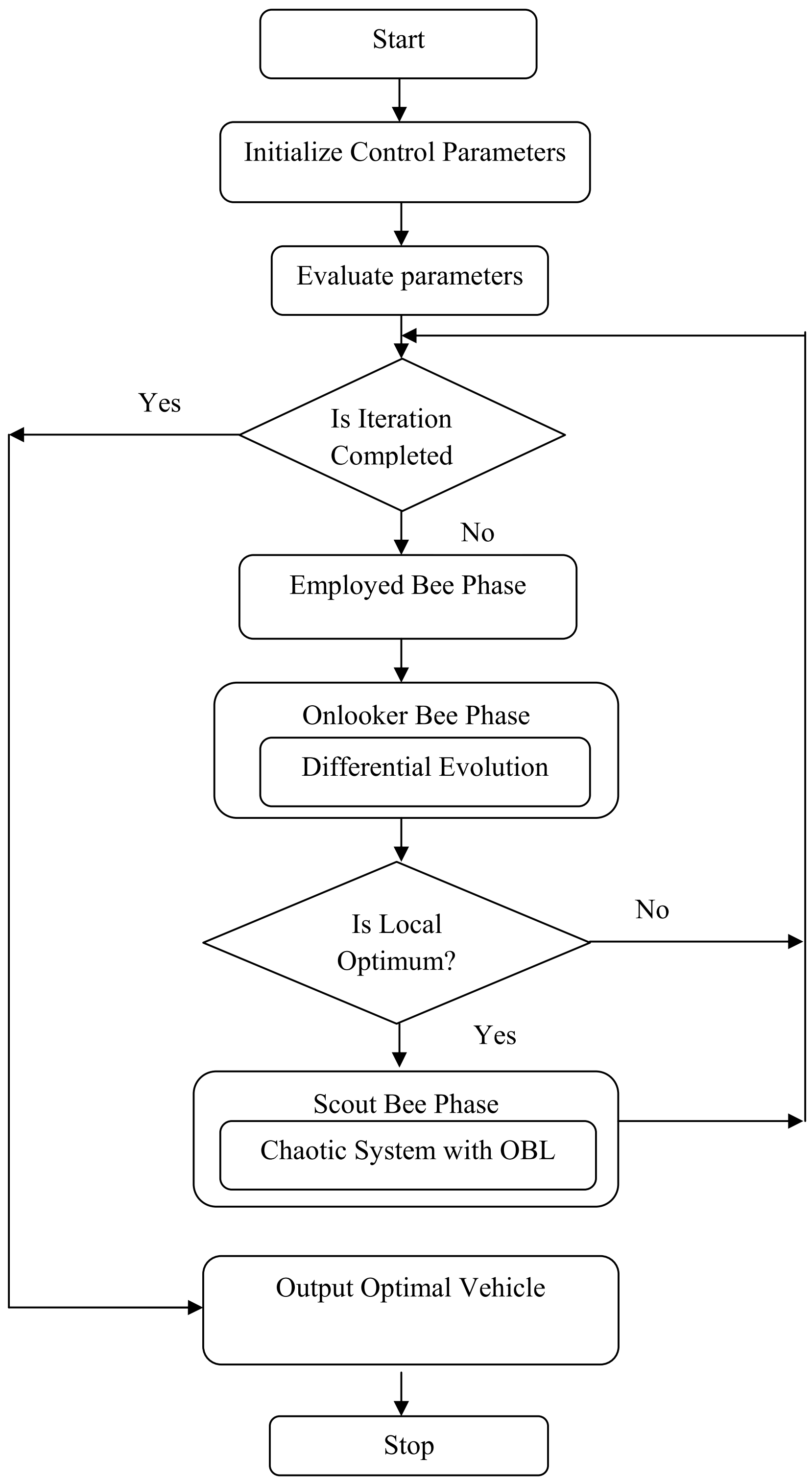


Figure 4.13: Flow Chart of MABCA

4.4 MABCA-Simulation Experiments and Results Analysis

The performance of CA-MABCA is analyzed with CA-ABCA and CA-ACO through ns-simulator that incorporates the SUMO traffic simulator for initiating vehicular traces. Two performance metrics namely prediction variance (meters) and mean prediction variance are utilized for investigating the performance of CA-MABCA. This performance investigation is achieved through three experiments in which the first two experiments are performed based on prediction variance (meters) and mean prediction variance. The performance is analyzed by varying the prediction interval and number of vehicular nodes respectively.

The simulation setup used for analyzing the potentiality of CA-MABCA over CA-ABCA and CA-ACO is portrayed in Table 4.5.

Table4.5: Simulation setup for CA-MABCA

Parameters	Value
Number of vehicular nodes	100,200
Range of transmission	400m
Threshold speed	30-50 m/sec
Acceleration of vehicular node	1.4 m/s^2
Retardation of vehicular node	6.2 m/s^2
Simulation time	400s
Prediction interval	10-120 s
MAC protocol	IEEE 802.11p
Refresh interval time	20s

Experiment 1-Performance evaluation CA-MABCA based on prediction variance (meters)

In experiment 1, performance of CA-MABCA is compared with CA-ABCA and CA-ACO based DDoS mitigation schemes by varying the prediction interval (in sec) with respect to prediction variance (meters).

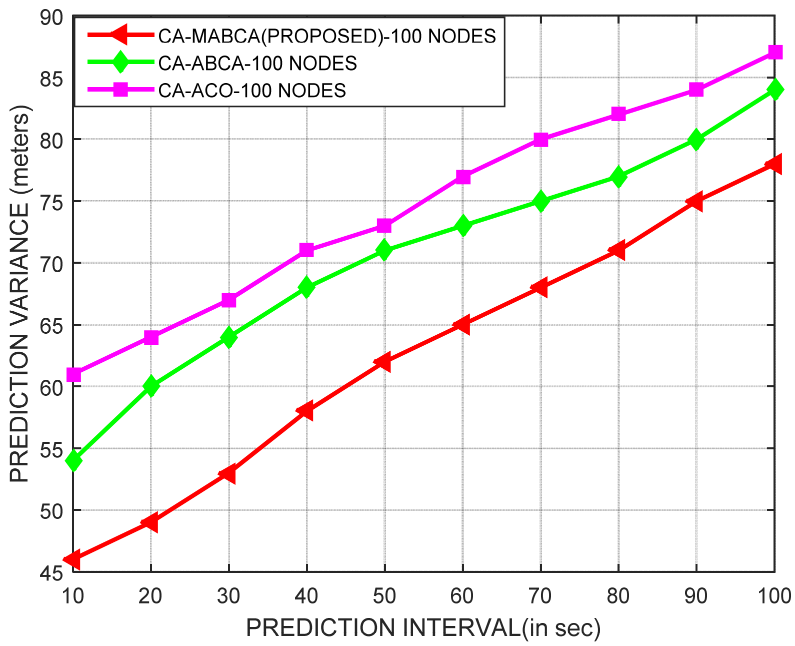


Figure 4.14 -Experiment 1-Performance of CA-MABCA-Prediction variance (meters)-100 nodes

Figure 4.14 highlights the performance of CA-MABCA evaluated based on prediction variance (meters) by varying the prediction interval (seconds) in steps with number of vehicular nodes set to 50. The results confirm that the prediction variance of CA-MABCA, CA-ABCA and CA-ACO seem to get systematically increased when there is a corresponding increase in the prediction interval. CA-MABCA is found to potentially sustain the deviation even when there is a proportional increase in the prediction interval. CA-MABCA is found to be potent in maintaining the decrease in prediction variance of about 16% than the baseline approaches used for investigation.

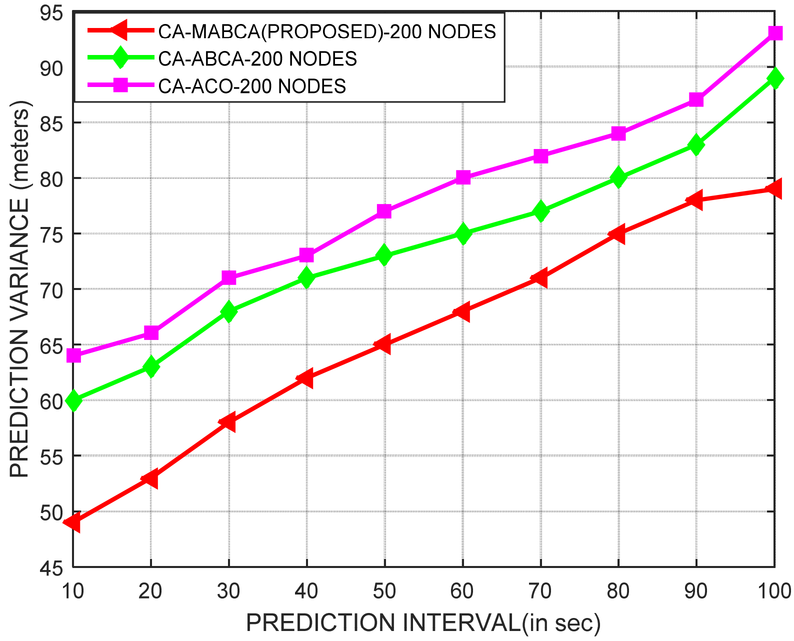


Figure 4.15 -Experiment 1-Performance of CA-MABCA-Prediction variance (meters)-200 nodes

Figure 4.15 depicts the performance of CA-MABCA evaluated in terms of prediction variance (meters) with corresponding variation in the prediction interval (seconds) under the impact of 100 vehicular nodes. Results confirm that the prediction variance for CA-MABCA, CA-ABCA and CA-ACO is found to systematically increase with increase in the prediction interval under the impact of 100 nodes. CA-MABCA is found to be potent in dynamically reducing the degree of variation and thus minimizes prediction variance to a maximum extent of 12% than the baseline approaches.

Experiment 2- Performance evaluation CA-MABCA based on mean prediction variance (meters)

In experiment 2, the performance of CA-MABCA is analyzed with CA-ABCA and CA-ACO based on mean prediction variance (meters) with accuracy interval of 90s and 100s respectively.

Figure 4.16 represents the relation between Mean prediction accuracy of CA-MABCA, CA-ABCA and CA-ACO achieved by varying the number of vehicular nodes from 10 to 80. This analysis is initially performed with an accuracy level set to 90sec as it is considered as the lower threshold value for accuracy prediction in CA-MABCA, CA-ABCA and CA-ACO. CA-MABCA exhibits only a slight variation of approximately 48-53m, but CA-ABCA and CA-ACO exhibits a deviation of 56-63m and 65-73 m respectively.

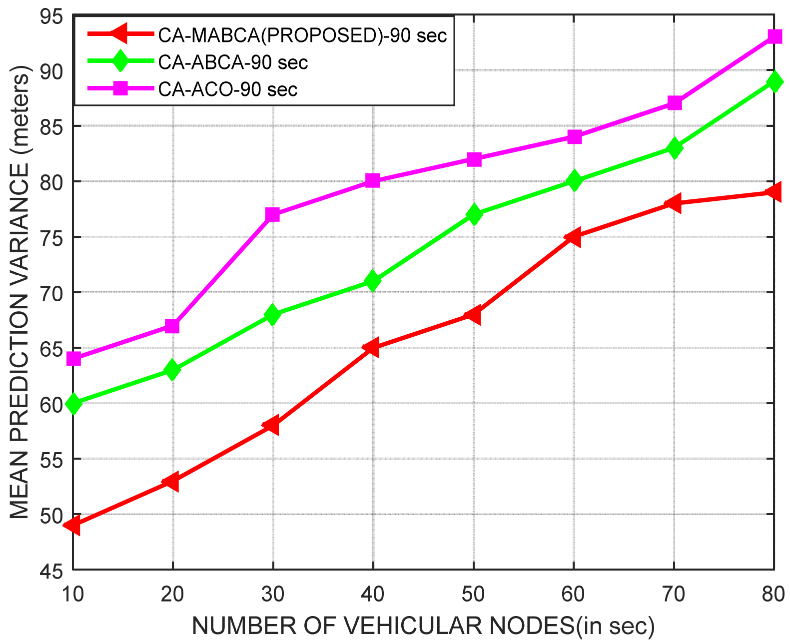


Figure 4.16 - Experiment 2-Performance of CA-MABCA-Mean Prediction variance-90seconds

Similarly, Figure 4.17 depicts the relation between Mean prediction accuracy of CA-MABCA, CA-ABCA and CA-ACO under an accuracy level of 100sec since it is identified as upper threshold value for accuracy prediction. The mean prediction variance of CA-MABCA confirms a deviation of about 51-56, but CA-ABCA and CA-ACO exhibit a deviation of 61-65m and 69-75 m respectively.

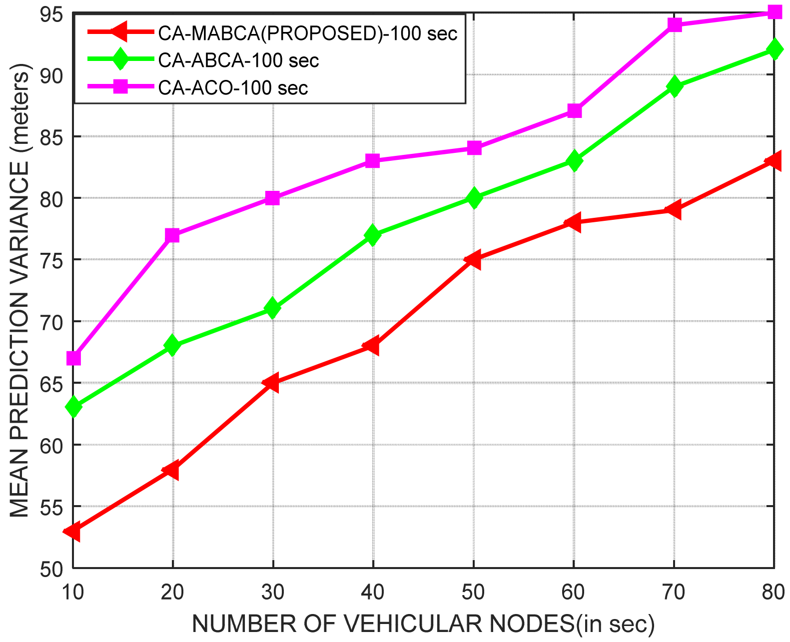


Figure 4.17 - Experiment 2-Performance of CA-MABCA-Mean Prediction variance-100seconds

The result analysis of the experiment conducted for the performance metrics prediction variance (seconds), prediction variance (meters) and average prediction variance is given in the following table. The table compares the proposed algorithm with each of the existing algorithms.

Table 4.6 Performance comparison of Average Prediction variance (measured in meters) of proposed MABCA algorithm with existing algorithms with decrease in percentage

Algorithm	No. of Vehicles	
	100	200
CA-ACO vs CA-MABCA	11	23
CA-ABCA vs CA-MABCA	15	18

Table 4.6 gives the comparison of prediction variance of proposed algorithm with other baseline approaches with the decrease in percentage. For 100 vehicles the decrease

in prediction variance is between 11-15 % when compared to the CA-ACO and CA-ABCA algorithms. For 200 vehicles the decrease in prediction variance is between 23-18 % when compared to the CA-ACO and CA-ABCA algorithms. The ABC algorithm with improved exploration and exploitation level using DE integrated with CA, results in better prediction of neighboring nodes.

Table 4.7 Performance comparison of Average Prediction variance (measured in meters) of proposed MABCA algorithm with existing algorithms with decrease in meters

Algorithm	Prediction Interval	
	90s	100s
CA-ACO	64-94	67-95
CA-ABCA	60-89	63-93
MABCA (Proposed Work)	**49-79**	**53-84**

Table 4.7 gives the comparison of average prediction variance of proposed algorithm with other baseline approaches. The experiment was conducted by varying the number of nodes from 10 to 80 and measuring the average prediction variance for each group. The result from the table4.7 reveals that for prediction interval of 90s the proposed algorithm has average prediction variance from 49-79m which is less when compared to all other existing algorithms. Similarly, the average prediction variance is measured by varying the number of vehicles with different prediction intervals of 100s which resulted in average prediction variance of 53-84 and it was found that the proposed algorithm has less average prediction variance among all existing algorithms.

4.5 Summary

In this chapter CA-IABCA is proposed and investigated for resolving the issue of mitigating DDoS attacks as they are potential in exploiting and exploring the search dimension for facilitating reliable packet delivery in VANETs. CA-IABCA uses the benefits of grenade explosion in the onlooker bee phase of ABCA for fine tuning the exploitation level of search and it uses Cauchy operator in the scout bee phase for selecting the optimal vehicular node for optimal replacement of DDoS compromised nodes. CA-IABCA ensures a genuine and precise search dimension that improves the extent of mitigation of attackers of the network to a maximum level. It is also identified that the precision in mitigation facilitated by CA-IABCA considerably increases as the size of exploitation enabled by grenade explosion mechanism increases gradually. The comparative investigation through simulation infers that CA-IABCA is capable of reducing the prediction variance and delay. The prediction variance is reduced to a maximum extent of 15% and delay to 18% even when the number of vehicular nodes is increased.

The same algorithm is modified by replacing the GEM with DE and Cauchy operator by chaotic system with Opposition Based Learning and the result showed that the algorithm fine-tuned the level of exploration to defend against DDoS attacks in VANETs. The simulation results of CA-MABCA confirm a predominant reduction in prediction variance and mean prediction variance of about 16% and 18% respectively.

CHAPTER 5

CELLULAR AUTOMTA INSPIRED MODIFIED TABU SEARCH ALGORITHM

The integration of Cellular Automata with any of the meta-heuristic algorithms results in determining the best optimal state of the node under reliable communication [132-134]. The idea of integrating the meta-heuristics algorithm with cellular automata can be applied to handle DDoS attack in VANETs by improving global search ability. The update rule of each and every cell depends on its current state and the present status of the neighbor estimated based on the status of the optimal cell. The update rules are based on re-estimation techniques that possess the capability of determining best neighbors of the network topology. The updating rule can be improved by applying Tabu Search algorithm.

In the proposed work, a Cellular Automata Inspired Modified Tabu Search Algorithm (CA-MTSA) is an attempt to integrate Cellular automata with the benefits of Meta-heuristic algorithms like the baseline Tabu Search. This Tabu Search Algorithm utilizes the benefits of an improved Tabu-list that updates the rules for identifying vehicular nodes based on five vital parameters viz., past velocity, past reliability factor, local optimal state, global optimal state and neighbor’s best state.

5.1 CELLULAR AUTOMTA INSPIRED MODIFIED TABU SEARCH ALGORITHM (CA-MTSA)

An improved Tabu search algorithm based on Cellular Automata is proposed for identifying the availability of vehicular nodes for ensuring connectivity under DDoS attack. This approach integrates the information pertaining to the designed cells, cellular spaces and cellular neighbors of each traffic model achieved through cellular automata by incorporating an improved Tabu-list. In CA-MTSA, the Tabu search algorithm is mainly combined with CA for enriching the ability of local and global search. In Tabu search algorithm, the current connectivity updates of each cell in the traffic model is updated to their neighboring cells. Each cell is closely associated in CA with some certain framed rules. Based on the framed rules, current information pertaining to the node availability is made known to the neighboring nodes. In the proposed work, the quantification for the availability of each cell depends not only on the current state of the cell but also on the state information of the neighboring nodes. In CA-MTSA, the improved Tabu-list is considered highly efficient as it bases the decision of mitigating DDoS attacks. It gathers five factors that relate to the availability quantification of vehicular nodes under communication since the traditional cellular automata based mitigation schemes utilize only three parameters like previous velocity, position, and mobility speed. The two-dimensional cellular grid structure used in the proposed work is given in Figure 5.1.

