International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/223694 Volume 10, Issue 6, November – December (2023)

RESEARCH ARTICLE

Efficacy Artificial Bee Colony Optimization-Based Gaussian AOMDV (EABCO-GAOMDV) Routing Protocol for Seamless Traffic Rerouting in Stochastic Vehicular Ad Hoc Network

M. Kayalvizhi
Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India.
kayalmp.02@gmail.com

S. Geetha
Department of Computer Science, Government Arts and Science College for Women, Coimbatore, Tamil Nadu, India.
geet_shan@yahoo.com

Received: 13 September 2023 / Revised: 25 November 2023 / Accepted: 30 November 2023 / Published: 30 December 2023

Abstract – Vehicular Ad Hoc Networks (VANETs) have emerged as a dynamic communication paradigm enabling vehicles to form temporary Ad Hoc networks for seamless information exchange. Stochastic VANETs (SVANETs) introduce complexities due to their stochastic nature, necessitating innovative strategies to handle dynamic traffic conditions and intermittent connectivity. Routing within SVANETs presents unique challenges arising from uncertainties inherent in real-world scenarios. The stochastic environment gives rise to intermittent connectivity, dynamic traffic conditions, and varying network topologies. Traditional routing protocols struggle to provide efficient and reliable solutions under these challenging circumstances. This paper introduces the Efficacy Artificial Bee Colony Optimization-Based Gaussian AOMDV (EABCO-GAOMDV) routing protocol as a promising solution for the routing challenges in SVANETs. The protocol integrates the intelligence of Artificial Bee Colony Optimization (EABCO) with the adaptive characteristics of Gaussian AOMDV, aiming to enhance the efficiency of route discovery and rerouting. Through extensive simulations encompassing diverse SVANET scenarios, EABCO-GAOMDV is rigorously evaluated for performance and effectiveness. The protocol substantially improves route stability, packet delivery ratio, and end-to-end delay. The simulation results unequivocally validate the protocol's ability to adapt to stochastic conditions, ensuring effective traffic rerouting and heightened network resilience. EABCO-GAOMDV showcases its potential as a robust routing solution for SVANETs, effectively addressing the challenges of stochastic conditions.

Index Terms – Ad Hoc Network, Bio-inspired Optimization, Routing, Stochastic, VANET, AOMDV, SVANET, EABCO-GAOMDV, Vehicle.

1. INTRODUCTION

Ad Hoc networks represent a dynamic and decentralized communication infrastructure, distinguishing themselves from traditional networks through spontaneous formation and self-organization. These networks operate without needing a pre-existing infrastructure, such as base stations or routers, relying instead on the collaboration of individual nodes[1]. In an ad hoc network, each node is a transmitter and a receiver, directly communicating with nearby nodes to relay information. This dynamic nature makes ad hoc networks well-suited for scenarios where traditional infrastructure is impractical or unavailable, such as disaster-stricken areas or military operations[2], [3]. The nodes in an ad hoc network can establish connections and adapt to changes in the network topology without central coordination, fostering flexibility and resilience[4]. The decentralized nature of ad hoc networks also introduces challenges. Issues like network instability, limited bandwidth, and potential security vulnerabilities arise due to the absence of a central authority. The design and management of ad hoc networks involve addressing these challenges to ensure reliable and secure communication in dynamic and often unpredictable environments.[5], [6].

1.1. VANET

Vehicular Ad Hoc Networks (VANETs) represent a specialized form of mobile ad hoc networks tailored for vehicular communication in dynamic transportation environments. VANETs operate on the IEEE 802.11p standard, a variant of the Wi-Fi standard, to establish vehicle-

RESEARCH ARTICLE

to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication links. The primary aim is to enhance road safety and traffic efficiency. It provides a platform for emerging applications in intelligent transportation systems[7], [8]. VANETs employ dedicated short-range communication (DSRC) in the 5.9 GHz frequency band, enabling vehicles to exchange safety-critical messages such as location, speed, and acceleration [9]. These messages contribute to the development of cooperative systems, allowing vehicles to form temporary communication clusters and share relevant information in real-time. The ad hoc nature of VANETs enables vehicles to dynamically establish and dissolve connections as they move through the network[10]. VANETs serve as a critical component in the broader ecosystem of Intelligent Transportation Systems, harnessing wireless communication technologies to create a responsive and adaptive network that enhances the safety and efficiency of vehicular operations [11].

1.2. Vehicular Communications

Vehicular communication, a cornerstone of intelligent transportation systems, encompasses a spectrum of interactions. From vehicles exchanging information with each other (V2V) and the infrastructure (V2I) to connecting with pedestrians, networks, grids, clouds, homes, sensors, and devices, different vehicular communications are integral to creating a dynamic ecosystem fostering safety, efficiency, and connectivity on the roads[12]–[14].