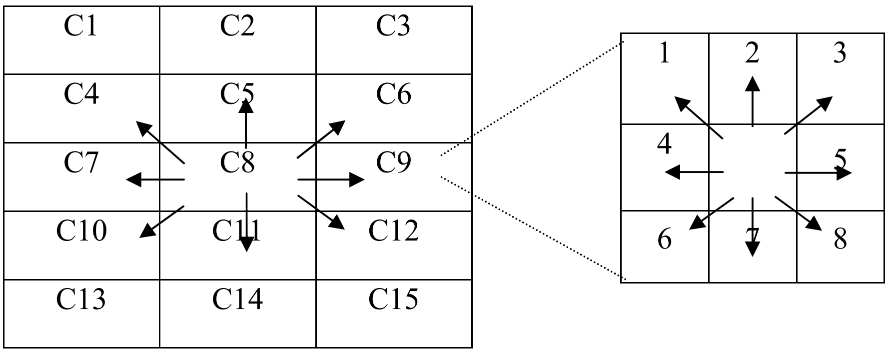


Figure 5.1: CA model for CA-MTSA

In this cellular automata inspired continuous co-ordinate space, the search is considered to be 'M' dimensional with an initial number of 'N' nodes. The availability of each node 'i' depends on its past velocity, past reliability factor, local optimal state, global optimal state and neighbor's best state. First, the past position and velocity of each node in time 't' is given by $P_{p(i)} = (P_{p(i1)}, P_{p(i2)}, P_{p(i3)}, \dots, P_{p(iM)})^T$ and $P_{v(i)} = (P_{v(i1)}, P_{v(i2)}, P_{v(i3)}, \dots, P_{v(iM)})^T$ respectively. The fitness value of each node in time 't+1' based on positional and velocity is achieved through fitness function fit (node (i)) =fit (node (($P_{p\,(i)}$), ($P_{v\,(i)}$)). The past velocity of each node of a constituent cell is updated using reference mobility rate 'r' that scales between 0 and 1.

Reference mobility rate refers to the relative velocity of each node defined in terms of its change in position varying with respect to time when compared to their neighboring nodes represented using equation 5.1.

$$r = \frac{(1 + D_{vel(o-e)}) + (r_{pre}(N_{HM} - 1))}{N_{HM}} \qquad (5.1)$$

Where, $D_{vel(o-e)}$, r_{pre}, N_{HM} depicts the difference between the observed and expected relative velocity, past reference mobility rate and number of hello packets used for updating this essential reference mobility rate respectively. The reference mobility rate is maximum when the velocity of the router node is nearly equal to its neighboring node within a cell. Each and every node of the cellular automation stores and sends the updates to their neighboring nodes through hello messages. The node with minimum reference mobility rate is considered for choosing the optimal node of the grid. Reference mobility rate ‘r’ can be used for both identifying the local optimal node within a cell and global optimal node within the entire grid. The value of ‘r’ is reset to zero at each time at refresh interval to accurately identify and reflect the availability of the nearest node for rapid forwarding of data packets.

Secondly, past reliability factor of each node present in each constituent cell is updated based on the computation of trust factor that follows Gwet’s kappa. This Gwet’s kappa based trust or reliability factor quantifies the degree to which a vehicular node may rely upon its neighboring nodes for packet forwarding. This reliability factor depends on two important parameters namely ‘p’ and e(β) that portrays the overall percent agreement and chance agreement probability of vehicular nodes classified based on two behaviours viz., co-operative and DDoS compromised. Thus the quantified Gwet’s kappa reliability factor is computation through using equation 5.2.

$$R_i(t) = \frac{p - e(\beta)}{1 - e(\beta)} \qquad (5.2)$$

Where, $p = \frac{c+d}{n}$ and $e(\beta) = 2q(1-q)$ in which ‘c’ and ‘d’ refers to the number of nodes identified as co-operative and DDoS compromised with chance agreement ‘q’ from the entire 'n' nodes of the grid. The chance agreement refers to the average between the number of times a node is rated as reliable or unreliable.

Thirdly, the local optimal state (los) of each node belonging to each corresponding cell follows conditional probability that gets updated based on the Bayes probability. Given the number of nodes present in a cell, the host nodes of the cell can be classified into co-operative and DDoS compromised based on P (C_c) and P (C_d) through prior knowledge. The Bayes theorem based conditional probability that quantifies the local neighbor search for identifying the reliable path between the source and destination vehicle of the network can be represented using P (T_x /C_c) and P (T_x /C_d) respectively. Thus the Bayes theorem based conditional probability that identifies a vehicular router node T_x of a cell as co-operative ($A_i(los)$) and DDoS compromised is given by equations 5.3, 5.4 and 5.5.

$$P(C_C/T_x) = \frac{P(C_c)P(T_x/C_c)}{P(C_c)P(T_x/C_c)+P(C_d)P(T_x/C_d)} \qquad (5.3)$$ and

$$P(C_C/T_x) = \frac{P(C_d)P(T_x/C_d)}{P(C_c)P(T_x/C_c)+P(C_d)P(T_x/C_d)} \qquad (5.4)$$

Where $P(C_c) + P(C_d) = 1$ (5.5)

Similarly, the global optimal state (gos) of each node belonging to the entire grid also follows conditional probability that can be updated by computing Dempster-Shafer Evidence Probability. This Dempster-Shafer Evidence Probability is a general form of Bayes probability that is highly suitable for identifying the global best node of the considered dimensional grid. This probability computed for estimating the global optimal node depends on three important assignments that relate to the probability of total belief mass, belief and plausibility. In this context, if m (C_c), bel (C_c) and pl (C_c) represents the total belief mass, belief and plausibility probabilities of co-operative node and m (C_d), bel (C_d) and pl(C_d) represents the total belief mass, belief and plausibility probabilities of

DDoS compromised node. The aforementioned probabilities for co-operative and DDoS are combined together to form a combinational evidence probability M (C_c) and M (C_d). Then the Dempster-Shafer Evidence Probability for identifying the best optimal router node (T_x) of the entire grid is given by equation 5.6.

$$A_i\ (gos) = \frac{\sum_{c \cap d = n} M(C_c).M(C_d)}{1-k} \quad \text{When } d \neq \phi \qquad (5.6)$$

Where the normalization factor $k = \sum_{c \cap d \neq \phi} M(C_c).M(C_d)$

Further, the global best node is re-analyzed based on convolutive x-averaging method that even improves the normalization factor of Dempster-Shafer Evidence Probability using equation 5.7.

$$P_{x-avg}(M_{cd} / T_x) = \sum_{\frac{c+d}{2} \neq n} M(C_c).M(C_d) \qquad (5.7)$$

Finally, the neighbor's best state identification is achieved in each cell or intra-cell by integrating three elucidated parameters $(R_i(t), A_i(los), A_i(gos))$ based on method of exponential moving average using equation 5.8.

$$N_i(best) = \alpha R_i(t) + \beta A_i(los) + \gamma A_i(gos) \qquad (5.8)$$

Where the sum of weights for the elucidated normalized factors is given by equation 5.9.

$$\alpha + \beta + \gamma = 1 \qquad (5.9)$$

Hence the complete information of each cell in CA-MTSA is updated based on the availability function given by equation 5.10.

$$A_i(t) = f((r, R_i(t), A_i(los), A_i(gos), N_i(best)) \qquad (5.10)$$

In addition, the fitness function of CA-MTSA that lies between 0 and 1 is iteratively identified for updating the Tabu-list for improving the effectiveness of search using the aforementioned availability function based on equation 5.11.

$$Fit\ (t+1) = fitness\ \ (A_I(t)) \tag{5.11}$$

The objective function Fit(t+1) must be maximized to identify a node as highly reliable and available for the forwarding packet. Based on the threshold fitness value (0.6), availability is accurately identified periodically and decision on choosing neighboring nodes for efficient packet delivery is facilitated. Thus CA-MTSA handles DDoS attacks by integrating the benefits of Tabu search and cellular automata.

5.2 Algorithm and Flow Chart of the proposed CA-MTSA

The following algorithm 5.1 and Figure 5.2 depicts a Cellular automata based Improved Tabu Search Algorithm for DDoS mitigation.

Algorithm 5.1. - Cellular automata based Improved Tabu Search Algorithm for DDoS mitigation.

Step 1**:** Start.

Step 2: Initialize the Vehicles in 2D Moore Model CA.

Step 3: Initialize the position and Velocity of the vehicles within the range (P_{min}, P_{max}) and (V_{min}, V_{max}) respectively.

Step 4: Compute fitness value using *fit (node (i))=fit(node((*$P_{p(i)}$*), (*$P_{v(i)}$*)* and best fit vehicle by *Bestfit(i)*.

Step 5: Loop1: Till Termination Condition.

Step 6: Loop2: No. of Neighbors.

Step 7: Find reliability of vehicle by *Gwet's kappa reliability factor.*

$$R_i(t) = \frac{p - e(\beta)}{1 - e(\beta)}$$

Step 8: Find Best Reliable Neighbor and update Tabu list.

Step 9: End Loop2.

Step 10: Loop3: No. of Neighbors.

Step 11: Find the local best neighbor by *Bayes theorem based conditional probability* .

$$P(C_c / T_x) = \frac{P(C_c) P(T_x / C_c)}{P(C_c) P(T_x / C_c) + P(C_d) P(T_x / C_d)}$$

Step 12: Find Best local Neighbor and update Tabu list.

Step 13: End Loop3.

Step 14: Loop4:No. of Neighbors.

Step 15: Find Best Reliable global using Dempster-Shafer Evidence Probability by.

$$A_i(gos) = \frac{\sum_{c \cap d = n} M(C_c) \cdot M(C_d)}{1 - k}$$

Step 16: Find Best Reliable global Neighbor and update Tabu list.

Step 17: End Loop4.

Step 18: Loop5:No. of Neighbors.

Step 19: For each cell in the cellular grid find the best neighbor cell through.

$$N_i(best) = \alpha R_i(t) + \beta A_i(los) + \gamma A_i(gos)$$

Step 20: End Loop5.

Step 21: For each neighbor node in the cell check availability function through.

$$A_i(t) = f((r, R_i(t), A_i(los), A_i(gos), N_i(best))$$

Step 22: Find the fitness of each vehicle.

Step 23: If Fit (Vi <0.6) then Vi is DDoS Compromised, Search next reliable neighbor.

Step 24: Else, Forward message.

Step 25: End.

The detailed flow of the CA-MTSA algorithm is depicted in Figure 5.2 to find the availability of neighbor vehicles for providing unbreakable services at all time.

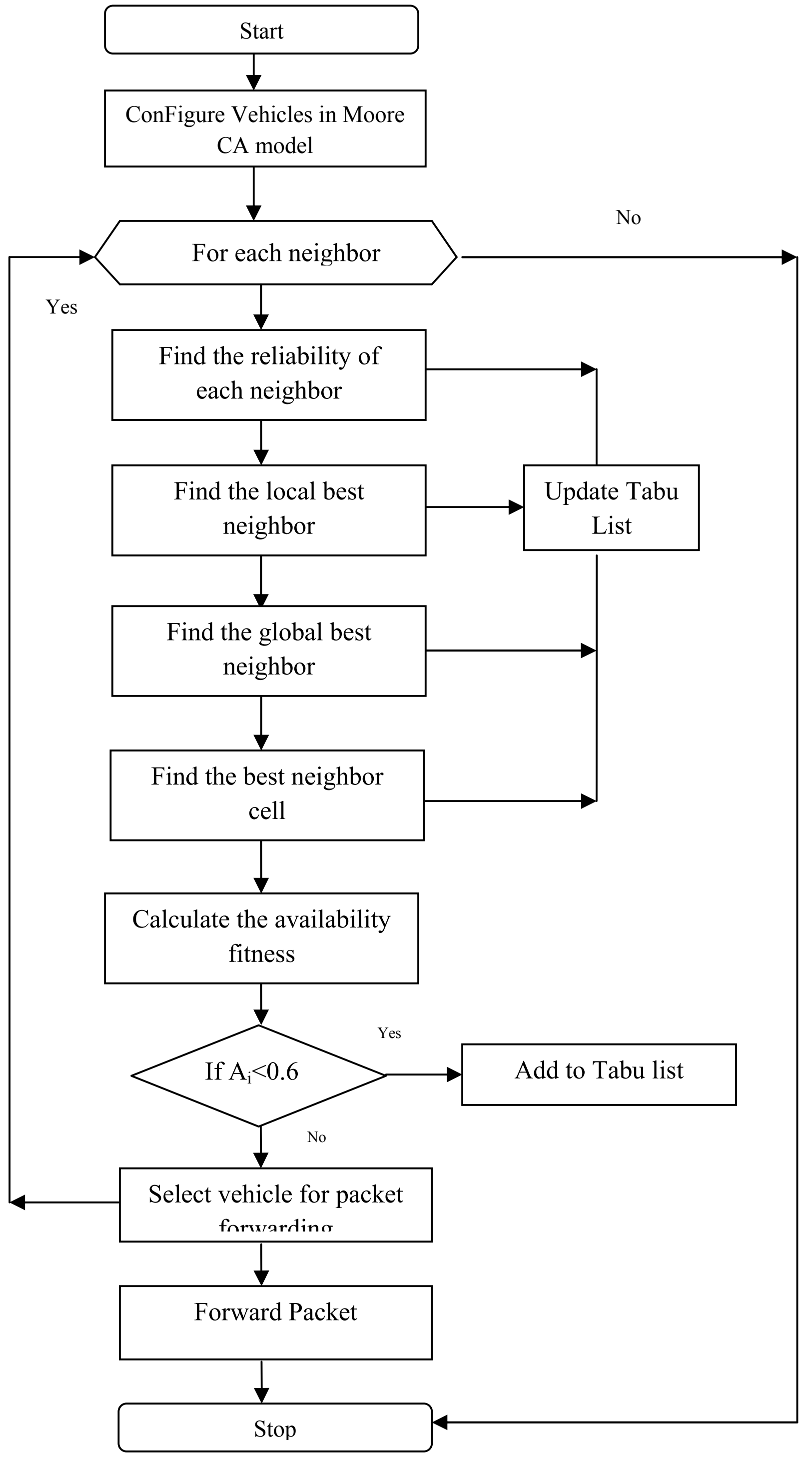


Figure 5.2: Flow chart of CA-MTSA for mitigating DDoS in VANETs

5.3 Simulation Experiments and Results Analysis

The performance of CA-MTSA is evaluated using ns-2 simulator. SUMO traffic simulator is used for generating vehicular mobility traces. Evaluation parameters viz.

packet delivery ratio, end-to-end message delay, prediction variance (meters) and prediction variance (seconds) are used for investigating the performance of CA-MTSA. In this performance analysis, the experiments are carried out either by varying the number of nodes or by varying the prediction interval. The comparative analysis of CA-MTSA is carried out with CAPSO and IPCPSO based on five experiments discussed below. The simulation setup used for experimental analysis is detailed in Table 5.1.

Table 5.1: Simulation setup for evaluating the performance of CA-MTSA

Parameters	Value
Number of vehicular nodes	100,200,300
Range of transmission	400m
Threshold speed	10-40 m/sec
Acceleration of vehicular node	1.0 m/s^2
Retardation of vehicular node	4.5 m/s^2
Simulation time	600s
Prediction interval	10-140 s
MAC protocol	IEEE 802.11p
Refresh interval time	20s

Experiment 1-Performance analysis of CA-MTSA based on prediction variance (meters) by varying prediction interval

In experiment 1, the comparative performance of CA-MTSA over the existing cellular automata based DDoS mitigation approaches like CAPSO and IPCPSO is investigated.

The prediction variance for 100 vehicular nodes is depicted in Figure 5.3 describing the performance of CA-MTSA. The performance is evaluated in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 100 vehicular nodes. The proposed algorithm produced better results when compared to the existing approaches to solve DDoS problem. From the graph it is clear that the prediction variance for CA-MTSA, CAPSO and IPCPSO increases gradually when prediction interval is varied from 10 to 100 seconds. But, CA-MTSA is able to dynamically decrease the variation especially when the prediction interval increases. CA-MTSA is found to exhibit a decrease in prediction variance to a maximum level of 23% than the compared baseline approaches.

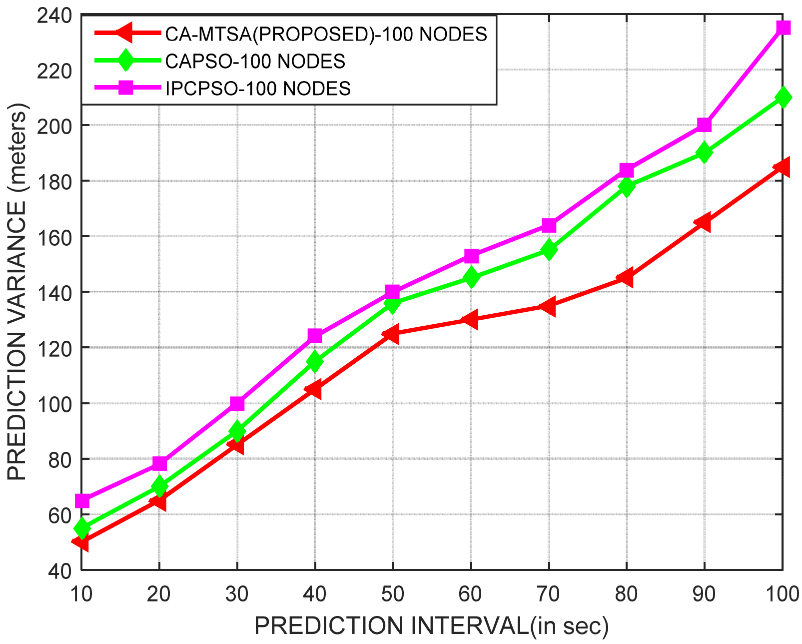


Figure 5.3 -Experiment 1-Performance of CA-MTSA-Prediction variance (meters)-100 nodes

Figure 5.4 portrays the performance of CA-MTSA in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 200 vehicular nodes. Results make it clear that the prediction variance for CA-MTSA, CAPSO and IPCPSO also increases phenomenally when prediction interval is varied from 10 to 100 seconds under the influence of 200 nodes. But, CA-MTSA is able to dynamically decrease the variation even when the number of nodes is increased by 200 especially when the prediction interval increases. CA-MTSA is found to exhibit a decrease in prediction variance to a maximum level of 20% than the compared baseline approaches.

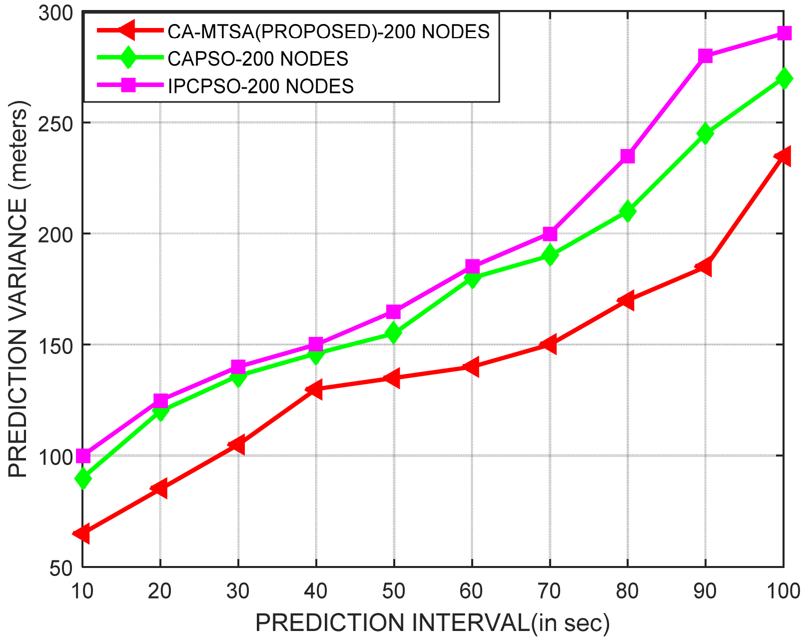


Figure 5.4-Experiment 1-Performance of CA-MTSA-Prediction variance (meters)-200 nodes

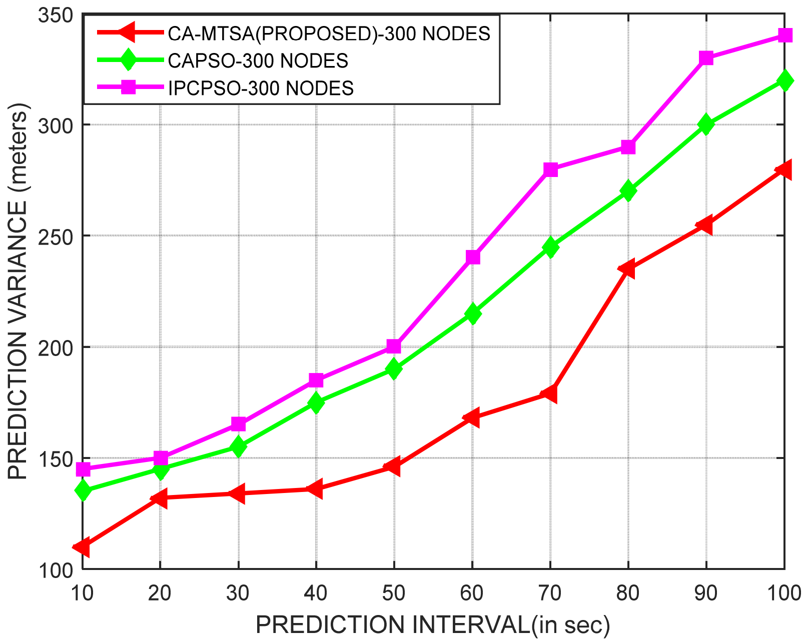


Figure 5.5 - Experiment 1-Performance of CA-MTSA-Prediction variance (meters)-300 nodes

Similarly, Figure 5.5 portrays the performance of CA-MTSA in terms of prediction variance (meters) obtained by varying the prediction interval (seconds) with respect to 300 vehicular nodes. The prediction variance increases as prediction interval increases due to the fast movement of the vehicles. The results make it clear that the prediction variance for CA-MTSA, CAPSO and IPCPSO also increases phenomenally when prediction interval is varied from 10 to 100 seconds under the influence of 300 vehicular nodes. But, CA-MTSA is able to dynamically decrease the variation even when the number of nodes is increased by 200 especially when the prediction interval increases. The proposed CA-MTSA is found to produce a decrease in prediction variance to a maximum level of 17% than the compared baseline approaches.