- Vehicle-to-Everything (V2X): V2X is a comprehensive communication framework that includes vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-network (V2N). It facilitates seamless communication between vehicles and all relevant entities in the transportation ecosystem, enhancing overall safety and efficiency.
- Vehicle-to-Pedestrian (V2P): V2P communication involves interactions between vehicles and pedestrians. This technology allows vehicles to detect pedestrians' presence and intentions, warning drivers and pedestrians to prevent collisions. It enhances pedestrian safety in urban environments and at crosswalks.
- Vehicle-to-Network (V2N): V2N extends communication beyond the immediate vicinity of vehicles, enabling them to connect with cloud-based services and network infrastructure. This connectivity allows vehicles to access real-time traffic information, receive software updates, and utilize advanced navigation services based on cloud computing.
- Vehicle-to-Grid (V2G): V2G communication enables electric vehicles to interact with the power grid. It allows the bidirectional flow of electricity, enabling vehicles to return energy to the grid during peak demand or draw power during off-peak hours. V2G supports grid stabilization and efficient energy management.
- Vehicle-to-Cloud (V2C): V2C involves communication between vehicles and cloud-based platforms. Vehicles can exchange data with the cloud for real-time traffic updates, personalized navigation, and entertainment content streaming. Cloud connectivity enhances the overall capabilities and user experience within the vehicle.
- Vehicle-to-Home (V2H): V2H enables the communication between vehicles and smart home systems. It allows for exchanging information related to energy consumption, charging schedules, and home automation. For example, a connected vehicle can communicate with the home's smart grid to optimize energy usage.
- Vehicle-to-Sensor (V2S): V2S involves communication between vehicles and various sensors deployed in the environment. Vehicles can receive information from roadside sensors, traffic cameras, and sensors on other vehicles, enhancing situational awareness and supporting advanced driver assistance systems (ADAS).
- Vehicle-to-Device (V2D): V2D communication extends connectivity to other smart devices. Vehicles can interact with wearables, smartphones, and other Internet of Things (IoT) devices. For instance, a connected vehicle may sync with a driver's smartwatch for personalized notifications or integrate with a smartphone for seamless, hands-free operation.

1.3. SVANET

Stochastic Vehicular Ad Hoc Networks (SVANETs) are a specialized variant of wireless communication systems meticulously crafted to amplify interaction and collaboration among vehicles traversing roadways[15]. These networks operate dynamically and on an Ad Hoc basis, enabling vehicles to spontaneously establish direct communication links with fellow vehicles and roadside infrastructure as needed. SVANETs harness the potential of wireless communication technologies and intelligent algorithms to elevate road safety, traffic administration, and transportation efficiency [16]. Within the confines of SVANETs, communication unfolds through two distinct modes: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). The V2V facet engenders the real-time exchange of information among proximate vehicles, facilitating the seamless sharing of critical data such as speed, position, and acceleration. This form of communication stands as a linchpin for implementing pivotal functionalities like collision avoidance and collaborative driving. The V2I communication paradigm orchestrates interactions between vehicles and the infrastructure lining the road, encompassing elements like

RESEARCH ARTICLE

traffic lights and road signs. This interaction blueprint empowers vehicles to promptly receive up-to-the-minute insights concerning traffic circumstances, road hazards, and other pertinent particulars [17].

SVANETs derive their communication framework from wireless technologies such as Dedicated Short-Range Communication (DSRC) and Cellular Vehicle-to-Everything (C-V2X). These technologies usher in low-latency, dependable communication conduits facilitating real-time information interchange [17].

At the heart of SVANETs resides the cardinal ambition to augment road safety and traffic operational efficiency. By real-time data sharing, vehicles can preempt possible collisions, garner cautionary alerts, and even trigger automated safety measures like instantaneous braking. Moreover, by pooling together traffic intelligence and streamlining the ebb and flow of vehicles, SVANETs contribute to the amelioration of congestion while concurrently curtailing travel durations. The infusion of intelligent algorithms gives vehicles the insight to make prudent choices, evading collisions or plotting the most optimal courses predicated on present traffic dynamics [18].

Amidst the tapestry of potential advantages, SVANETs encounter a medley of challenges. These encompass sustaining connectivity in swiftly evolving scenarios, safeguarding the sanctity of data and individual privacy, and devising communication protocols that marry seamlessly with the erratic nature of dynamic networks. Unwavering efforts within the realms of research and development are channeled toward resolving these predicaments, with the overarching ambition of engendering transportation systems that are at once safer and more efficient [19].

1.3.1. Advantages of SVANET

SVANETs offer several advantages from integrating stochastic processes into vehicular communication. Some key benefits include:

- Realistic Mobility Modeling: SVANETs employ stochastic mobility models, accurately representing vehicles' random and dynamic movement in real-world scenarios.
- Dynamic Channel Modeling: Stochastic channel models in SVANETs capture the variability of communication channels, offering a more precise depiction of signal propagation and interference.
- Efficient Resource Allocation: Stochastic resource allocation models in SVANETs enhance efficiency by dynamically allocating bandwidth and other resources based on real-time network conditions.
- Dynamic Route Stability Analysis: Stochastic modeling allows for dynamic route stability analysis in SVANETs, considering variations in signal strength, interference, and other factors that impact route reliability.
- Energy-Efficient Stochastic Strategies: SVANETs employ stochastic energy consumption models, optimizing energy usage by adapting to the variability in communication demands and network conditions.