Experiment 2- Performance analysis of CA-MTSA based on prediction variance by varying number of nodes

In experiment 2, the performance analysis of CA-MTSA, CAPSO and IPCPSO is carried out in terms of prediction variance per second by varying the number of vehicular nodes from 100 to 300 based on varying prediction interval.

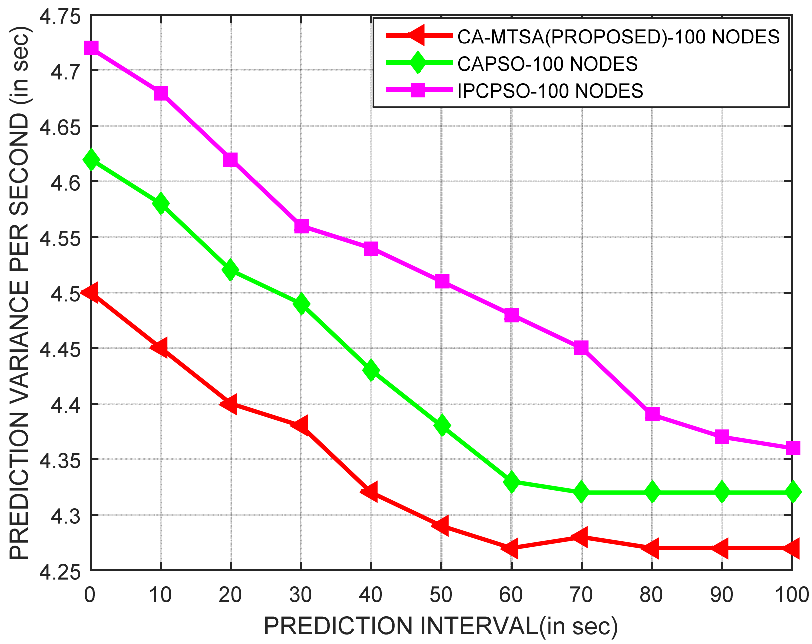


Figure 5.6 - Experiment 2-Performance of CA-MTSA-Prediction variance per second-100 nodes

Figure 5.6 describes the performance of CA-MTSA, CAPSO and IPCPSO based on prediction variance in seconds evaluated by varying the prediction interval. It is found that the prediction variance per second for CA-MTSACAPSO and IPCPSO is considerably decreased when the prediction interval is increased. But CA-MTSA is able to update dynamically than CAPSO and IPCPSO as it uses an improved tabu list for updating information at a rapid rate. Hence CA-MTSA decreases the prediction variance in seconds than CAPSO by 16% and IPCPSO by 19% under the influence of 100 nodes.

The performance of CA-MTSA, CAPSO and IPCPSO based on prediction variance in seconds evaluated by varying the prediction interval with 200 nodes is depicted in Figure 5.7. It is found that the prediction variance per second for CA-MTSA, CAPSO and IPCPSO for every iteration is considerably decreasing. The variance deceases for 200 nodes since the availability of vehicles is more in situation of DDoS mitigation call. The CA-MTSA decreases the prediction variance in seconds than CAPSO by 13% and IPCPSO by 15%.

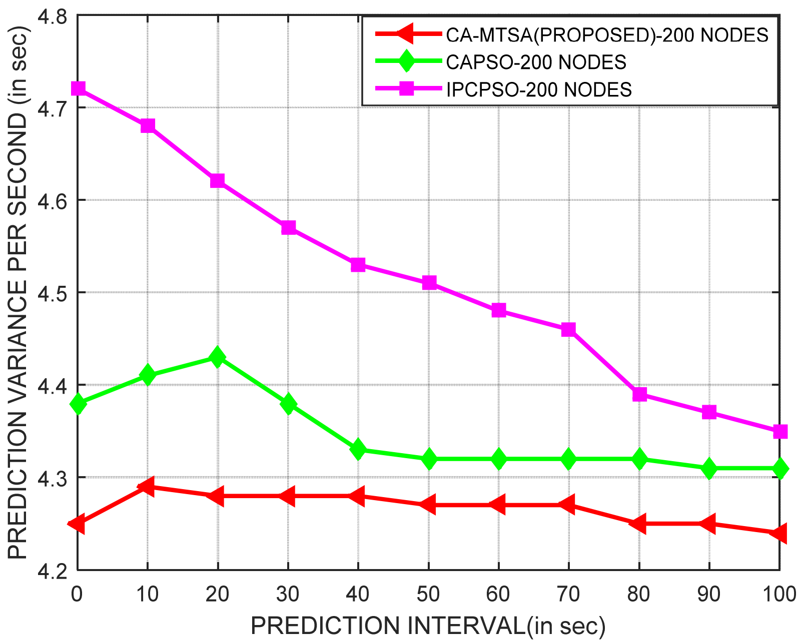


Figure 5.7 - Experiment 2-Performance of CA-MTSA-Prediction variance per second-200 nodes

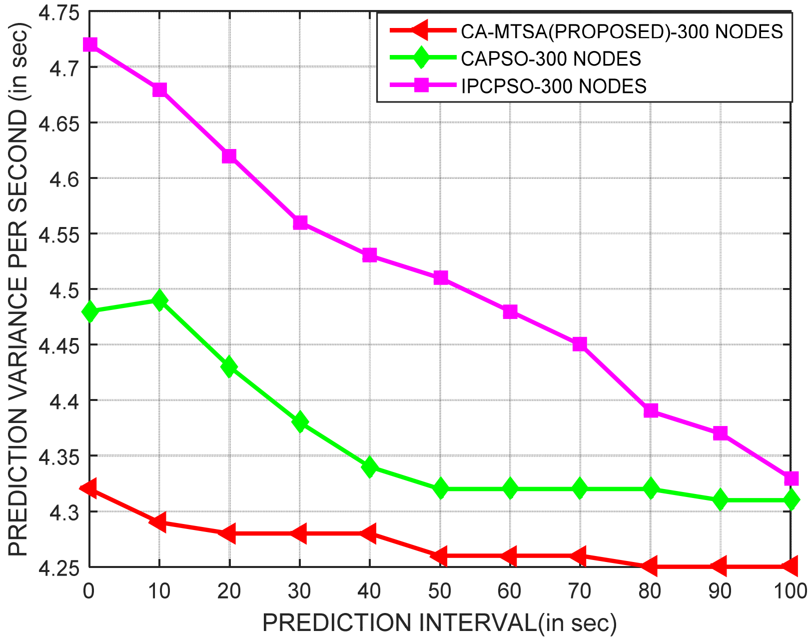


Figure 5.8 - Experiment 2-Performance of CA-MTSA-Prediction variance per second-300 nodes

Similarly, Figure 5.8 shows the performance of CA-MTSA, CAPSO and IPCPSO based on prediction variance in seconds evaluated by varying the prediction interval with 300 nodes. The variance reaches 0 for the prediction interval of 80, 90and 100 since there are more vehicles available with optimal path learning capability. From the graph it is found that the prediction variance per second for CA-MTSA, CAPSO and IPCPSO is highly decreasing when the number of nodes is increased to 300. But CA-MTSA decreases the prediction variance in seconds than CAPSO and IPCPSO to a considerable level of 6% and 10% respectively.

Experiment 3- Performance analysis of CA-MTSA based on prediction variance by varying number of nodes

In experiment 3, the performance analysis of CA-MTSA, CAPSO and IPCPSO is carried out in terms of prediction variance (meters) by varying the number of vehicular nodes from 100 to 300 with fixed accuracy intervals of 70s, 80s and 100s respectively.

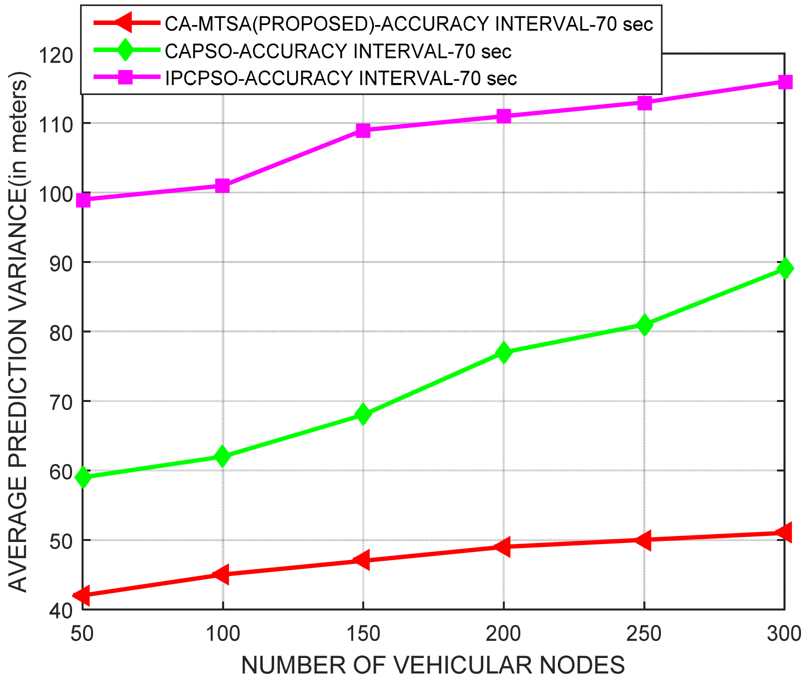


Figure 5.9 - Experiment 3-Performance of CA-MTSA-Average Prediction variance-70 sec

Figure 5.9 represents the relationship of deviation that exists between average prediction accuracy and varying number of vehicular nodes. Initially, this relationship is analyzed with an accuracy level of 70sec as it is considered the minimum optimal value for accuracy prediction for CA-MTSA, CAPSO and IPCPSO. In CA-MTSA mechanism, the average prediction variance of CA-MTSA shows a meager variation of approximately 42-48m with varying number of vehicular nodes, whereas for CAPSO, the deviation is 58-88 and for IPCPSO, it is found to be 98-117m. This least deviation in CA-MTSA is mainly due to the incorporation of ITS, that aids in stable and accurate prediction.

The average prediction variance against the number of vehicular nodes is measured by setting the accuracy level as 80 seconds as a minimum optimal value for accuracy prediction. The graph in Figure 5.10 represents this relationship of deviation that exists between average prediction accuracy and varying number of vehicular nodes. In CA-MTSA mechanism, the average prediction variance of CA-MTSA shows a meager variation of approximately 42-48m with varying number of vehicular nodes, whereas for CAPSO, the deviation is 58-88 and for IPCPSO, it is found to be 98-117m. This least deviation in CA-MTSA is mainly due to the good prediction capability of the algorithm to result in stable and accurate prediction.

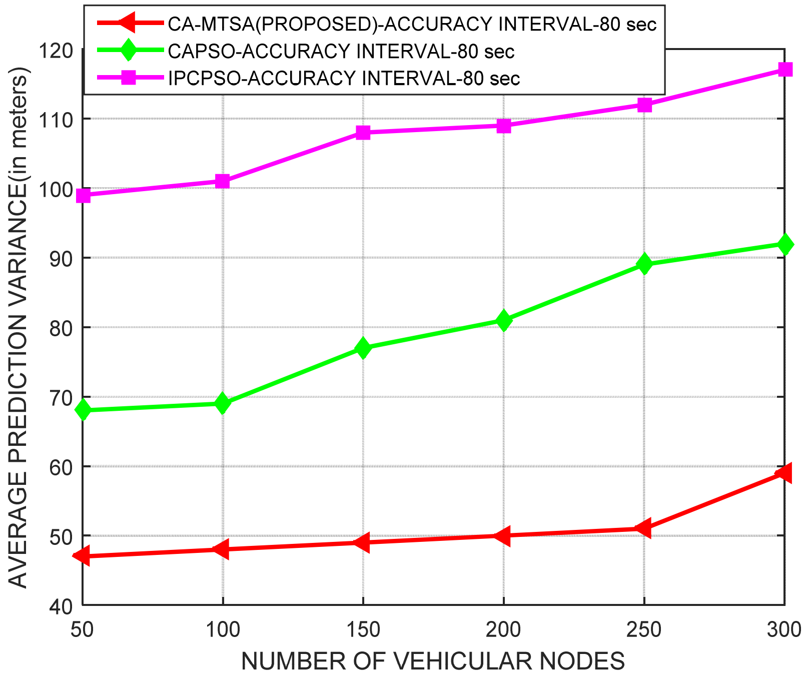


Figure 5.10 - Experiment 3-Performance of CA-MTSA-Average Prediction variance-80 sec

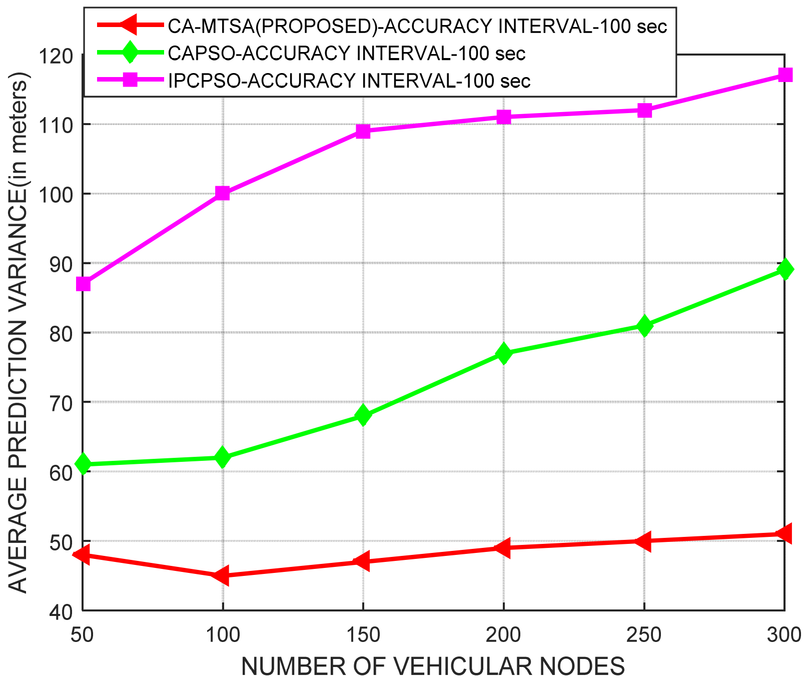


Figure 5.11 - Experiment 3-Performance of CA-MTSA-Average Prediction variance-100 sec

Similarly, the average prediction variance of the CA-MTSA is measured against the number of vehicular nodes by setting the accuracy prediction interval as 100 seconds as shown in Figure 5.11. The relationship that exists between average prediction accuracy and number of vehicular nodes shows that the average prediction variance for accuracy interval of 100 seconds has improvement. From the graph it is inferred that CA-MTSA promoted better prediction than CAPSO and IPCPSO. In CA-MTSA mechanism, the average prediction variance of CA-MTSA shows a meager variation of approximately 48-53m with varying number of vehicular nodes, whereas for CAPSO, the deviation is 62-87 and for IPCPSO, it is found to be 88-119m.This least deviation in CA-MTSA even when the number of vehicular nodes is increased is mainly due to the incorporation of ITS that aids in stable and accurate prediction.

Experiment 4-Performance analysis of CA-MTSA based on PDR by varying the prediction interval

In experiment 4, the performance analysis of CA-MTSA, CAPSO and IPCPSO is carried out in terms of packet delivery ratio by varying the number of vehicular nodes from 100 to 300 under the influence of varying prediction interval.

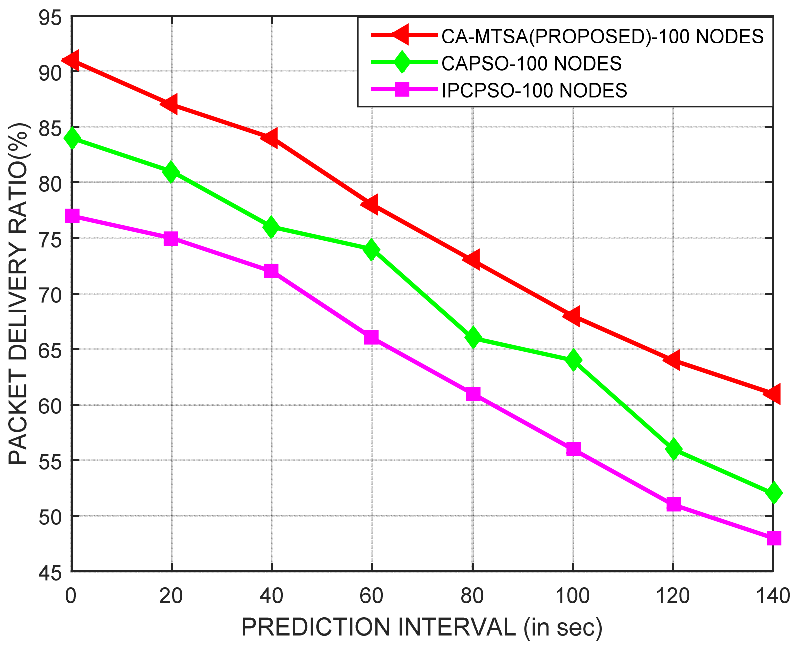


Figure 5.12 - Experiment 4-Performance of CA-MTSA based on PDR-100 nodes

Figure 5.12 represents the deviation in packet delivery ratio evaluated by varying the prediction interval with 100 vehicular nodes. Traditionally, the packet delivery ratio decreases as the prediction interval increases due to the consumption of higher message overhead. It is noticed that the PDR of CA-MTSA, CAPSO and IPCPSO decreases proportionally with the increase in the prediction interval as the amount of traffic introduced into the network increases. The PDR of CA-MTSA is comparatively higher and ranges from 92%-62% in contrast to the range of PDR for CAPSO and IPCPSO that lies between 84% to 53% and 78% to 42% respectively. It is inferred that this increase in PDR is mainly due to the use of improved Tabu search algorithm which reduces packet overhead than CAPSO and IPCPSO.

Likewise, Figure 5.13 depicts the deviation in packet delivery ratio evaluated by varying the prediction interval with 200 vehicular nodes. It is noticed that the PDR of CA-MTSA, CAPSO and IPCPSO decreases with increase in the number of vehicular nodes as the amount of traffic introduced into the network increases proportionally with respect to increased vehicular nodes. However, the PDR of CA-MTSA is comparatively higher and is capable of sustaining its range from 84% to 52% in contrast to the range of PDR for CAPSO and IPCPSO that lies between 77% to 43% and 68% to 34% respectively. Hence it is proved that CA-MTSA is more potential than CAPSO and IPCPSO in increasing the PDR as it dynamically updates the improved Tabu search list which policies packet overhead to a maximum degree even when the number of nodes increases.

Similarly, Figure 5.14 shows packet delivery ratio evaluated by varying the prediction interval with 300 vehicular nodes. It is noticed that the PDR of CA-MTSA, CAPSO and IPCPSO decreases as the prediction interval increases due to the dense traffic in the network. However, the PDR of CA-MTSA is comparatively higher and is potential in withstanding its range from 74% to 43% in contrast to the range of PDR for CAPSO and IPCPSO that lies between 68% to 34% and 57% to 28% respectively.

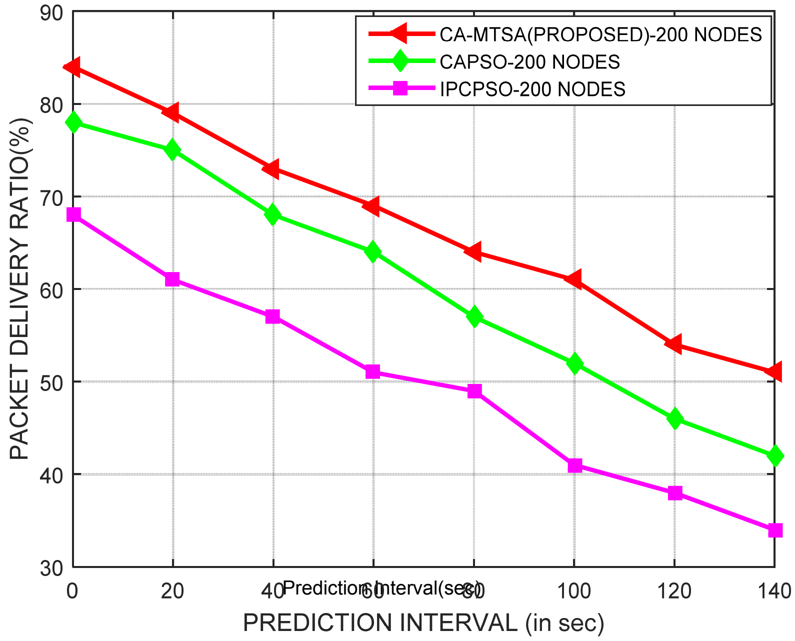


Figure 5.13 - Experiment 4-Performance of CA-MTSA based on PDR-200 nodes

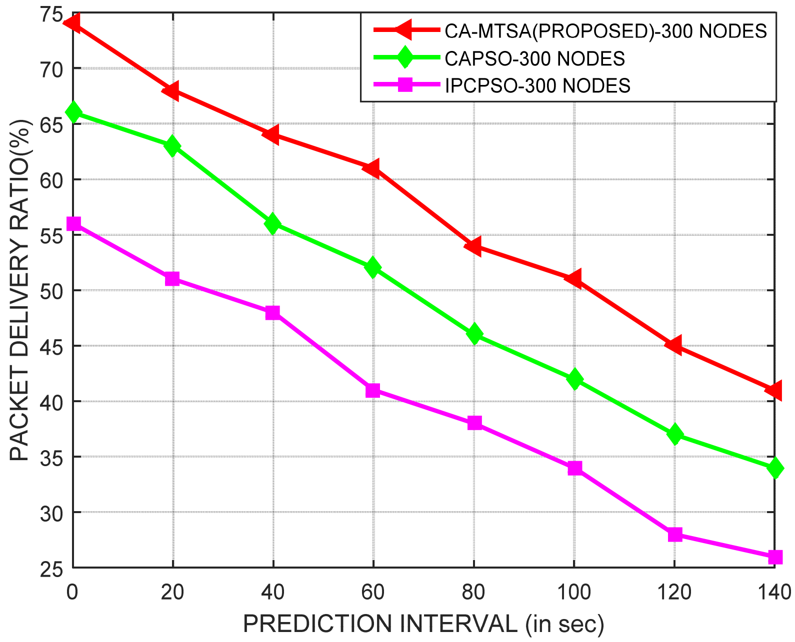


Figure 5.14 - Experiment 4-Performance of CA-MTSA based on PDR-300 nodes

Experiment 5-Performance analysis of CA-MTSA based on End-to-End message delay by varying the prediction interval

In experiment-5, the performance analysis of CA-MTSA, CAPSO, and IPCPSO is carried out in terms of End-to-End message delay by varying the number of vehicular nodes from 100 to 300 under the influence of varying prediction interval.

In Figure 5.15, the performance of CA-MTSA, CAPSO and IPCPSO evaluated based on end-to-end latency by varying the prediction interval with 200 vehicular nodes is portrayed. The results infer that the latency increases linearly when the prediction interval increases but CA-MTSA incurs a lower End-to-End message delay than CAPSO and IPCPSO. This improvement in performance of CA-MTSA is due to the periodic updating of prediction accuracy factor that aids in reducing End-to-End message delay. CA-MTSA is found to decrease the End-to-End message delay by 21% and 26% greater than CAPSO and IPCPSO.

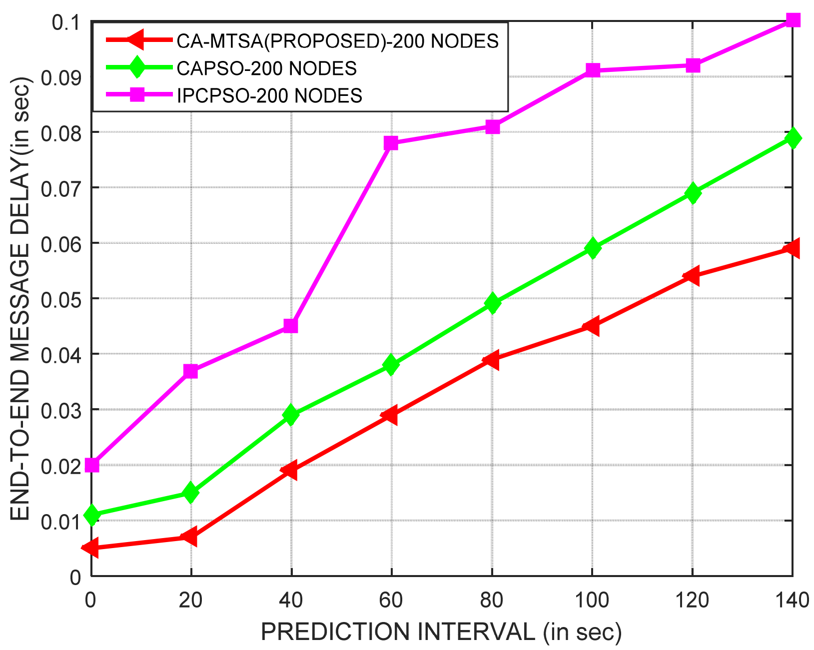


Figure 5.15 - Experiment 4-Performance of CA-MTSA based on End-to-End delay -200 nodes

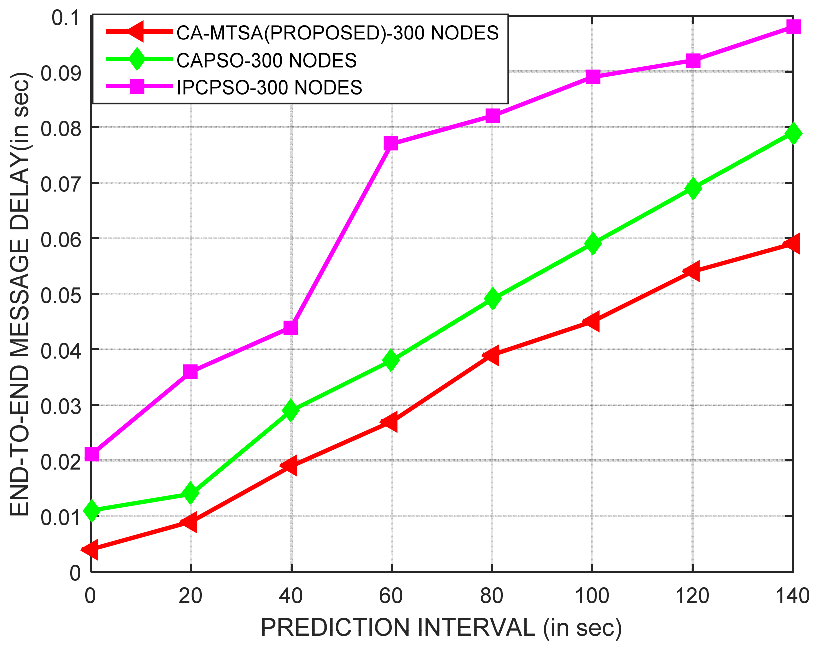


Figure 5.16 - Experiment 4-Performance of CA-MTSA based on End-to-End delay-300 nodes

In Figure 5.16, the performance of CA-MTSA, CAPSO, and IPCPSO evaluated based on end-to-end latency by varying the prediction interval with 300 vehicular nodes is portrayed. The results infer that the latency increases linearly when the prediction interval increases but CA-MTSA incurs a lower End-to-End message delay than CAPSO and IPCPSO even when the nodes are increased. But CA-MTSA improvises its performance by periodically updating prediction accuracy information that significantly reduces End-to-End message delay. CA-MTSA is found to decrease the End-to-End message delay by 18% and 23% greater than CAPSO and IPCPSO under the influence of 300 nodes.

The result analysis of the experiment conducted for the performance metrics prediction variance (seconds), prediction variance (meters) and average prediction variance are given in the following table. The table compares the proposed algorithm with each of the existing algorithms.

Table 5.2 Performance comparison of Average Prediction variance (measured in meters) proposed CA-MTSA algorithm with existing algorithms

Algorithm	No. of vehicles		
	100	200	300
IPCPSO vs CA-MTSA	19	35	28
CAPSO vs CA-MTSA	14	28	22

Table 5.2 gives the comparison of prediction variance of proposed algorithm with other baseline approaches with the decrease in percentage. For 100 vehicles the decrease in prediction variance is between 14% to 19 % when compared to the CAPSO, IPCPSO algorithms. For 200 vehicles the decrease in prediction variance is between 28-35% when compared to the CAPSO, IPCPSO algorithms. For 300 vehicles the decrease in prediction variance is between 22% to 28% when compared to the CAPSO, IPCPSO algorithms. By integrating the features of Cellular Automata and Tabu Search the CA-MTSA algorithm results in better prediction of the best neighboring node of each cell in the entire grid under local search or global search.

Table 5.3 Performance comparison of Prediction variance (measured in seconds) proposed CA-MTSA algorithm with existing algorithms

Algorithm	No. of vehicles		
	100	200	300
IPCPSO vs CA-MTSA	5	10	9
CAPSO vs CA-MTSA	3	2	3

Table 5.3 gives the comparison of delays incurred by proposed algorithm with other baseline approaches with the decrease in percentage. For 100 vehicles the decrease in prediction variance is between 3% to 5% when compared to the CAPSO, IPCPSO

algorithms. For 200 vehicles the decrease in prediction variance is between 2% to 10% when compared to the CAPSO, IPCPSO algorithms. For 300 vehicles the decrease in prediction variance is between 3% to 9% when compared to the CAPSO, IPCPSO algorithms. From the analysis it is revealed that CA-MTSA decreases in prediction delay when compared to other existing algorithms.

Table 5.4 Performance comparison of Average Prediction variance (measured in meters) proposed CA-MTSA algorithm with existing algorithms

Algorithm	Prediction interval		
	100	200	300
IPCPSO	98-117	100-118	88-119
CAPSO	58-88	68-92	62-87
CA-MTSA	**42-51**	**48-60**	**48-51**

Table 5.4 gives the comparison of average prediction variance of proposed algorithm with other baseline approaches. The experiment was conducted by varying the number of nodes from 50 to 300 and measuring the average prediction variance for each group.

The result from the Table 5.4 reveals that for prediction interval of 70s the proposed algorithm has average prediction variance from 42-51m which is less when compared to all other existing algorithms. Similarly, the average prediction variance is measured by varying the number of vehicles with different prediction intervals of 80s and 100s which resulted in average prediction variance of 48-60 and 48-51 respectively and was found that the proposed algorithm has less average prediction variance among all existing algorithms.

Table 5.5 Performance comparison of PDR of proposed CA-MTSA algorithm with existing algorithms

Algorithm	No. of vehicles		
	100	**200**	**300**
IPCPSO vs CA-MTSA	13	21	25
CAPSO vs CA-MTSA	7	8	10

Table 5.5 compares the PDR of CA-MTSA with existing algorithms with the increase percentage of proposed algorithm with IPCPSO and CAPSO. By varying the density of nodes from 100 to 300 the PDR is measured and found to have 20% increase in PDR than IPCPSO and 8% increase in PDR than CAPSO.

Table 5.6 Performance comparison of End to End delay of proposed CA-MTSA algorithm with existing algorithms

Algorithm	No. of vehicles	
	200	300
IPCPSO vs CA-MTSA	65	75
CAPSO vs CA-MTSA	36	63

Table 5.6 gives the analysis of End to End delay of CA-MTSA with existing algorithms. By varying the density of nodes from 200 to 300, the End to End delay is measured and found to have decrease in percentage by 70% than IPCPSO and 50% than CAPSO.

5.4 Summary

In this chapter CA-MTSA integrates the Tabu Search and Cellular Automata to improve the global searching ability of the algorithm to search for the optimal neighbor for data forwarding. The proposed work successfully mitigates the DDoS by selecting the reliable node from the CA Moore model. The availability of each node for effective data forwarding depends on its past velocity, past reliability factor, local optimal state, global optimal state and neighbor's best state. Simulation results prove that CA-MTSA is significant compared to the considered baseline techniques in terms of robustness, local search capability, global search capability and its efficiency in identifying reliable routing path under DDoS attack. Results also prove that CA-MTSA is effective in terms of packet delivery rate by 23% and prediction accuracy by 27% even when the number of vehicular nodes is increased.

CHAPTER 6

COMPARATIVE INVESTIGATIONS OF MACOA-CA, CA-IABCA, CA-MABCA THROUGH MULTIMODAL FUNCTIONS

To validate and compare the optimization algorithms test problems are required with different characteristics and constraints. These test functions are called as the bench mark functions with properties like continuous, differentiable, scalable, multimodal, scalable, non-scalable, etc. The optimization algorithms can be tested against these benchmarks for measuring the efficiency and reliability of the algorithm against the existing algorithms. In literature many benchmark functions are available which can be chosen based on their properties and algorithm capability. The function with more than one local optimum is called multi modal function which refers to the most difficult class of problems to be solved. The proposed algorithms such as MACOA-CA, CA-IABCA, MABCA and CA-MTSA are validated based on selected benchmark multimodal functions to measure their performance.

6.1 Performance analyses based on Quartic function

The importance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is estimated based on the benchmark multi-modal function Quartic with search dimension D=5 and D=10 respectively.

Performance analysis of MACOA-CA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to the benchmark functions Quartic with search dimension D=5 by varying the number of search iterations that pertain to the average rate of function values.

From Figure 6.1, it is found that MACOA-CA initially provides a unique level of performance with Quartic. MACOA-CA confirms a more systematic growth than CA-ACOA due to permissible global search facility made possible by it.

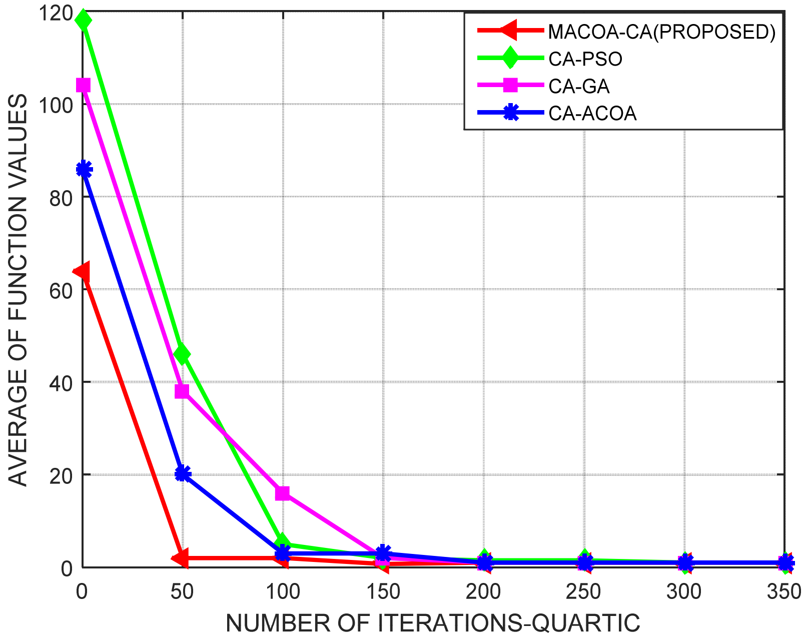


Figure 6.1 - Performance of MACOA-CA- Quartic (D=5)

From the graph in Figure 6.1 MACOA-CA attains its convergence after 50^{th} iteration and meets the mean global optimum for Quartic after 285 iterations.

Performance analysis of MACOA-CA with search dimension (D=10)

The comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to Quartic function with search dimension D=10 by varying the number of search iterations which pertain to the average rate of function values.

From 6.2 it is found that MACOA-CA initially provides a unique level of performance with Quartic benchmark multi-modal functions. Initially the MACOA-CA results in higher value and after 50th iteration it starts to converge and attains the mean global optimum for Quartic after 295 iterations.

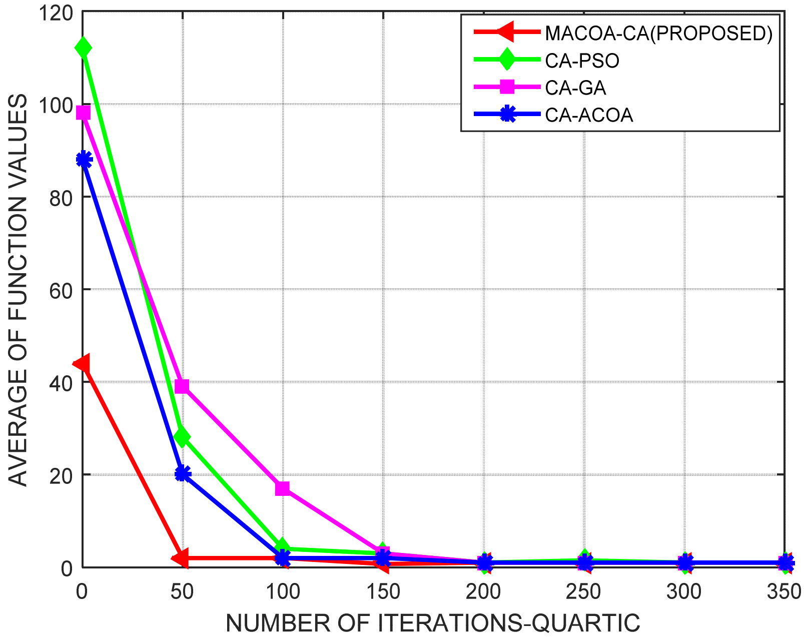


Figure 6.2 - Performance of MACOA-CA- Quartic (D=10)

6.2 Performance analyses based on Schwefel-2.26 function

The importance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is estimated based on the benchmark multi-modal function Schwefel-2.26 with search dimension D=5 and D=10 respectively.

Performance analysis of MACOA-CA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to the benchmark function Schwefel-2.26 with search dimension D=5 by varying the number of search iterations that pertains to the average rate of function values.

It is found that MACOA-CA initially provides a unique level of performance with Schwefel-2.26 shown in Figure 6.3. MACOA-CA confirms a more systematic growth than CA-ACOA due to permissible global search facility. The MACOA-CA provides better convergence degree than CA-ACOA, CA-GA and CA-PSO and it attains the global point after 275 iterations using Schwefel-2.26.

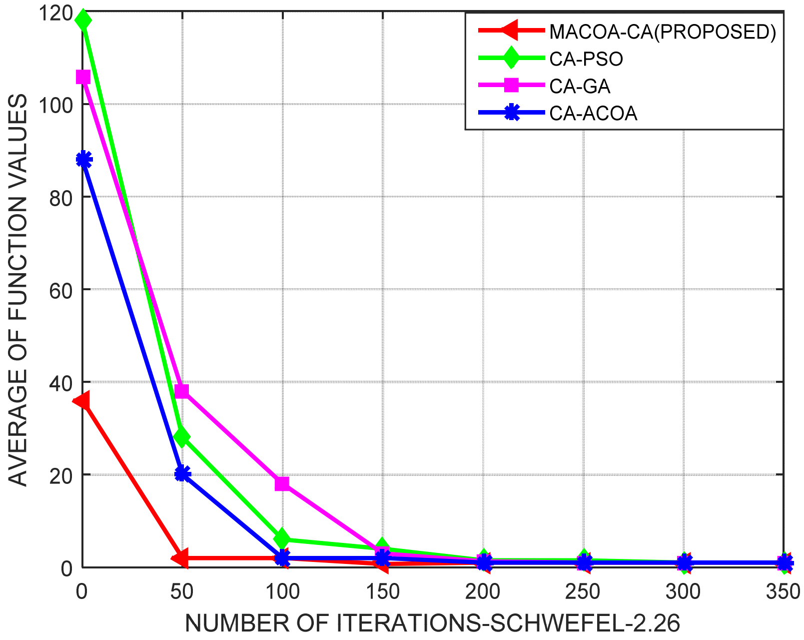


Figure 6.3 -Performance of MACOA-CA- Schwefel-2.26 (D=5)

The proposed algorithm MACOA-CA attains the mean global optimum for Schwefel-2.26 after 290 iterations. Hence MACOA-CA is potent in its performance during the investigation with Schwefel-2.26 function as it reaches the optimal point of convergence at a more rapid rate of 21% than the compared Quartic, Exponential and Sumsquare functions.