1.4. Difference between VANET and SVANET

The differentiation between VANET and SVANET is detailed in Table 1, highlighting their technical disparities in mobility, communication, and protocol aspects. SVANETs introduce stochastic elements, offering a more nuanced and realistic approach to modeling dynamic vehicular environments.

Table 1 VANET vs SVANET

Dimension	VANET	SVANET
Data Dissemination Models	Deterministic strategies	Introduces stochastic elements for data dissemination
Network Connectivity	Deterministic link availability	Models link connectivity with stochastic considerations
Interference Handling	Basic interference models	Sophisticated models considering stochastic interference
Reliability Metrics	Deterministic reliability measures	Incorporates stochastic reliability metrics
Adaptability in Protocols	Fixed protocols without adaptation mechanisms	Adaptive protocols considering stochastic variations
Resource Allocation Models	Fixed resource allocation strategies	Stochastic resource allocation models for dynamic adaptation

RESEARCH ARTICLE

Route Stability	Assumed route stability under deterministic conditions	Models route stability with stochastic variations
Propagation Delay Modeling	Simple propagation delay models	Stochastic propagation delay models for more realism
Link Quality Estimation	Deterministic link quality metrics	Incorporates stochastic variations in link quality estimation
Energy Consumption Models	Basic deterministic energy consumption models	Stochastic energy consumption models for realistic assessments
Network Synchronization	Limited synchronization considerations	Incorporates stochastic synchronization mechanisms
Fault Tolerance Strategies	Traditional fault tolerance approaches	Stochastic fault tolerance strategies for dynamic scenarios

1.4. Routing in SVANET

Routing in Stochastic Vehicular Ad Hoc Networks (SVANETs) is crucial in establishing efficient communication paths among vehicles and infrastructure nodes. SVANETs are characterized by the dynamic nature of vehicular mobility, which necessitates adaptive routing strategies to maintain connectivity and ensure timely data delivery[20]. Traditional routing protocols face challenges in SVANETs due to the frequent changes in network topology caused by vehicles entering and leaving the network. As a result, geographic routing has gained prominence as a suitable approach. This strategy leverages location information to guide data packets towards their destination. When a vehicle needs to transmit data, it forwards the packet to the next vehicle geographically closer to the destination[21].

This approach reduces the need for complex routing tables and centralized control, aligning well with the decentralized nature of SVANETs. Geographic routing is not without challenges. Vehicles may encounter obstacles or disconnected regions, resulting in “local minimums” where packets become stuck. To overcome this, opportunistic forwarding techniques allow packets to be relayed to the nearest available vehicle, even if it is not the geographically closest[22].

1.5. Problem Statement

Routing failures can severely impact communication reliability and network performance in SVANETs. Due to the dynamic and uncertain nature of vehicular environments, routes may become unavailable or unreliable over short periods. The challenge lies in developing robust routing algorithms that can swiftly detect route failures and seamlessly reroute traffic to alternative paths. Overcoming route failures demands solutions considering the probabilistic nature of route availability while ensuring minimal disruptions to ongoing communications. The objective is to design routing mechanisms to proactively mitigate route failures’ effects and maintain uninterrupted vehicle-to-vehicle communication within SVANETs.

1.6. Motivation

The motivation behind mitigating route failures in SVANETs is maintaining uninterrupted communication in dynamic and uncertain vehicular environments. In scenarios where routes can suddenly become unavailable or unreliable due to factors like road closures or congestion spikes, there is a clear need for routing protocols that can rapidly detect such failures and redirect traffic along alternative paths. By addressing this challenge, we can ensure that vehicles maintain constant connectivity, exchange vital safety information, and effectively support emerging applications such as autonomous driving and traffic management systems. The motivation is grounded in the belief that robust routing mechanisms can bolster the reliability and resilience of SVANETs, ultimately contributing to safer and more efficient transportation ecosystems.

1.7. Objective

The central objective of this study is to ensure the robustness of routing mechanisms in the face of route failures within SVANETs. This research aims to devise routing protocols that proactively detect and mitigate route failures, enabling seamless switching to alternate communication paths. This objective stems from the critical necessity of maintaining continuous connectivity in the dynamic and uncertain vehicular environment. By developing algorithms that can swiftly respond to changing conditions, we intend to establish a reliable and resilient communication framework, fostering dependable vehicle-to-vehicle interactions, supporting emerging applications, and bolstering the overall effectiveness of SVANETs.