Performance analysis of MACOA-CA with search dimension (D=10)

The comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to Schwefel-2.26 function with search dimension D=10 by varying the number of search iterations which pertains to the average rate of function values.

The MACOA-CA initially provides a unique level of performance with Schwefel-2.26 benchmark multi-modal functions as shown in Figure 6.4. MACOA-CA provides better convergence degree than CA-ACOA, CA-GA and CA-PSO, and it attains global optimal point after 285 iterations using Schwefel-2.26.

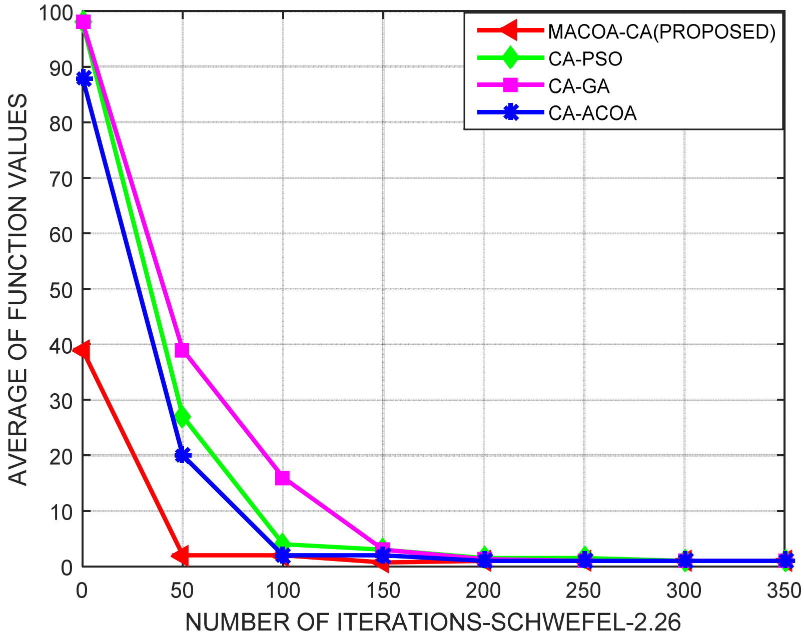


Figure 6.4 - Performance of MACOA-CA- Schwefel-2.26 (D=10)

Similarly, MACOA-CA attains the mean global optimum for Schwefel-2.26 after 298 iterations. Hence MACOA-CA is potent in its performance during the investigation with Schwefel-2.26 function as it reaches the optimal point of convergence at a more rapid rate of 17% than the compared Quartic, Exponential and Sumsquare functions even when the search dimensions are increased.

6.3 Performance analysis based on Exponential function

The importance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is estimated based on the benchmark multi-modal function Schwefel-2.26 with search dimension D=5 and D=10 respectively.

Performance analysis of MACOA-CA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to the benchmark function Exponential with search dimension D=5 by varying the number of search iterations that pertain to the average rate of function values.

The Figure 6.5, shows that MACOA-CA initially provides a unique level of performance with Exponential. MACOA-CA confirms a more systematic growth than CA-ACOA due to permissible global search facility made possible by it.

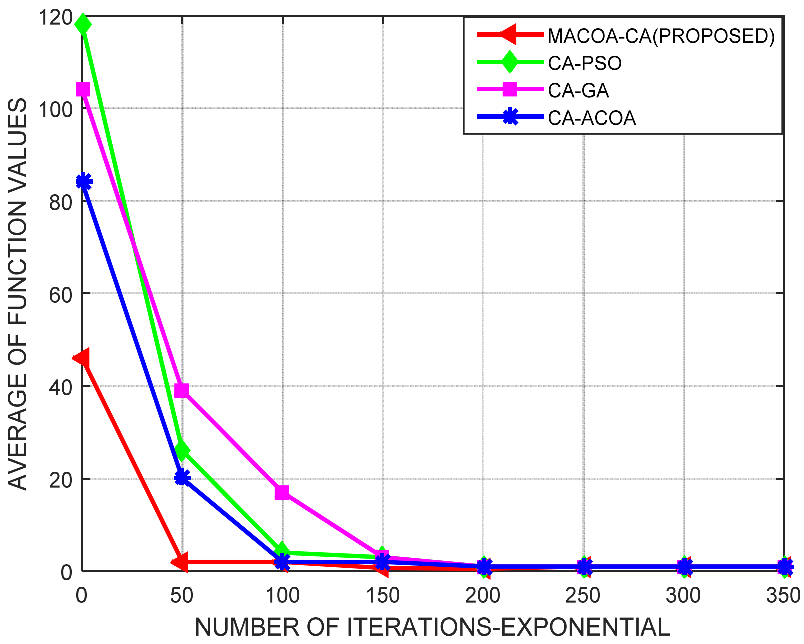


Figure 6.5 - Performance of MACOA-CA- Exponential (D=5)

The investigation of MACOA-CA reveals that the algorithm has a faster convergence rate compared to other existing algorithms. The MACOA-CA attains the mean global optimum for Exponential after 298 iterations.

Performance analysis of MACOA-CA with search dimension (D=10)

The comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to Exponential function with search dimension D=10 by varying the number of search iterations which pertain to the average rate of function values.

It is found that MACOA-CA initially provides a unique level of performance with Exponential benchmark multi-modal functions from the Figure 6.6.

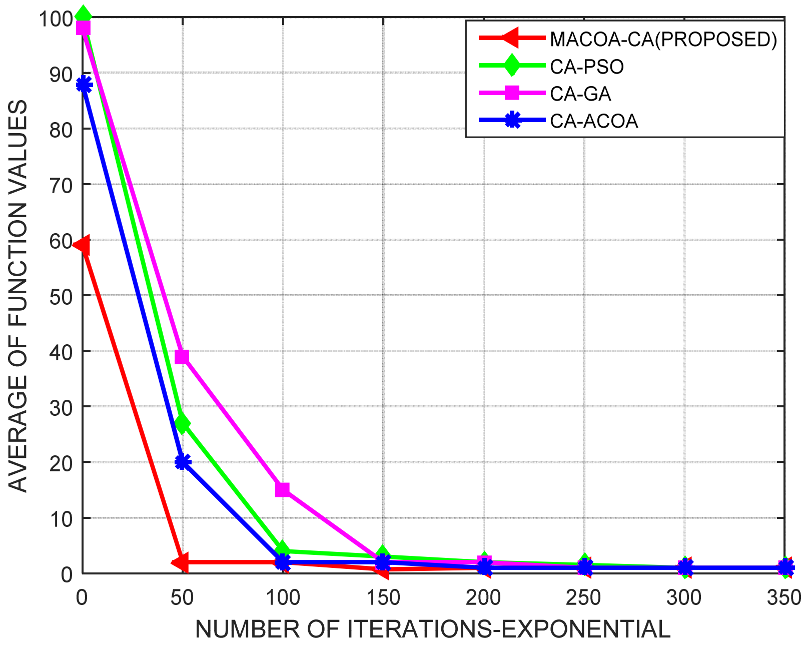


Figure 6.6 - Performance of MACOA-CA -Exponential (D=10)

The MACOA-CA attains the mean global optimum for Exponential after 305 iterations. The graph reveals that the proposed algorithm has a faster convergence rate than the CA-PSO, CA-GA and CA-ACOA algorithms.

6.4 Performance analyses based on Sumsquare function

The importance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is estimated based on the benchmark multi-modal function Sumsquare with search dimension D=5 and D=10 respectively.

Performance analysis of MACOA-CA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to the benchmark function Sumsquare with search dimension D=5 by varying the number of search iterations that pertain to the average rate of function values.

Figure 6.7, proves that MACOA-CA initially provides a unique level of performance with Sumsquare. MACOA-CA confirms a more systematic growth than CA-ACOA due to permissible global search facility.

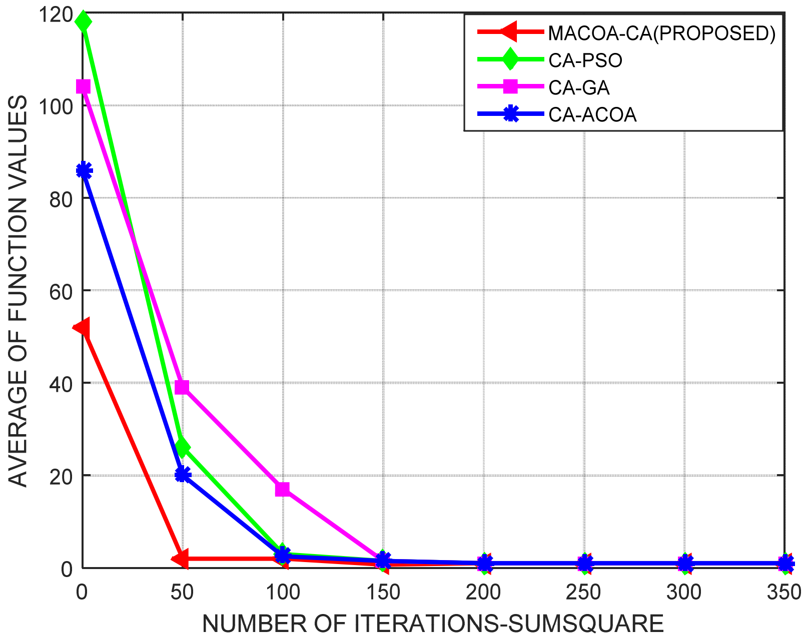


Figure 6.7 - Performance of MACOA-CA- Sumsquare (D=5)

The MACOA-CA attains the mean global optimum for Sumsquare after 285 iterations with the faster convergence rate.

Performance analysis of MACOA-CA with search dimension (D=10)

The comparative performance of MACOA-CA over CA-ACOA, CA-GA and CA-PSO is investigated with respect to Sumsquare function with search dimension D=10 by varying the number of search iterations which pertain to the average rate of function values.

Figure 6.8 shows that MACOA-CA initially provides a unique level of performance with Sumsquare benchmark multi-modal functions. Figure 6.8 depicts that MACOA-CA compared to the existing algorithms attains the best result. The MACOA-CA attains the mean global optimum for Sumsquare after 285 iterations.

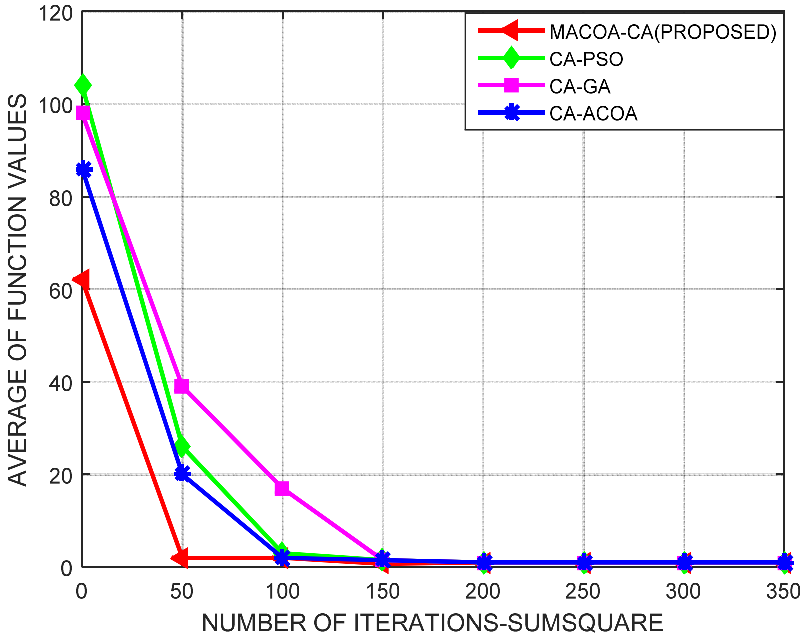


Figure 6.8 - Experiment 5-Performance of MACOA-CA- Sumsquare (D=10)

6.5 Performance analyses based on Rastrigin function

The importance of CA-IABCA over SOA-ABCA, SOA-ACO and SOA-PSO is estimated based on the benchmark multi-modal function Rastrigin with search dimension D=5 and D=10 respectively.

Performance analysis of CA-IABCA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over SOA-ABCA, SOA-ACO and SOA-PSO is investigated with respect to the benchmark function Rastrigin with search dimension D=5 by varying the number of search iterations that pertain to the average rate of function values.

It is observed that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Rastrigin function as shown in Figure 6.9. CA-IABCA also exhibits a gradual fluctuation than SOA-ABCA due to wider feasible search solutions. In addition, the CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that CA-IABCA attains the mean global optimum for Rastrigin function after 275 iterations.

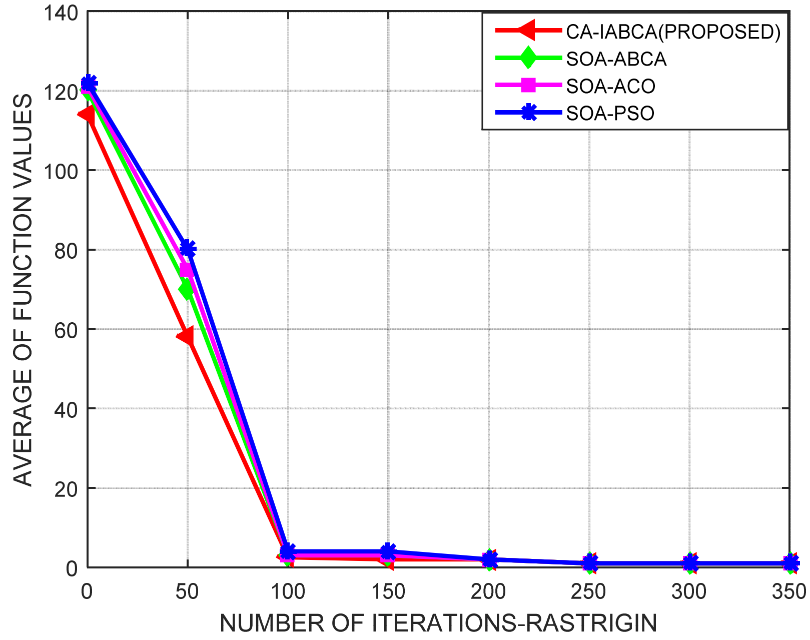


Figure 6.9 - Performance of CA-IABCA-Rastrigin Function (D=5)

Performance analysis of CA-IABCA with search dimension (D=10)

In the experiment, the performance analysis of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is carried out in terms of four benchmark functions such as Rastrigin function with search dimension D=10 by varying the number of iterations of search with respect to mean of function values.

Figure 6.10, shows that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Rastrigin function even when ‘D’ is increased to 10. CA-IABCA also exhibits a gradual fluctuation than SOA-ABCA and it is capable of ensuring wider search solutions. In addition, CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that that CA-IABCA attains the mean global optimum for Rastrigin function after 255 iterations.

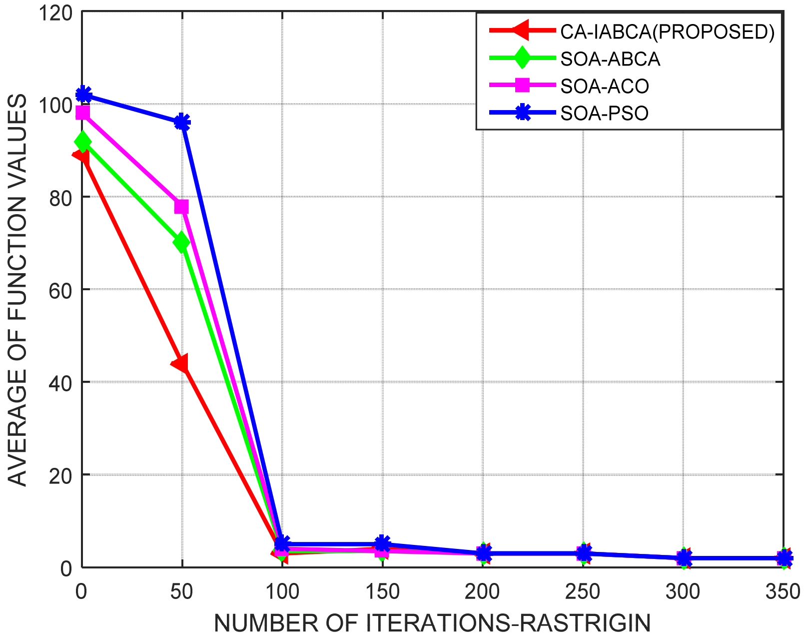


Figure 6.10 - Performance of CA-IABCA-Rastrigin Function (D=10)

6.6 Performance analyses based on Ackley function

The importance of CA-IABCA over SOA-ABCA, SOA-ACO and SOA-PSO is estimated based on the benchmark multi-modal function Ackley with search dimension D=5 and D=10 respectively.

Performance analysis of CA-IABCA with search dimension (D=5)

In the experiment, the comparative performance of MACOA-CA over SOA-ABCA, SOA-ACO and SOA-PSO is investigated with respect to the benchmark function Ackley with search dimension D=5 by varying the number of search iterations that pertain to the average rate of function values.

In Figure 6.11, it is observed that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Ackley function. CA-IABCA also exhibits a gradual fluctuation than SOA-ABCA due to wider search solutions. In addition, the CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that that CA-IABCA attains the mean global optimum for Ackley function after 285 iterations.

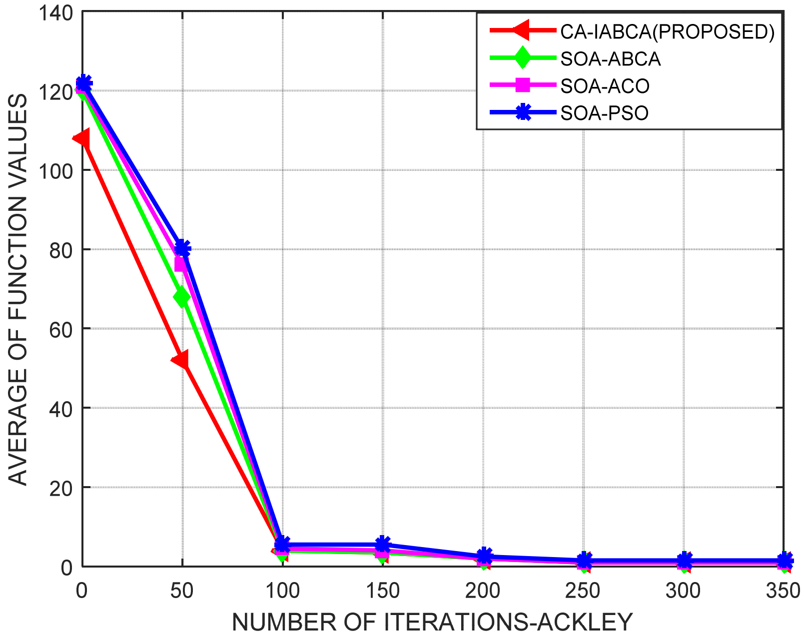


Figure 6.11 - Performance of CA-IABCA-Ackley Function (D=5)

Performance analysis of CA-IABCA with search dimension (D=10)

In the experiment, the performance analysis of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is carried out in terms of four benchmark functions such as Ackley function with search dimension D=10 by varying the number of iterations of search with respect to mean of function values.

Figures 6.12, shows that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Ackley function even when 'D' is increased to 10. CA-IABCA also exhibits a less gradual fluctuation than SOA-ABCA and it is capable of ensuring wider search solutions. In addition, CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that CA-IABCA attains the mean global optimum for Ackley function after 258 iterations.