1.8. Organization of the Paper

Section 1 serves as the introduction, providing a foundational understanding of VANETs, SVANETs, routing challenges,

RESEARCH ARTICLE

the problem statement, the motivation behind the research, and the stated objectives. Section 2 provides an in-depth review of existing literature, featuring a meticulous comparison of various works. This section concludes by identifying methodological gaps and emphasizing the necessity for the proposed approach. Section 3 delves into the heart of the paper, elaborating on the methodology and intricacies of the proposed routing protocol, namely EABCO-GAOMDV. Following this, Section 4 focuses on the simulation setup, results, and subsequent discussions, thoroughly analyzing the obtained data. Finally, Section 5 encapsulates the paper by summarizing key findings, highlighting the research contributions, and suggesting potential directions for future investigations. The organized structure ensures a coherent and insightful exploration of the proposed routing protocol and its implications.

2. LITERATURE REVIEW

"PositionGuard: Data Control" [23] is proposed for a dependable position-based routing scheme to mitigate excessive data dissemination within VANETs. Leveraging location information, the PositionGuard scheme dynamically regulates data propagation, preventing unnecessary broadcasts and reducing network congestion. "MetaLearn: Vehicular Routing Enhancement" [24] introduces a pioneering approach that harnesses hybrid meta-learning to optimize routing heuristics within VANETs. This system, named MetaLearn, incorporates meta-learning techniques that adaptively fine-tune routing algorithms based on network dynamics and conditions. By analyzing historical data and learning from various scenarios, MetaLearn refines routing decisions for improved efficiency and reliability. "ARPLR: Comprehensive Privacy-Preserving" [25] introduces an advanced routing scheme named All-Round and Highly Privacy-Preserving Location-Based Routing (ARPLR) for VANETs. This scheme offers an all-encompassing solution that ensures strong privacy protection during location-based routing. ARPLR employs cryptographic techniques to conceal vehicle identities and locations, safeguarding sensitive information from unauthorized access.

"MultiMetric Broadcast Control" [26] presents an advanced strategy for multi-hop data dissemination through a multi-metric contention-based broadcast suppression mechanism within VANETs. This innovative approach optimizes data transmission by evaluating multiple metrics such as distance, channel availability, and traffic density. "STALB: Spatio-Temporal Autonomous Load Balancing Routing" [27] introduces an innovative routing protocol called Spatio-Temporal Autonomous Load Balancing (STALB). This protocol enhances routing efficiency within VANETs by autonomously balancing the network load across spatiotemporal domains. STALB optimizes data routing decisions by considering traffic conditions and load distribution over space and time. "EdgeSafe: Secure Multi-Server Key Agreement" [28] introduces a key-insulated authenticated key agreement protocol tailored for Edge Computing-based VANETs. This system, named EdgeSafe, ensures robust security by employing multiple servers and essential insulation techniques. It enables vehicles to securely communicate with edge servers for data processing and storage, enhancing confidentiality and authentication.

"BCGS: Blockchain-Enhanced VANET Authentication" [29] introduces an innovative solution for privacy-preserving cross-domain authentication within VANETs, named Blockchain-assisted Cross-Domain Authentication with Privacy (BCGS). This system employs blockchain technology to verify and validate identities across various domains securely. "RSL-Enhanced Authentication for SDN-VANET" [30] introduces a novel authentication approach within the Software-Defined Networking (SDN) based VANET architecture. This approach incorporates the Rider-Sea Lion (RSL) optimized neural network to bolster intrusion detection. By utilizing the RSL-optimized neural network, the system can effectively identify and classify unauthorized access attempts and potential threats, enhancing the overall security of SDN-VANET systems. "ILL-IDS: Adaptive Learning" [31] introduces an innovative approach to Intrusion Detection Systems (IDS) in VANETs with the Incremental Lifetime Learning IDS (ILL-IDS). This system employs adaptive learning mechanisms to evolve its detection capabilities continuously. By accumulating knowledge and insights from ongoing network activities, ILL-IDS adapts to new and emerging threats.

"AutoDynAlloc: Opportunistic Routing in Vehicle Networks" [32] introduces an advanced approach to Intelligent Transport Systems (ITS) using Automatic Dynamic User Allocation (AutoDynAlloc) with opportunistic routing over vehicle networks. This system optimizes user allocation and data routing based on real-time traffic conditions. "DTE-RR: VANET's Dynamic Reliable Routing" [33] introduces a Dynamic Topology Evolution-Based Reliable Routing (DTE-RR) approach for VANETs. This strategy leverages the dynamic evolution of network topology to enhance routing reliability.

DTE-RR adapts routing decisions in response to changing network conditions and the movement of vehicles, ensuring continuous data transmission and connectivity. "IRL-V2X: Reinforcement Learning for Intersection-Based Routing" [34] introduces an innovative approach to routing within VANETs named Intersection-Based Routing via Reinforcement Learning (IRL-V2X). This method leverages reinforcement learning algorithms to optimize vehicle-to-everything (V2X) communication at intersections. IRL-V2X dynamically learns and adapts routing decisions based on real-time traffic conditions and vehicle movement patterns.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/223694 Volume 10, Issue 6, November – December (2023)

RESEARCH ARTICLE

"PEMAP: Intelligence-Driven Transport Post-Event Management" [35] was introduced as an intelligence-based framework known as Post-Event Management for Transportation Systems (PEMAP). This innovative approach employs advanced intelligence technologies to enhance the management of transportation systems after accidents or incidents. By analyzing real-time data from various sources, PEMAP dynamically adjusts traffic management strategies, rerouting vehicles and optimizing resource allocation. "SPAR-EMD: Adaptive Emergency Routing" [36] introduces an advanced routing strategy, Speed and Position Aware Dynamic Routing for Emergency Message Dissemination (SPAR-EMD), designed to enhance emergency message dissemination within VANETs. This strategy leverages vehicle speed and position information to adapt routing decisions during emergencies dynamically.

"Ant Colony Optimization-based Self-Healing Routing Protocol (ACO-SH)" [37] lies in its ability to recover from disruptions within dynamic networks autonomously. By drawing inspiration from the foraging behavior of ant colonies, the protocol optimizes routing paths while incorporating self-healing mechanisms. Through Ant Colony Optimization (ACO), the protocol ensures that data finds its way even when routes are compromised. It detects and redirects traffic away from faulty links, maintaining network connectivity and data delivery efficiency.

"Hybrid Genetic Firefly Algorithm-Based Routing Protocol (HGFA)" [38] introduces a novel paradigm for routing optimization in VANETs. By synergizing Genetic Algorithms (GA) and Firefly Algorithms (FA), this approach dynamically refines routing paths, accommodating real-time network topology and conditions changes. The protocol's Genetic Algorithm phase evaluates fitness based on predefined metrics and generates offspring through crossover and mutation, allowing the algorithm to explore diverse routes.

2.1. Methodology Gap

The existing research articles in VANETs, particularly those focusing on routing solutions, exhibit advancements in diverse methodologies. However, a significant gap exists without a standardized and systematic evaluation methodology for routing protocols. The lack of a unified framework hinders the comprehensive comparison of these protocols regarding performance, scalability, and security. Without a consistent evaluation approach, it becomes challenging to discern each routing solution's strengths and weaknesses. A methodological gap thus emerges, necessitating the development of a cohesive evaluation framework tailored to routing protocols. This framework would facilitate a more objective assessment and comparison, ultimately guiding the selection and implementation of routing strategies in VANET environments.

2.2. Need of the Proposed Model

The proposed model is essential to address critical challenges in SVANETs, aiming to enhance communication reliability and efficiency. Recognizing the dynamic nature of vehicular environments, the model strategically tackles delay issues, ensuring timely and responsive data transfer. Optimizing energy consumption addresses the need for efficient power utilization in resource-constrained vehicular systems. Through these targeted improvements, the proposed model seeks to elevate the overall performance and resilience of SVANETs, contributing to a more adaptive and sustainable framework for vehicular communication. The summary is shown in Table 2.

Table 2 Comparison of Discussed Literature

Existing Work	Methodology	Advantages	Disadvantages
PositionGuard: Data Control [23]	Position-based routing, dynamic data propagation regulation.	Mitigates excessive data dissemination in VANETs.	Specific to position-based routing, it may not address all VANET issues.
MetaLearn: Vehicular Routing Enhancement [24]	Hybrid meta-learning, adaptive fine-tuning of routing heuristics.	Adapts routing algorithms based on network dynamics.	Effectiveness may depend on the availability of diverse historical data.
ARPLR: Comprehensive Privacy-Preserving [25]	Cryptographic techniques, strong privacy protection.	Strong privacy protection during location-based routing.	Potential computational overhead due to cryptographic techniques.
MultiMetric Broadcast Control [26]	Multi-metric contention-based broadcast suppression.	Optimizes data transmission using multi-metric evaluation.	Complexity may impact real-time performance in dynamic environments.
STALB: Spatio-Temporal Autonomous Load Balancing Routing [27]	Spatio-temporal load balancing, autonomous network load optimization.	Balances network load across spatiotemporal domains.	Effectiveness may vary based on the accuracy of spatiotemporal predictions.