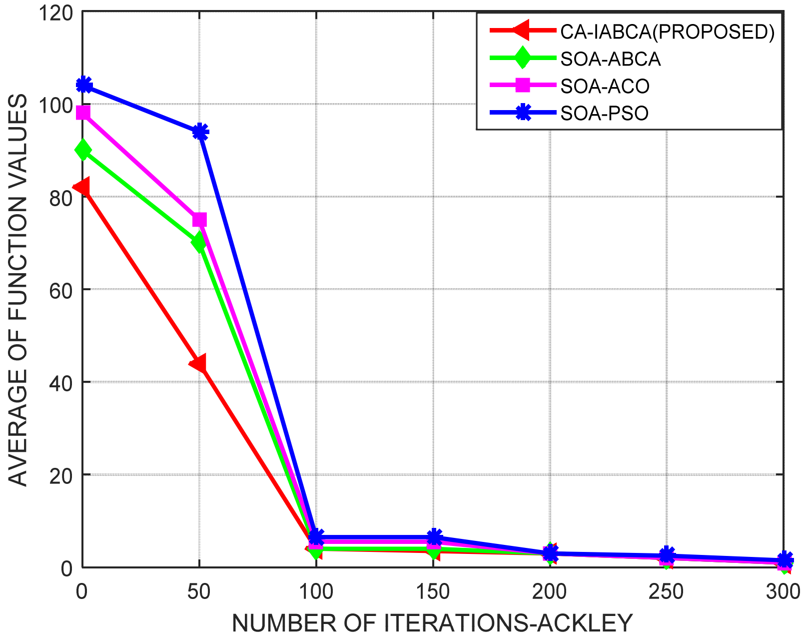


Figure 6.12 –Performance of CA-IABCA-Ackley Function (D=10)

6.7 Performance analyses based on Griewank function

The importance of CA-IABCA over SOA-ABCA, SOA-ACO and SOA-PSO is estimated based on the benchmark multi-modal function Griewank with search dimension D=5 and D=10 respectively.

Performance analysis of CA-IABCA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over SOA-ABCA, SOA-ACO and SOA-PSO is investigated with respect to the benchmark function Griewank with search dimension D=5 by varying the number of search iterations that pertain to the average rate of function values.

Figure 6.13, shows that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Griewank function. CA-IABCA also exhibits a less gradual fluctuation than SOA-ABCA due to wider search solutions. In addition, the CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that that CA-IABCA attains the mean global optimum for Griewank function after 280 iterations.

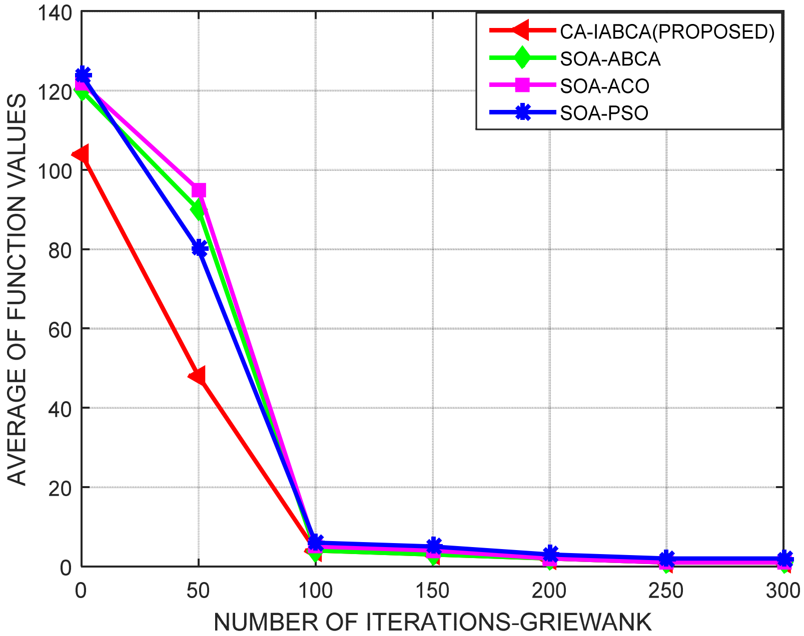


Figure 6.13 - Performance of CA-IABCA-Griewank Function (D=5)

Performance analysis of CA-IABCA with search dimension (D=10)

In experiment, the performance analysis of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is carried out in terms of four benchmark functions such as Griewank function with search dimension D=10 by varying the number of iterations of search with respect to mean of function values.

Figure 6.14, shows that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Griewank function even when 'D' is increased to 10. CA-IABCA also exhibits a gradual fluctuation than SOA-ABCA and it is capable of ensuring wider search solutions. In addition, CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that that CA-IABCA attains the mean global optimum for Griewank function after 260 iterations.

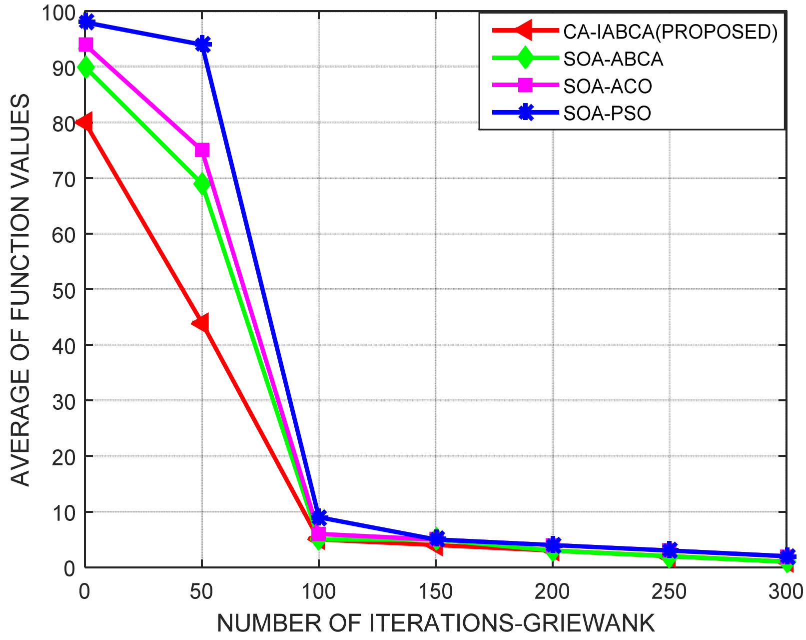


Figure 6.14 - Performance of CA-IABCA-Griewank Function (D=10)

6.8 Performance analyses based on Shaffer function

The importance of CA-IABCA over SOA-ABCA, SOA-ACO and SOA-PSO is estimated based on the benchmark multi-modal function Shaffer with search dimension D=5 and D=10 respectively.

Performance analysis of CA-IABCA with search dimension (D=5)

In experiment, the comparative performance of MACOA-CA over SOA-ABCA, SOA-ACO and SOA-PSO is investigated with respect to the benchmark function Shaffer with search dimension D=5 by varying the number of search iterations that pertains to the average rate of function values.

Figure 6.15, shows that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Shaffer function. CA-IABCA also exhibits a less gradual fluctuation than SOA-ABCA due to wider search solutions made. In addition, the CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and

SOA-PSO. It is also interesting to realize that that CA-IABCA attains the mean global optimum for Shaffer function after 270 iterations.

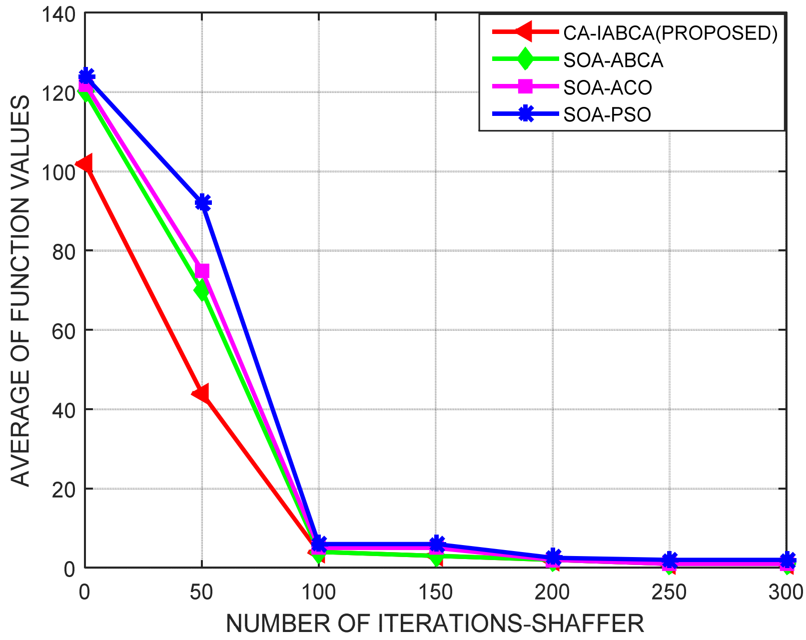


Figure 6.15 - Performance of CA-IABCA-Shaffer Function (D=5)

Performance analysis of CA-IABCA with search dimension (D=10)

In experiment, the performance analysis of CA-IABCA, SOA-ABCA, SOA-ACO and SOA-PSO is carried out in terms of four benchmark functions such as Shaffer function with search dimension D=10 by varying the number of iterations of search with respect to mean of function values

Figures 6.16, shows that CA-IABCA exhibits a similar level of performance to SOA-ABCA when evaluated with Shaffer function even when 'D' is increased to 10. CA-IABCA also exhibits a less gradual fluctuation than SOA-ABCA and it is capable of ensuring wider search solutions. In addition, CA-IABCA provides better convergence degree than SOA-ABCA, SOA-ACO and SOA-PSO. It is also interesting to realize that CA-IABCA attains the mean global optimum for Shaffer function after 252 iterations.

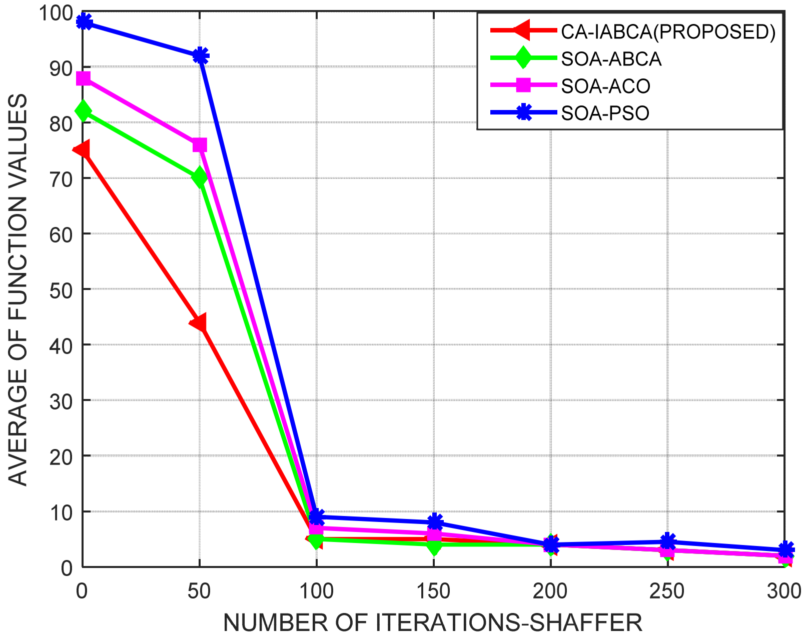


Figure 6.16 - Experiment -Performance of CA-IABCA-Shaffer Function (D=10)

6.9 Performance analyses of CA-MABCA based on Sphere function

In the experiment, the superior performance of CA-MABCA and its potentiality is analyzed based on multi-modal benchmark function namely Sphere under the search dimension of D=20 by varying the number of iterations of search with respect to average function values.

Figure 6.17 shows that CA-MABCA confirm almost a similar kind of performance to CA-ABCA and CA-ACO during its evaluation facilitated with Sphere multi-modal function. But CA-MABCA is seen to exhibit better performance in a gradual manner than CA-ABCA and CA-ACO as it uses the benefits of chaotic system and opposition-based learning for steepening the optimal convergence rate of finding optimal solution. CA-MABCA is found to facilitate a better convergence rate after 280 iterations than CA-ABCA and CA-ACO.

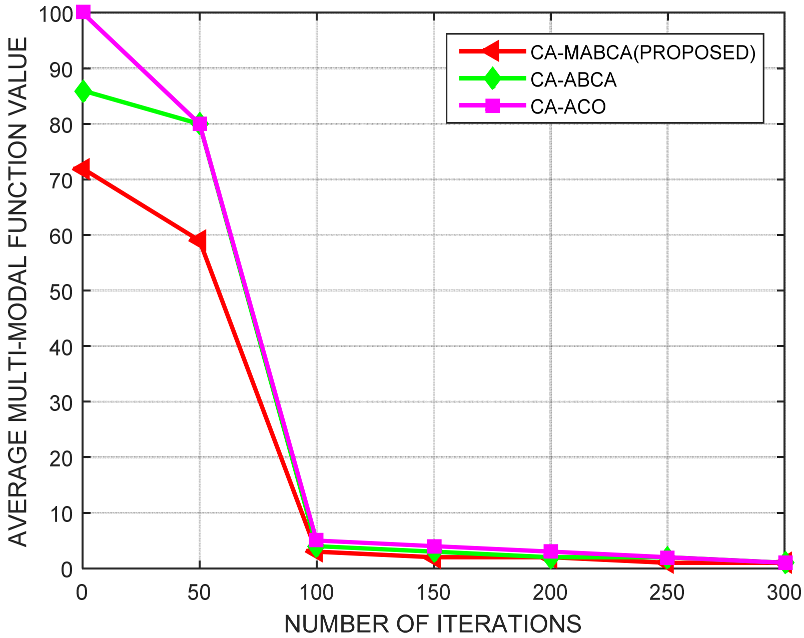


Figure 6.17 - Performance of CA-MABCA-Sphere Function

6.10 Performance analyses of CA-MABCA based on Step function

In the experiment, the superior performance of CA-MABCA and its potentiality is analyzed based on multi-modal benchmark function namely Step under the search dimension of D=20 by varying the number of iterations of search with respect to average function values.

Figure 6.18 shows that CA-MABCA confirm almost a similar kind of performance to CA-ABCA and CA-ACO during its evaluation facilitated with Step multi-modal function. But CA-MABCA is seen to exhibit better performance than CA-ABCA and CA-ACO in a gradual manner as it uses the benefits of chaotic system and opposition-based learning for steepening the optimal convergence rate of finding optimal solution. CA-MABCA is found to facilitate a better convergence rate after 280 iterations than CA-ABCA and CA-ACO.

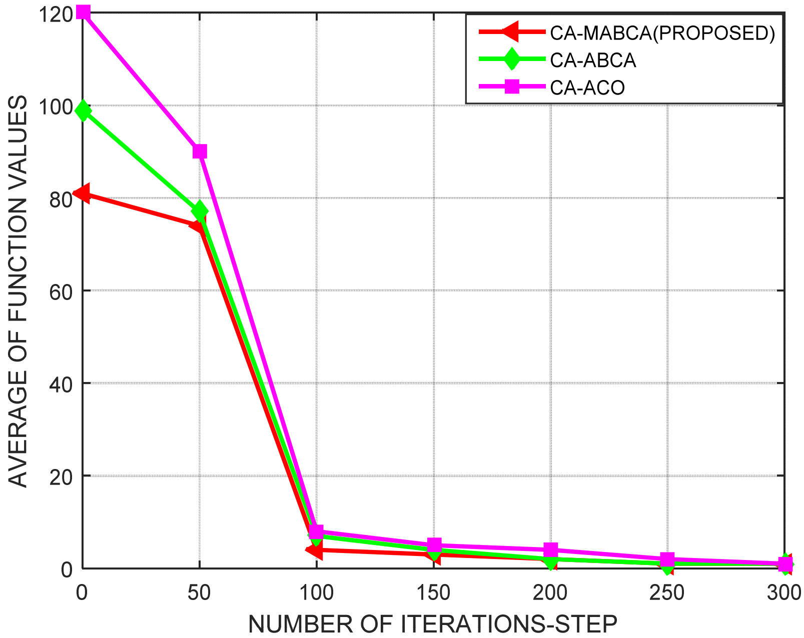


Figure 6.18 - Performance of CA-MABCA-Step Function

6.11 Performance analyses of CA-MABCA based on Schwefel-2.21 function

In the experiment, the superior performance of CA-MABCA and its potentiality is analyzed based on multi-modal benchmark function namely Schwefel-2.21 under the search dimension of D=20 by varying the number of iterations of search with respect to average function values.

Figure 6.19 shows that CA-MABCA confirm almost a similar kind of performance to CA-ABCA and CA-ACO during its evaluation facilitated with Schwefel-2.21 multi-modal function. But CA-MABCA is seem to exhibit better performance in a gradual manner than CA-ABCA and CA-ACO as it uses the benefits of chaotic system and opposition-based learning for steepening the optimal convergence rate of finding optimal solution. CA-MABCA is found to facilitate a better convergence rate after 280 iterations than CA-ABCA and CA-ACO.

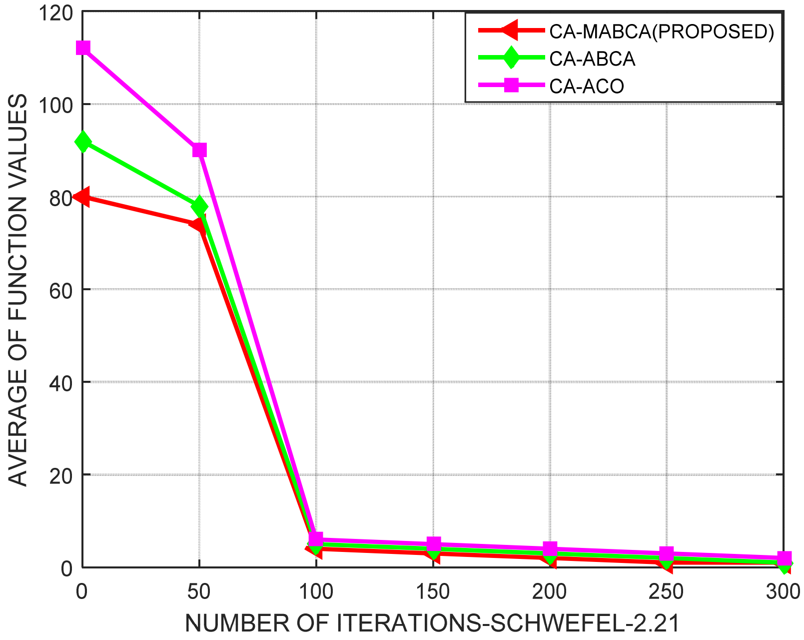


Figure 6.19 - Performance of CA-MABCA- Schwefel-2.21 Function

6.12 Performance analyses of CA-MABCA based on Sumpower function

In the experiment, the superior performance of CA-MABCA and its potentiality is analyzed based on multi-modal benchmark function namely Sumpower under the search dimension of D=20 by varying the number of iterations of search with respect to average function values.

Figure 6.20 shows that CA-MABCA confirm almost a similar kind of performance to CA-ABCA and CA-ACO during its evaluation facilitated with Sumpower multi-modal function. But CA-MABCA is seen to exhibit better performance in a gradual manner than CA-ABCA and CA-ACO as it uses the benefits of chaotic system and opposition-based learning for steepening the optimal convergence rate of finding optimal solution. CA-MABCA is found to facilitate a better convergence rate after 280 iterations than CA-ABCA and CA-ACO.

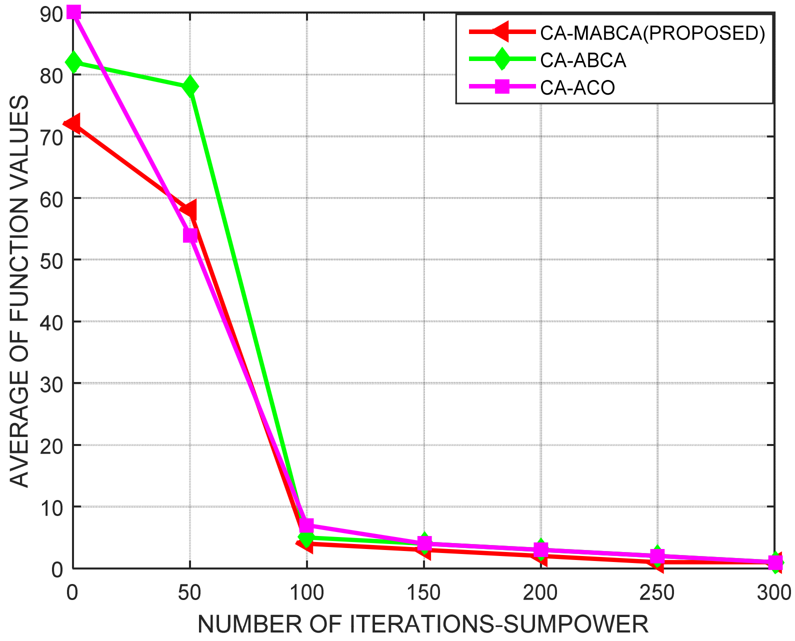


Figure 6.20 - Performance of CA-MABCA- Sumpower Function

6.13 Performance analyses of CA-MABCA based on Elliptic function

In the experiment, the superior performance of CA-MABCA and its potentiality is analyzed based on multi-modal benchmark function namely Elliptic under the search dimension of D=20 by varying the number of iterations of search with respect to average function values.