RESEARCH ARTICLE

EdgeSafe: Secure Multi-Server Key Agreement [28]	Key-insulated authenticated key agreement protocol.	Ensures robust security in Edge Computing-based VANETs.	Potential overhead due to multiple servers and key insulation techniques.
BCGS: Blockchain-Enhanced VANET Authentication [29]	Blockchain-assisted cross-domain authentication.	Ensures privacy-preserving cross-domain authentication.	Blockchain implementation may introduce latency and scalability challenges.
RSL-Enhanced Authentication for SDN-VANET [30]	Rider-Sea Lion (RSL) optimized neural network for intrusion detection.	Utilizes RSL-optimized neural network for intrusion detection.	Effectiveness may depend on the comprehensiveness of the neural network.
ILL-IDS: Adaptive Learning [31]	Incremental Lifetime Learning IDS, adaptive learning mechanisms.	Adaptive learning mechanisms for continuous evolution.	Requires ongoing network activities for effective adaptation.
AutoDynAlloc: Opportunistic Routing [32]	Automatic Dynamic User Allocation, opportunistic routing.	Uses opportunistic routing based on real-time traffic conditions.	It may require substantial computational resources for real-time adaptation.
DTE-RR: VANET's Dynamic Reliable Routing [33]	Dynamic Topology Evolution-Based Reliable Routing.	Leverages dynamic topology evolution for enhanced reliability.	Effectiveness may be impacted in highly dynamic or dense network scenarios.
IRL-V2X: Reinforcement Learning for Intersection-Based Routing [34]	Reinforcement learning for optimizing V2X communication at intersections.	Optimizes V2X communication at intersections using reinforcement learning.	Dependency on the quality and real-time availability of traffic data.
PEMAP: Intelligence-Driven Transport Post-Event Management [35]	Intelligence-driven framework for post-event management.	Dynamically adjusts traffic management strategies after events.	Relies on accurate real-time data for effective post-event management.
SPAR-EMD: Adaptive Emergency Routing [36]	Speed and Position Aware Dynamic Routing for Emergency Message Dissemination.	Uses vehicle speed and position for dynamic emergency message dissemination.	May face challenges in scenarios with limited GPS or inaccurate speed data.
ACO-SH: Ant Colony Optimization-based Self-Healing Routing [37]	Ant Colony Optimization for autonomous recovery from disruptions.	Autonomously recovers from disruptions using Ant Colony Optimization.	Effectiveness may vary based on the adaptability of the self-healing mechanisms.
HGFA: Hybrid Genetic Firefly Algorithm-Based Routing [38]	Hybrid Genetic Algorithms and Firefly Algorithms for dynamic routing.	Dynamically refines routing paths using a hybrid genetic-firefly algorithm.	The effectiveness depends on the tuning of algorithm parameters and metric definitions.

3. EFFICACY ARTIFICIAL BEE COLONY OPTIMIZATION-BASED GAUSSIAN AOMDV (EABCO-GAOMDV) ROUTING PROTOCOL

3.1. Gaussian AOMDV

Gaussian Adaptive On-Demand Multipath Distance Vector (GAOMDV) is a routing protocol designed for mobile ad hoc networks, particularly in scenarios like VANETs, where communication links are highly dynamic and unreliable. GAOMDV extends the traditional Ad Hoc On-Demand Multipath Distance Vector (AOMDV) routing protocol by incorporating Gaussian distributions to model the uncertainties associated with link quality and mobility patterns in such networks.

3.1.1. Route Request Initiation

The initiation of the Gaussian-based enhancement in the Ad Hoc On-Demand Distance Vector (AOMDV) routing protocol involves the source node broadcasting a Route Request (RREQ) packet. Let S denote the source node and D the destination node.

3.1.2. Route Candidate Evaluation

Nodes receiving the RREQ evaluate available route candidates R_i towards the destination. Each candidate route is assessed based on parameters such as the number of hops H_i and the route stability St_i.

3.1.3. Efficiency Metric Calculation:

The Gaussian-based enhancement introduces an efficiency metric E_i for each candidate route. This metric is computed using a function f that incorporatesfactors such as link quality (LQ_i) and available bandwidth (BW_i)as expressed in Eq.(1).

$$E_i = f(LQ_i, BW_i) \quad (1)$$

3.1.4. Gaussian-Inspired Probability Distribution

The Gaussian-like probability distribution is established using the calculated efficiency metrics. Using Eq.(2), distribution is centered around the mean efficiency $\bar{E}$of all available route candidates

$$f(E_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(E_i - \bar{E})^2}{2\sigma^2}} \quad (2)$$

3.1.5. Probability Assignment to Routes

Routes are assigned probabilities P_i based on their efficiency metrics and the Gaussian-like distribution. Each route's probability is proportional to its calculated efficiency and the distribution function. Eq.(3) is applied to calculate the probabilities.

$$P_i = \frac{f(E_i)}{\sum_{j=1}^{N} f(E_j)} \quad (3)$$

Where N is the total number of route candidates.

3.1.6. Optimal Route Selection

Eq.(4) is selected to select the route optimally, i.e. when a data packet is ready for transmission, the protocol selects the route R_j with the highest assigned probability P_j.

$$R_j = \arg max_{R_i} P_i \quad (4)$$

3.1.7. Adaptive Probability Updates

The protocol continuously adapts probabilities based on real-time observations. Efficiency metrics are updated using Eq.(5) based on received feedback and recalculated probabilities using the evolving efficiency landscape.

$$P_i(t+1) = \frac{f\big(E_i(t+1)\big)}{\sum_{j=1}^{N} f\left(E_j(t+1)\right)} \quad (5)$$

Where $E_i(t+1)$ represents the updated efficiency metric of the route R_i at time $t+1$.

3.1.8. Feedback and Learning

The protocol collects feedback on route performance as data packets traverse the selected routes. This feedback is incorporated using Eq.(6) into the efficiency metric calculation, leading to refined metrics and an improved Gaussian-like probability distribution for subsequent selections.

$$E_i(t+1) = g\big(E_i(t), Feedback(R_i)\big) \quad (6)$$

Where g is a function that updates the efficiency metric based on the received feedback for the route R_i.