Figure 6.21 shows that CA-MABCA confirm almost a similar kind of performance to CA-ABCA and CA-ACO during its evaluation facilitated with Elliptic multi-modal function. But CA-MABCA is seen to exhibit better performance in a gradual manner than CA-ABCA and CA-ACO as it uses the benefits of chaotic system and opposition-based learning for steepening the optimal convergence rate of finding optimal solution. CA-MABCA is found to facilitate a better convergence rate after 280 iterations than CA-ABCA and CA-ACO.

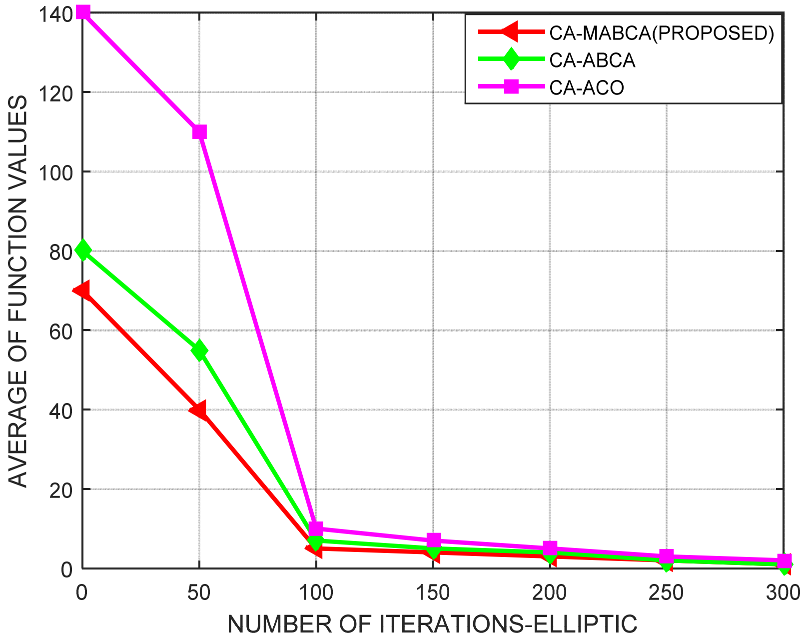


Figure 6.21 - Performance of CA-MABCA- Elliptic Function

6.14 Performance analyses of CA-MABCA based on Exponential function

In the experiment, the superior performance of CA-MABCA and its potentiality is analyzed based on multi-modal benchmark function namely Exponential under the search dimension of D=20 by varying the number of iterations of search with respect to average function values.

Figure 6.22 shows that CA-MABCA confirm almost a similar kind of performance to CA-ABCA and CA-ACO during its evaluation facilitated with Exponential multi-modal function. But CA-MABCA is seen to exhibit better performance in a gradual manner than CA-ABCA and CA-ACO as it uses the benefits of chaotic system and opposition-based learning for steepening the optimal convergence rate of finding optimal solution. CA-MABCA is found to facilitate a better convergence rate after 280 iterations than CA-ABCA and CA-ACO.

It is confirmed that CA-MABCA under elliptic function analysis performs better in terms of average multi-modal functions at 270 iterations in comparison to the Sphere, Step, Schwefel-2.21, Sumpower and Exponential functions. Thus CA-MABCA is predominant in its performance as it achieves global point of optimization at a rapid convergence rate of 21% with elliptic function than the compared multi-modal functions.

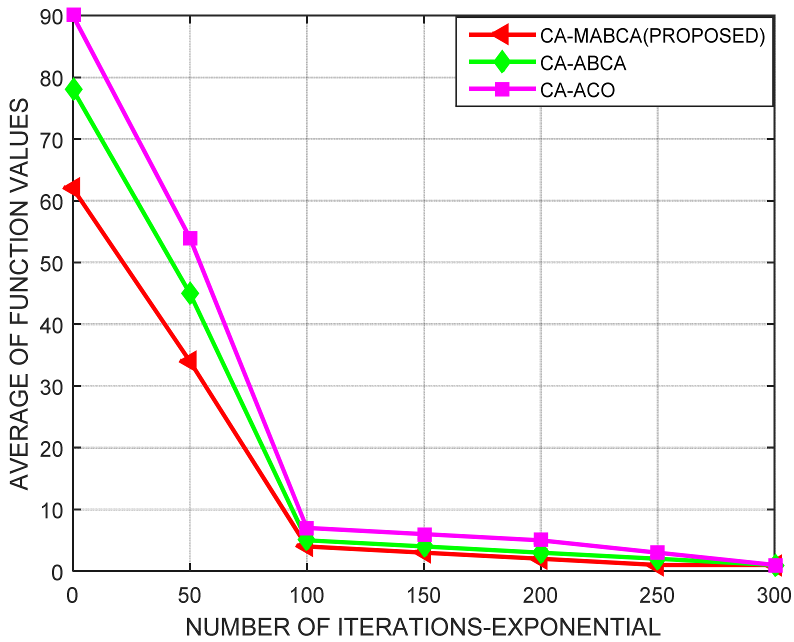


Figure 6.22 - Performance of CA-MABCA- Exponential Function

6.15 Summary

The proposed algorithms such as MACOA-CA, CA-IABCA and MABCA are validated based on selected benchmark multimodal functions to measure their performance in terms of efficiency and reliability of the algorithm. The proposed MACOA-CA is validated based on the multimodal functions Quartic, Schwefel-2.26, Exponential and Sumsquare functions and is found that MACOA-CA reaches the optimal rate of convergence at a faster rate than the other existing algorithms due to the enhanced global searching capability of the algorithm. The CA-IABCA algorithm performance is checked based on the multimodal functions Rastrigin, Ackley, Griewank and Shaffer and is found that the algorithm has better convergence rate than the existing algorithms. Similarly, MABCA is also validated against the multimodal functions and results have shown better performance compared to the existing algorithms.

Chapter 7

Conclusions and Future Research Directions

This chapter concludes the thesis with the research contributions and summary of the work carried out. It highlights the effectiveness and performance of the improved meta-heuristics algorithm integrated with Cellular Automata proposed for preventing DDoS attack by ensuring the neighbor availability in VANETs. This chapter also gives an idea about the future research directions that can be carried out to enhance the neighbor availability in VANETs.

7.1 Conclusion

In VANETs the cooperation among the neighbors is needed for proper vehicle communication to enable safe driving on the road. The neighbor availability is the main security requirement in VANETs for sustained operation of VANET services in ITS. Many attacks threaten the availability of VANETs resources; DDoS attack is a severe form of attack that threatens the availability of the network in a drastic way. From the survey carried out it reveals that DDoS attacks are difficult to be avoided. Many works focused on detecting the DDoS attack or induced other form of DDoS attack affecting the performance of the entire VANETs system.

The proposed works focus on enhancing the availability of the vehicular nodes for V-V communication to prevent DDoS attack due to neighbor unreachability. This thesis details the four improved meta-heuristics approaches based on Cellular Automata to avoid DDoS attack in VANETs by finding reachable and available neighbor.

The proposed algorithms aim to enhance the availability of the neighboring nodes in VANETs for continuous communication among the vehicular nodes. The availability of the nodes is found by applying the ACA, ABC, TS meta-heuristics algorithms since they have the good tendency of finding the global solution for large problems in a reasonable time. Combining the unique features of the algorithms with fast manipulation tendency of the CA, the meta-heuristics algorithms find the neighbor nodes in short duration with good prediction accuracy. Based on this concept four algorithms were proposed. They are:-

- MACOA-CA improved by dynamic evaporation rule and adaptive pheromone update rule achieves better exploration and exploitation. The exploration of ant colony algorithm was improved by adapting dynamic and adaptive pheromone rule for evaporation and updation successively. CA had been integrated with this improved ACA to give the global solution in less duration.

- CA-IABCA improved by using Grenade Explosion method in onlooker bee phase for exploitation and Cauchy Operator in scout bee Phase for exploration gives good solution. The improved ABC algorithm along with the CA neighboring model executes the algorithm in a fast manner to find the best neighbor for data dissemination.

- MABCA improves DDoS handling techniques by applying Differential Evolution on onlooker bee phase and Chaotic learning strategy in scout bee phase. The good random strategy of the algorithm enhances the searching capability and gives good solution.

- CA-MTSA improves TS by Gwet's kappa reliability factor and Bayes conditional probability to handle DDoS. The reliable and reference factor of the vehicles are used to find the good neighbor from the CA neighbor model.

From the result analysis it is found that the proposed algorithms based on meta-heuristics and CA achieved the following result in finding the availability of the neighbor vehicles when compared to the existing algorithms.

a) less prediction delay,

b) less prediction variance and

c) less average prediction delay and variance.

Testing the proposed algorithms against the Benchmark functions also proved that the performance of the improved algorithms is good.

7.2 Contributions

The research work is carried out to enhance the availability of the vehicular node in VANETs for enabling undisrupted data dissemination services to the vehicles on the road. The contributions of the research works are briefed as follows.

- The research paves a path for identifying the significance of meta-heuristic optimization techniques for mitigating DDoS attacks in VANET where emergency data dissemination remains an indispensable part.
- The MACOA-CA helps in predicting the cooperative vehicular nodes for forwarding the data packets successfully with the increased percentage of 15% in a faster rate. The algorithm is applicable in dense environment to help the vehicles escape from DDoS attack.
- For handling the DDoS in highway environment the proposed algorithms CA-IABCA and MABCA are applicable with their improved exploration level at a faster rate. The prediction variance produced by these algorithms is at an increased percentage of up to 16% for highway scenario.
- The CA-MTSA enhanced the degree of cooperation of the vehicles to a considerable level by 12% . The proposed algorithm paves the path to identify the compromised node in a simple and efficient way.
- The research enables the formulation of a generic framework that alternatively and suitably employs any one of the proposed meta-heuristic optimization techniques for DDoS mitigation for improving the rapidity and effectiveness of detection rate.
- The research also emphasizes on how far the detection rate of DDoS attacks can be improved for providing seamless service to the potential users and needy under the context of VANETs.
- The research also portrays the potential of each multi-modal benchmark function over the other for proving the possibility of best fit enabled by each of the proposed mitigation techniques.

7.3 Future Research Directions

The future possibilities of this research work can be extended in the below mentioned ways. They are:

a) Binary Modified artificial bee and multi-parameter inspired Modified bee colony optimization scheme inspired may be formulated to mitigate the DDoS attack for enhancing the availability of the vehicles in VANET.

b) A generic framework may be formulated for handling DDoS attack which alternately chooses any one of the proposed meta-heuristics inspired mitigation mechanisms based on the input parameters and constraints imposed on mitigation.

c) The proposed schemes are being planned to be deployed in other types of cellular model which are equally potent to Moore model in order to analyze the performance for understanding their constraint, superiority and context of application with their suitability.

REFERENCES

[1] Zeadally, Sherali, et al. "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems,* vol.50, no.4, pp.217-241, 2012.

[2] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, Alejandro Quintero. "VANET security surveys," *Computer Communications*, vol.44, pp.1-13, 2014.

[3] M.S. Al-kahtani. "Survey on security attacks in vehicular ad hoc networks (VANETs)," *International Conference on Signal Processing and Communication Systems (ICSPCS)*, IEEE, pp.1–9, 2012.

[4] Hasbullah, Halabi, and Irshad Ahmed Soomro. "Denial of service (DOS) attack and its possible solutions in VANET," *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* , vol.4, no.5, pp. 813-817, 2010.

[5] "IEEE. 1609.2: Trial-use standard for wireless access in vehicular environments-security services for applications and management messages," *IEEE Standards*, 2006.

[6] David R. Raymond and Scott F. Midkiff. "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses," *IEEE Pervasive Computing*, vol.7, no.1, pp. 74-81, 2008.

[7] Mejri, Mohamed Nidhal, Jalel Ben-Othman, and Mohamed Hamdi. "Survey on VANET security challenges and possible cryptographic solutions," *Vehicular Communications* , vol.1, no.2, pp.53-66, 2014.

[8] Li, Yunxin Jeff. "An overview of the DSRC/WAVE technology," *In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Springer, pp.544-558, 2010.

[9] Amit Dua, Neeraj Kumar, and Seema Bawa. "A systematic review on routing protocols for Vehicular Ad Hoc Networks," *Vehicular Communications*, vol.1, no.1, pp.33–52, 2014.

[10] Wang, Zhou, and Chunxiao Chigan. "Cooperation enhancement for message transmission in VANETs," *Wireless Personal Communications*, vol.43, no.1, pp.141-156, 2007.

[11] Michiardi, Pietro, and Refik Molva. "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," *Advanced communications and multimedia security*, Springer, pp.107-121, 2002.

[12] Buchegger, Sonja, and Jean-Yves Le Boudec. "Performance analysis of the CONFIDANT protocol," *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM, pp.226-236, 2002.

[13] Dotzer, Florian, Lars Fischer, and Przemyslaw Magiera. "Vars: A vehicle ad-hoc network reputation system," *International Symposium on a World of Wireless Mobile and Multimedia Networks,* IEEE, pp.454-456, 2005.

[14] Marti, Sergio, et al. "Mitigating routing misbehavior in mobile ad hoc networks," *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, pp.255-265, 2000.

[15] Bansal, Pooja, Shabnam Sharma, and Aditya Prakash. "A Novel approach for Detection of Distributed Denial of Service attack in VANET," *International Journal of Computer Applications,* vol.120, no.5 ,2015.

[16] Chen Lyu, Dawu Gu, Yunze Zeng, and Prasant Mohapatra. "PBA:Prediction-based Authentication for Vehicle-to-Vehicle Communications," *IEEE Transactions on Dependable and Secure Computing*, vol.13, no.1, pp.71-83, 2016.

[17] Biswas, Subir, Jelena Misic, and Vojislav Misic. "DDoS attack on WAVE-enabled VANET through synchronization," *In Global Communications Conference (GLOBECOM),* IEEE, pp.1079-1084, 2012.

[18] Priyadharshini, V., and K. Kuppusamy. "Prevention of DDOS Attacks using New Cracking Algorithm," *International Journal of Engineering Research and Applications*, vol.2 no.3, pp.2263-2267, 2012.

[19] Sumra, Irshad Ahmed, Halabi Bin Hasbullah, and Jamalul-lail Bin AbManan. "Attacks on security goals (confidentiality, integrity, availability) in VANET: a survey," *Vehicular Ad-Hoc Networks for Smart Cities*. Springer Singapore, pp.51-61, 2015.

[20] Schiff, J. L. " Introduction to cellular automata," http://psoup.math.wisc.edu/pub/Schiff_CAbook.pdf, pp.79-81, 2005.

[21] Wolfram, Stephen. "*Theory and applications of cellular automata*," *World scientific*, vol.1, 1986.

[22] Fattahi, M.J., Abad, H., Salari, S.M. and Saadatjoo, M.A. “A new method to improve the particle swarm optimization using cellular learning automata,” *International Symposium on Computational Intelligence and Informatics*, Budapest,Hungary, *IEEE,* pp.247-252, 2011.

[23] Shohei, Sato, and Hitoshi Kanoh. “Evolutionary Design of Rule-Changing Cellular Automata guided by Parameter indicating Propagation of Information,” *Japan Society for Fuzzy Theory and Intelligent Informatics*, *SCIS & ISIS*, pp.1737-1742, 2008.

[24] Tzedakis, Georgios, et al. “The importance of neighborhood scheme selection in agent-based tumor growth modeling,” *Cancer informatics* , vol.14, Suppl 4, pp.67, 2015.

[25] Davies, C. H. J. “The effect of neighborhood on the kinetics of a cellular automaton recrystallisation model,” *Scripta metallurgica et materialia*, vol.33, no.7, pp.1139-1143, 1995.

[26] Xu, H. L., X. Qian, and L. Zhang. “Study of ACO algorithm optimization based on improved tent chaotic mapping,” *Journal of Information and Computational Science*, vol.9, no.6, pp.1653-1660, 2012.

[27] Voss, Stefan, et al., “*Meta-heuristics: Advances and trends in local search paradigms for optimization*,” *Springer Science & Business Media,* 2012.

[28] Osman, Ibrahim H., and James P. Kelly. “Meta-heuristics: an overview,” *Meta-heuristics*. Springer, pp.1-21, 1996.

[29] Tonguz OK, Viriyasitavat W, Bai F. “Modeling urban traffic: a cellular automata approach,” *IEEE Communications Magazine*, vol.47, no.5, 2009.

[30] Sanguesa JA, Fogue M, Garrido P, Martinez FJ, Cano JC, and Calafate CT. “A survey and comparative study of broadcast warning message dissemination schemes for VANETs,” *Mobile Information Systems*, 2016.

[31] Cheng ST, Horng GJ, and Chou CL. “Using cellular automata to form car society in vehicular ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol.12, no.4, pp.1374-84, 2011.

[32] Rabbani, Amirhosein, Hossein Aghababaee, and Mohammad A. Rajabi. “Modeling dynamic urban growth using hybrid cellular automata and particle swarm optimization,” *Journal of Applied Remote Sensing*, vol.6, no.1, pp.063582-1, 2012.

[33] Cheng ST, Chen YJ, Horng GJ, Wang CH. " Using Cellular Automata to Reduce Congestion for Tourist Navigation Systems in Mobile Environments," *Wireless personal communications*, vol.73, no.3, pp.441-61, 2013.

[34] Kumaravel A. "Cellular Automata Construction for Novel Routing over Hexagonal Grid for Mobile Ad-hoc Networks," Indian *Journal of Science and Technology,* vol.8, no.31, 2015.

[35] Amirhosein, R., Hossein, A. and Rajabi, M.A. "Modeling dynamic urban growth using hybrid cellular automata and particle swarm optimization," *Journal of Applied Remote Sensing*, vol. 6, no. 1, pp.063582-1–063582-10, 2012.

[36] Yuntao Dai, Liqiang Liu, Ying Li and Jingyi Song. "An improved particle swarm optimisation based on cellular automata," *International Journal of Computing Science and Mathematics*, vol.5, no.1, pp.94-106, 2014.

[37] Marler, R. Timothy, and Jasbir S. Arora. " Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol.26, no.6, pp.369-395, 2012.

[38] S. Misra, V. Tiwari and M.S. Obaidat. "LACAS: Learning automata-based congestion avoidance scheme for healthcare wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol.27, no.4, pp.466–479, 2009.

[39] Aghababa, A. B., Fathinavid, A., Salari, A., and Zavareh, S. E. "A novel approach for malicious nodes detection in ad-hoc networks based on cellular learning automata," *World Congress on Information and Communication Technologies*, vol.3, no.2, pp.34-43, 2012.

[40] Kim YS, DeBruhl B, Tague P. "Stochastic optimization of flow-jamming attacks in multichannel wireless networks," *International Conference in Communications (ICC), IEEE,* pp. 2165-2170, 2013.

[41] Ozturk C, Karaboga D, Gorkemli B. "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm," *Sensors*. 3, vol.11, no.6, pp.6056-65, 2011.

[42] Santhosh KM, Isaac E. "Defending DDoS Attack using Stochastic Model based Puzzle Controller," *International Journal of Computer Science and Network Security (IJCSNS)*, vol.13, no.4, pp.100, 2013.

[43] Afshar MH. “Extension of the constrained particle swarm optimization algorithm to optimal operation of multi-reservoirs system,” *Electrical Power and Energy Systems*, vol. 51, no. 8,pp.71–81, 2013.

[44] Kokash N. “An introduction to heuristic algorithms,” *Department of Informatics and Telecommunications*, pp.1-8, 2005.

[45] Blum C, Roli A. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Computing Surveys (CSUR*), vol.35, no.3, pp.268-308, 2003.