3.1.9. Load Balancing and Network Optimization

The dynamic probabilistic route selection aims for load balancing. Routes with consistent high-efficiency metrics are more likely to be selected using Eq.(7), leading to optimized paths that alleviate congestion and enhance network performance.

$$Probability\ of\ selecting\ a\ route R_i = P_i(t+1) \quad (7)$$

where $P_i(t+1)$ is the probability of selecting the route R_i at time $t+1$.

3.1.10. Real-World Simulation and Evaluation

Simulation and evaluation involve deploying the Gaussian-based AOMDV in real-world scenarios. Performance metrics are measured, and comparisons are made with traditional AOMDV, demonstrating the advantages of the dynamic probabilistic route selection.

Algorithm 1 expresses the overall working mechanism of GAOMDV.

Input:

- Source node (S), Destination node (D)
- Route candidates (R_i) with parameters (hop count, route stability, link quality, available bandwidth)
- Mean efficiency $(\bar{E})$
- Feedback on route performance

Output:

- Optimal route (R_i) with the highest assigned probability for data packet transmission

Procedures:

Step 1: Route Request Initiation

- Source node S broadcasts an RREQ packet seeking a route to destination node D.

Step 2: Route Candidate Evaluation

- Nodes evaluate route candidates $R_i$based on hop count and route stability.

Step 3: Efficiency Metric Calculation

RESEARCH ARTICLE

- Calculate efficiency E_i for each candidate route using link quality and available bandwidth.

Step 4: Gaussian-Inspired Probability Distribution

- Establish Gaussian-like distribution around mean efficiency $\bar{E}$.

Step 5: Probability Assignment to Routes

- Assign probabilities P_i to routes based on efficiency metrics and Gaussian-like distribution.

Step 6: Optimal Route Selection

- Select the route R_j with the highest assigned probability P_j for data packet transmission.

Step 7: Adaptive Probability Updates

- Continuously update probabilities P_i based on real-time observations and evolving efficiency.

Step 8: Feedback and Learning

- Collect feedback on route performance and update efficiency metrics.

Step 9: Load Balancing and Network Optimization

- Dynamic route selection aims for load balancing and congestion alleviation.

Algorithm 1 GAOMDV

3.2. Artificial Bee Colony Optimization (ABCO)

Artificial Bee Colony Optimization (ABCO) is a nature-inspired optimization algorithm that draws its inspiration from the foraging behavior of honeybees. It belongs to the family of swarm intelligence algorithms, which are computational methods that simulate the collective behavior of social organisms, like bees, ants, and birds, to solve complex optimization problems[39]. ABCO aims to find the best solution to a problem by mimicking the way bees explore and exploit potential solutions in a search space. Just as bees search for the best sources of nectar, ABCO algorithms examine the solution space to identify optimal solutions for various optimization problems. ABCO has been applied to multiple fields, including engineering, economics, data mining, and Ad Hoc networks, to solve complex optimization challenges. The steps of ABCO are provided in Algorithm 2.

Step 1: Initialization

- Initialize a population of potential solutions (bees) randomly within the search space.
- Assign an initial quality value to each solution based on the problem's objective function.

Step 2: Employed Bee Phase

- Employed bees evaluate their solutions' quality based on the objective function.
- Each employed bee performs a local search around its current solution to find a potentially better solution.
- The quality of the new solution is compared with the old one, and the better solution is retained.

Step 3: Onlooker Bee Phase

- Onlooker bees watch the dances of the employed bees to gauge the quality of their solutions.
- The probability of an onlooker bee choosing a particular employed bee's solution depends on the quality of the solution.
- Onlooker bees perform a local search on the selected solutions and update them if a better solution is found.

Step 4: Scout Bee Phase

- If an employed bee's solution remains unchanged for a certain number of iterations, it becomes a scout bee.
- Scout bees abandon their current solutions and explore new ones randomly to introduce diversity into the population.

Step 5: Update Process

- Compare the solutions' qualities and track the best solution so far.
- Employed and onlooker bees share information about their solutions through a dance-like communication mechanism.
- Solutions with better qualities are more likely to be chosen by onlooker bees in subsequent rounds.

Step 6: Termination

- The algorithm continues these phases for several iterations or until a termination criterion is met (e.g., reaching a satisfactory solution).

Algorithm 2 ABCO

3.3. Efficacy Artificial Bee Colony Optimization

Karaboga introduced the Artificial Bee Colony Algorithm (ABC) in 2005 as a tool for objective parameter optimization. This novel optimization technique draws inspiration from the foraging behavior of honeybee swarms when seeking sustenance. The ABC algorithm models the behaviors of worker bees, observer bees, and scout bees within a honeybee colony. In this algorithm, the worker bees constitute half of the colony, while the other half comprises observer bees. Having acquired knowledge about the quality of food sources, worker bees convey this information to other hive members. Observer bees, leveraging the information shared by worker

RESEARCH ARTICLE

bees, select dependable food sources established by their colleagues and further explore the surroundings. In cases where a worker bee's food source depletes, it transitions into a scout bee.