[46] Osman, Ibrahim H., and Gilbert Laporte. “Metaheuristics: A bibliography,” pp.511-623, 1996.

[47] BoussaïD I, Lepagnot J, and Siarry P. “A survey on optimization metaheuristics,” *Information Sciences*, vol.237, pp.82-117,2013.

[48] Yang XS. “Engineering optimization: an introduction with metaheuristic applications,” *John Wiley & Sons*, 2010.

[49] Desale S, Rasool A, Andhale S, Rane P. “Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey,” *International Journal of Computer Engineering in Research Trends*, vol.2, no.5, pp.296-304, 2015.

[50] Chen HH, Huang SK. “LDDoS Attack Detection by Using Ant Colony Optimization Algorithms,” *Journal of Information Science & Engineering*, vol.32, no.4, 2016.

[51] Vangili A, Thangadurai K. “Detection of black hole attack in mobile ad-hoc networks using ant colony optimization–simulation analysis,” *Indian Journal of Science and Technology*, vol.8, no.13, 2015.

[52] Anuar, Syahid, Ali Selamat, and Roselina Sallehuddin. “A modified scout bee for artificial bee colony algorithm and its performance on optimization problems,” *Journal of King Saud University-Computer and Information Sciences*, vol.28, no.4 pp.395-406, 2016.

[53] Liquan, Zhao, Wang Xin, and Wang Lin. “A novel artificial bee colony algorithm for numerical function optimization,” *International Conference on Control Science and Systems Engineering (ICCSSE),* 2016.

[54] Zheng, Jian-Guo, Chao-Qun Zhang, and Yong-Quan Zhou. “Artificial Bee Colony Algorithm Combined with Grenade Explosion Method and Cauchy Operator for Global Optimization.” *Mathematical Problems in Engineering* 2015.

[55] Zhang, Song, and Sanyang Liu. “A novel artificial bee colony algorithm for function optimization,” *Mathematical Problems in Engineering*, 2015.

[56] Alam, Mohammad Shafiul, Md Monirul Islam, and Kazuyuki Murase. “Artificial bee colony algorithm with improved explorations for numerical function optimization,” *International Conference on Intelligent Data Engineering and Automated Learning*, Springer Berlin Heidelberg, 2012.

[57] Fekair, Mohammed El Amine, Abderrahmane Lakas, and Ahmed Korichi. “CBQoS-Vanet: Cluster-based artificial bee colony algorithm for QoS routing protocol in VANET,” *IEEE International Conference on Mobile & Wireless Networking (MoWNeT),* 2016.

[58] Zhang, Xiu, Xin Zhang, and Cheng Gu. “A micro-artificial bee colony based multicast routing in vehicular ad hoc networks,” *Ad Hoc Networks,* 2016.

[59] Ait Sahed, Oussama, et al. “An efficient artificial bee colony algorithm with application to nonlinear predictive control,” *International Journal of General Systems* , vol.45, no.4, pp.393-417, 2016.

[60] Sulaiman, Noorazliza, Junita Mohamad-Saleh, and Abdul Ghani Abro. “New enhanced Artificial Bee Colony (JA-ABC5) algorithm with application for reactive power optimization,” *The Scientific World Journal* 2015, 2015.

[61] Gao, Wei-feng, and San-yang Liu. “A modified artificial bee colony algorithm,” *Computers & Operations Research* , vol.39, no.3, pp.687-697, 2012.

[62] Y. Xu, P. Fan, and L. Yuan. “A simple and efficient artificial bee colony algorithm,” *Mathematical Problems in Engineering*, vol.9,2013.

[63] Kong, Xiangyu, Sanyang Liu, and Zhen Wang. “An improved artificial Bee Colony algorithm and its application,” *International Journal of Signal Processing, Image Processing and Pattern Recognition* , vol.6, no.6, pp.259-274, 2013.

[64] Bansal, Jagdish Chand, et al. “Balanced artificial bee colony algorithm,” *International Journal of Artificial Intelligence and Soft Computing*, vol.3, no.3, pp.222-243,2013.

[65] Li, Guoqiang, Peifeng Niu, and Xingjun Xiao. “Development and investigation of efficient artificial bee colony algorithm for numerical function optimization,” *Applied soft computing* ,vol.12, no.1, pp.320-332, 2012.

[66] Gao, Weifeng, and Sanyang Liu. “Improved artificial bee colony algorithm for global optimization,” *Information Processing Letters*, vol.111, no.17, pp.871-882, 2011.

[67] Kuang, Fangjun, et al. “A novel chaotic artificial bee colony algorithm based on tent map,” *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014.

[68] Ozturk, Celal, Emrah Hancer, and Dervis Karaboga. “Dynamic clustering with improved binary artificial bee colony algorithm,” *Applied Soft Computing* , vol.28, pp.69-80, 2015.

[69] Back T, “Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms”, *Oxford university press*; 1996 .

[70] Back, Thomas, and Hans-Paul Schwefel. “An overview of evolutionary algorithms for parameter optimization,” *Evolutionary computation*, vol.1, no.1, pp.1-23, 1993.

[71] Fogel DB. “An introduction to simulated evolutionary optimization,” *IEEE transactions on neural networks*, vol.5 no.1, pp.3-14,1994.

[72] Yao X, Lin G, Liu Y. “An analysis of evolutionary algorithms based on neighborhood and step sizes,” *International Conference on Evolutionary Programming*, pp. 297-307, 1997.

[73] Yao, Xin, and Yong Liu. “Fast Evolutionary Programming,” *Evolutionary Programming* ,vol.3, pp.451-460, 1996.

[74] Kappler C. “Are evolutionary algorithms improved by large mutations?,” *International Conference on Parallel Problem Solving from Nature* , pp.346-355, 1996.

[75] Szu HH, and Hartley RL. “Nonconvex optimization by fast simulated annealing,” *Proceedings of the IEEE*, vol.75, no.11, pp.1538-40, 1987.

[76] Liu J, Xu W, and Sun J. “Quantum-behaved particle swarm optimization with mutation operator,” *International Conference in Tools with Artificial Intelligence, ICTAI 05*, pp. 4, 2005.

[77] Yao X, Liu Y, and Lin G. “Evolutionary programming made faster,” *IEEE Transactions on Evolutionary computation*, vol.3, no.2, pp.82-102, 1999.

[78] Li Y, Jiao L, Li P, Wu B. "A hybrid memetic algorithm for global optimization," *Neuro computing*, vol.134, pp.132-9, 2014.

[79] Zhang L, Tang Y, Hua C, and Guan X. "A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques," *Applied Soft Computing*, vol.28, pp.138-49, 2015.

[80] Dong F, Liu D, Wu J, Ke L, Song C, Wang H, and Zhu Z. "Constructing core backbone network based on survivability of power grid," *International Journal of Electrical Power & Energy Systems*, vol.67, pp.161-7, 2015.

[81] Bianchi, Leonora, Marco Dorigo, Luca Maria Gambardella, Walter J. Gutjahr. "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing: an international journal*, vol.8, no.2, pp.239–287, 2009.

[82] Duan, Ping, and A. I. Yong. " Research on an Improved Ant Colony Optimization Algorithm and its Application," *International Journal of Hybrid Information*, vol.9, no.4, pp.223-234, 2016.

[83] Liu, Yunheng. "Research on the Algorithm Optimization of Improved Ant Colony Algorithm-LSACA," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol.9, no.3, pp.143-154, 2016.

[84] Xu, H. L., X. Qian, and L. Zhang. "Study of ACO algorithm optimization based on improved tent chaotic mapping," *Journal of Information and Computational Science* ,vol.9, no.6 ,pp.1653-1660, 2012.

[85] Rana, Himani, Parimala Thulasiraman, and Ruppa K. Thulasiram. "MAZACORNET: Mobility aware zone based ant colony optimization routing for VANET," *Evolutionary Computation (CEC), IEEE Congress on*, pp. 2948-2955, 2013.

[86] Zhang, J., et al. "An Improved Ant Colony Algorithm Based Dynamic Scheduling Method in Job Shop with Parallel Machines," *Advanced Materials Research*, vol. 628, 2013.

[87] Yang, Jingan, and Yanbin Zhuang. "An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem," *Applied Soft Computing*, vol.10, no.2, pp.653-660, 2010.

[88] Sagban, Rafid, Ku Ruhana Ku-Mahamud, and Muhamad Shahbani Abu Bakar, "ACOustic: A nature-inspired exploration indicator for ant colony optimization," *The Scientific World Journal*, 2015.

[89] Tuba, Milan, and Raka Jovanovic. "Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem," *International Journal of Computers Communications & Control*, *vol.*8, no.3, pp.477-485, 2013.

[90] Aljanaby, Alaa, Ku Ruhana Ku-Mahamud, and Norita Md Norwawi. "Interacted Multiple Ant Colonies Optimization Framework: an Experimental Study of the Evaluation and the Exploration Techniques to Control the Search Stagnation," *Int. J. Adv. Comp. Techn.,* vol.2, no.1 ,pp.78-85, 2010.

[91] Hlaing, Zar Chi Su Su, and May Aye Khine. "Solving traveling salesman problem by using improved ant colony optimization algorithm," *International Journal of Information and Education Technology* , vol.1,no.5, pp.404, 2011.

[92] Lee, SeungGwan, and SangHyeok An. "Ant Colony Optimization Algorithm using Back-tracing and Diversification Strategy," *Proceedings of the World Congress on Engineering*, vol. 1, 2016.

[93] Yue, Yimeng, and Xin Wang. "An Improved Ant Colony Optimization Algorithm for Solving TSP," *International Journal of Multimedia and Ubiquitous Engineering*, vol.10, no.12, 2015.

[94] Lei, Wenli, and Fubao Wang. "Research on an Improved Ant Colony Optimization Algorithm for Solving Traveling Salesmen Problem," International *Journal of Database Theory and Application*, vol.9, no.9, pp.25-36, 2016.

[95] Liu, Yue, and Xiaoting Wang. "Study on an Improved ACO Algorithm Based on Multi-Strategy in Solving Function Problem," *International Journal of Database Theory and Application*, vol.8, no.5, pp.223-232, 2015.

[96] Chen, Changai, and Yanwen Xu. "An Improved Quantum Ant Colony Optimization Algorithm for Solving Complex Function Problems," *International Journal of Multimedia and Ubiquitous Engineering*, vol.10, no.11, pp.193-204, 2015.

[97] Zhao, Jiang, Dingding Cheng, and Chongqing Hao. "An Improved Ant Colony Algorithm for Solving the Path Planning Problem of the Omnidirectional Mobile Vehicle," *Mathematical Problems in Engineering*, 2016.

[98] Zhao N, Wu Z, Zhao Y, and Quan T. "Ant colony optimization algorithm with mutation mechanism and its applications," *Expert Systems with Applications*, vol.37, no.7, pp.4805-10, 2010.

[99] Shokouhifar M, Sabet S. "PMACO: A pheromone-mutation based ant colony optimization for traveling salesman problem," *Innovations in Intelligent Systems and Applications (INISTA)*, *International Symposium* , pp.1-5, 2012.

[100] Panigrahi BK, Das S, Suganthan PN, and Nanda PK. "Swarm, Evolutionary, and Memetic Computing," I*nternational Conference SEMCCO*, 2014.

[101] Oliveto PS, Lehre PK, and Neumann F. "Theoretical analysis of rank-based mutation-combining exploration and exploitation," *Evolutionary Computation, CEC'09. IEEE*, pp. 1455-1462, 2009.

[102] Xiang, Wan-li et al. "An Enhanced Differential Evolution Algorithm Based on Multiple Mutation Strategies," *Computational Intelligence and Neuroscience* , pp.285730, 2017.

[103] Mohsen AM, "Annealing Ant Colony Optimization with Mutation Operator for Solving TSP, " *Computational Intelligence and Neuroscience*, 2016.

[104] Glover F, Kelly J. P., and Laguna M. "Genetic Algorithms and Tabu Search: Hybrids for Optimization," *Computers and Operations Research*, vol. 22, no. 1, pp. 111 – 134, 1995.

[105] Glover Fred and Manuel Laguna. " Tabu Search," *Springer* New York, 2013.

[106] Glover, Fred. "*Tabu search fundamentals and uses*, " *Boulder: Graduate School of Business, University of Colorado*, pp.80309, 1995.

[107] Ghaboosi, Nejla, and Abolfazl T. Haghighat. "Tabu search based algorithms for bandwidth-delay-constrained least-cost multicast routing," *Telecommunication Systems*,vol.34, no. 3-4, pp.147, 2007.

[108] Semchedine, Fouzi, Louiza Bouallouche-Medjkoune, Leila Bennacer, Naim Aber, and Djamil Aïssani. "Routing protocol based on Tabu search for wireless sensor networks," *Wireless Personal Communications*, pp.1-8, 2012.

[109] Jang, Kil-Woong. "A tabu search algorithm for routing optimization in mobile ad-hoc networks," *Telecommunication Systems,* vol.51, no.2, pp.177-191, 2012.

[110] Cheng, Hui, and Shengxiang Yang. "Joint multicast routing and channel assignment in multiradio multichannel wireless mesh networks using tabu search," *2009 Fifth International Conference on Natural Computation*, vol. 4, 2009.

[111] Dahiya, Pragati, and Rahul Johari. "VAST: Volume adaptive searching technique for optimized routing in mobile ad-hoc networks," *Advance Computing Conference (IACC), 2014 IEEE International*. IEEE, 2014.

[112] Du, Lingling, and Ruhan He. "Combining nearest neighbor search with tabu search for large-scale vehicle routing problem," *Physics Procedia* , vol.25, pp.1536-1546, 2012.

[113] Moridi, Elham, and Hamid Barati. "RMRPTS: a reliable multi-level routing protocol with tabu search in VANET," *Telecommunication Systems*, vol.65, no.1, pp.127-137, 2017.

[114] Barbarosoglu, Gulay, and Demet Ozgur. "A tabu search algorithm for the vehicle routing problem," *Computers & Operations Research*, vol.26, no.3, pp.255-270, 1999.

[115] Jia, Hongmei, et al. "An improved tabu search approach to vehicle routing problem," *Procedia-Social and Behavioral Sciences*, vol.96, pp.1208-1217, 2013.

[116] Orojloo H, and Haghighat AT. "A Tabu search based routing algorithm for wireless sensor networks, " Wireless Networks, vol.22 no.5, pp.1711-24, 2016.

[117] Taherkhani N, and Pierre S. "Improving dynamic and distributed congestion control in vehicular ad hoc networks," *Ad Hoc Networks*, vol.33, pp.112-25, 2015.

[118] Taherkhani N, and Pierre S. "Congestion control in vehicular ad hoc networks using meta-heuristic techniques," *International symposium on Design and analysis of intelligent vehicular networks and applications*, ACM, pp. 47-54, 2012.

[119] Urquiza-Aguiar, Luis, Carolina Tripp-Barba, and Mónica Aguilar Igartua. "A Geographical Heuristic Routing Protocol for VANETs," *Sensors*, vol.16, no.10, pp.1567, 2016.

[120] Gurav AA, and Nene MJ. "Multiple Optimal Path Identification using Ant Colony Optimization in Wireless Sensor Network," *International Journal of Wireless and Mobile Networks*, vol.5, no.5, pp.119, 2013.

[121] Baskaran R, Victer Paul P and Dhavachelvan P. "Ant colony optimization for data cache technique in MANET," *International Conference on Advances in Computing* , pp. 873-878, 2013..

[122] Ruperez Canas, D., Sandoval Orozco, A. L., Garcia Villalba, L. J., and Hong, P. S. "Hybrid ACO routing protocol for mobile Ad hoc networks," *International Journal of Distributed Sensor Networks*, vol.9, no.5, pp.265485, 2013.

[123] Sharvani GS, Ananth AG, and Rangaswamy TM. "Efficient stagnation avoidance for manets with local repair strategy using ant colony optimization," *International Journal of Distributed and Parallel Systems*, vol.3, no.5, pp.123, 2012.

[124] B. Basturk, and D. Karaboga. "An artificial bee colony (ABC) algorithm for numeric function optimization," *IEEE Swarm Intelligence Symposium* 2006, Indianapolis, 2006.

[125] Karaboga D, and Basturk B. "On the performance of artificial bee colony (ABC) algorithm," *Applied soft computing*, vol.8, no.1,pp. 687-97, 2008.

[126] Ahrari A, Shariat-Panahi M, Atai AA. "GEM: a novel evolutionary optimization method with improved neighborhood search," *Applied Mathematics and Computation*, vol.210, no.2, pp.376-86, 2009.

[127] Ahrari, Ali, and Reza Ahrari. "On the utility of randomly generated functions for performance evaluation of evolutionary algorithms," *Optimization letters*, vol. 4, no.4, pp.531-541, 2010.

[128] Coello, Carlos A. Coello, Arturo Hernandez Aguirre, and Eckart Zitzler. "Evolutionary multi-criterion optimization," *Springer-Verlag Berlin/Heidelberg*, 2005.

[129] Gao, W., & Liu, S. "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol.111, no.17, pp.871-882, 2011.

[130] Rao, R., Savsani, V., and Vakharia, D. "Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems," *Information Sciences*, vol.183, no.1, pp.1-15, 2012.

[131] Rahmani, R., and Yusof, R. "A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: Radial Movement Optimization," *Applied Mathematics and Computation*, vol.248, pp-287-300, 2014.

[132] Huang, Yangcheng, Saleem N. Bhatti, and Daryl Parker, "Tuning olsr," *International Symposium on Personal, Indoor and Mobile Radio Communications*, *IEEE,* pp. 1-5, 2006.

[133] Garcia-Nieto, J., and Alba, E . "Automatic parameter tuning with metaheuristics of the AODV routing protocol for vehicular ad-hoc networks," *Conference on the Applications of Evolutionary Computation*, Springer Berlin, pp. 21-30, 2010.

[134] Santos, H. G., Ochi, L. S., and Souza, M. J. "A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem," *Journal of Experimental Algorithmic (JEA)*, vol.10, pp.2-9, 2005.

LIST OF PUBLICATIONS

International Conferences

1. Thilak, K. Deepa, and A. Amuthan. “DoS attack on VANET routing and possible defending solutions-A survey,” *International Conference on Information Communication and Embedded Systems (ICICES),IEEE, 2016.*

2. Amuthan, A., and K. Deepa Thilak. “Survey on Tabu Search Meta-Heuristic Optimization,” *International Conference on Signal Processing, Communication Power & Embedded System SCOPES, IEEE, 2016.*

3. K. Deepa Thilak, and Amuthan.A. “Survey on Artificial Bee Colony Algorithm for balancing exploitation and exploration,” International Conference on Contemporary Topics in Power Engineering and Aiding Technologies (ICCPEAT2017)", February, 2017.

4. Amuthan.A and K. Deepa Thilak. “Improved Ant Colony Algorithms for Eliminating Stagnation and Local Optimum Problem- A Survey,” International Conference on Technical Advancements in Computers and Communication, April 2017.

Journals

1. A Amuthan, and K Deepa Thilak. “Improved Tabu Search based Cellular Automaton Inspired Algorithm for DDoS attacks in VANETs,” International Journal of Modern Electronics and Communication Engineering, Vol. 5, no. 5, pp-25-34, 2017.

2. A Amuthan, and K Deepa Thilak. “An Improved Stochastic Optimization Algorithm based on Cellular Automata for mitigating DDoS attacks in VANETs,” Journal of Advanced Research in Dynamical and Control Systems, Vol.9, No.14, pp-155-177, 2017.

3. K Deepa Thilak, and A Amuthan. “Cellular Automata-based Improved Ant Colony Optimization Algorithm for mitigating DDoS attacks in VANETs,” *Future Generation Computer Systems,* 2017. (Accepted in Press)

4. K Deepa Thilak, and A Amuthan. “Cellular Automata-based Modified Artificial Bee Colony Algorithm for DDoS attack mitigation in VANETs,” *Egyptian Informatics Journal*, 2017. (Accepted with Minor Revision in Press)

VITAE

K. DEEPA THILAK, the author of this thesis is a Research Scholar in the Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India. She received B.Tech. degree in Information Technology and M.E degree in Computer Science and Engineering from Anna University, Chennai, in the year 2005 and 2010 respectively. She has published more than six research papers in reputed International Journals and Conferences. Her areas of interest include Vehicular Ad-hoc Networks, Mobile Ad hoc Network and Information Security.