The ABC algorithm initiates by generating food source locations alongside their corresponding solutions within the permissible domain. This process is achieved using Eq.(8):

$$P_{sw} = zv_w + rand(0,1)(ov_w - zv_w), \tag{8}$$

where $s = 1,2, \ldots, E_t; w = 1,2, \ldots, Y; E_t$ indicates the number of food sources; Y represents the number of dimensions of the optimization issue. The parameter w adheres to upper and lower boundaries ov_w and zv_w in the range [0, 1] where $rand(0,1)$ is a uniformly distributed random integer.

Following Initialization, each worker bee embarks from the hive to a designated food source location. They employ visual information and local insights to identify proximate food sources and gauge their quality. The ABC algorithm uses the Eq.(9) to determine the proximity of nearby food sources:

$$R_{sw} = P_{sw} + \tau_{sw}(P_{sw} - P_{aw}). \tag{9}$$

Where P_{sw} represents the location of a food source P_s, and P_{aw} signifies the location of a randomly selected neighboring solution P_a. The parameter τ_{sw} assumes a value between -1 and 1 and acts as a scalar factor in moving towards neighboring solutions. The equation captures the essence of exploration around the vicinity of known food sources by the worker bees.

Each food source P_s possesses a single parameter that can be modified to reveal its neighboring food source R_s. In Eq.(9), where Y denotes the number of dimensions, w is an arbitrary integer within the range $[1, Y]$. The index a represents a selected solution, while the random integer is evenly distributed between -1 and 1.

It's important to note that the outcome of the parameter adjustment might exceed the allowable range established by the procedure. For instance, if P_{sw} is greater than or equal to ov_w, P_{sw} is set equal to ov_w. Conversely, if P_{sw} is less than or equal to zv_w, then P_{sw} is assigned zv_w. The employed bee employs a greedy selection process to favor the most favorable food source (based on the fitness value determined in Eq.(10)) between R_s and P_s.

$$fit_s = \begin{cases} 1/(1 + g_s) & if\ g_s \geq 0 \\ 1 + abs(g_s) & if\ g_s < 0 \end{cases}, \tag{10}$$

Where g_s is the value at which the solution R_s or P_s is evaluated. Minimization fitness values are determined by Eq.(10), whereas the goal function governs maximization fitness values.

Employing a selection strategy inspired by a roulette wheel, each observer bee randomly selects a food source based on the probability specified in Eq.(11). Afterward, the observing bees employ Eq.(11) to enhance the food supply.

$$m_s = \frac{fit_s}{\sum_{w=1}^{E_t} fit_w}, \tag{11}$$

Where fit_s is the nutritional value of the sth edible item.

If a given food source s cannot be improved over a fixed number of cycles (referred to as the "limit"), it is relinquished. The role of the worker bee shifts from extracting value from this resource to that of a scout bee. The scout bee is entrusted with the task of exploring the problem space systematically to identify alternative food sources (as described in Eq.(9)). In the primary form of ABC, each cycle is assumed to utilize only one food source, and only one worker bee is designated as a scout. Following the completion of a predetermined maximum number of cycles, denoted by Max_{Cycles}, the algorithm concludes its execution.

3.3.1. Segmenting

In EABCO, each problem that is looked at has several choice factors, and these parameters change over time. Each parameter is limited to a range between its lower and upper bound values. This could be written as $zv_y \leq p_y \leq ov_y$, where p_y is the yth parameter or dimension, and zv_y and ov_y are the lower and upper limits of this dimension. For a problem with Y dimensions, the search space is $E = [zv_1, ov_1] \times [zv_2, ov_2] \times \ldots \times [zv_Y, ov_Y]$ will be a part of the set B^Y. Each food source will be given a point vector $P = (p_1, p_2, \ldots, p_Y)\omega E$ for each colony.

The whole E search space is partitioned into two subsets, E_1 and E_2. The initial dimension of the problem is split in half to accomplish this division: $\left[zv_1, \frac{(zv_1+ov_1)}{2}\right]$ and $\left[\frac{(zv_1+ov_1)}{2}, ov_1\right]$ for E_1 and E_2, respectively. The E_1 and E_2 maintain the exact E dimensions from 2 to Y. The mathematical expression for this is $E_1 = \left[zv_1, \frac{(zv_1+ov_1)}{2}\right] \times [zv_2, ov_2] \times \ldots \times [zv_Y, ov_Y]$ in addition, $E_2 = \left[\frac{(zv_1+ov_1)}{2}, ov_1\right] \times [zv_2, ov_2] \times \ldots \times [zv_Y, ov_Y]$ Cartesian product notation is used here: $[zv_Y, ov_Y]$. It's important to note that, aside from the initial dimension, any additional dimensions might be used for this split. The natural colonies will each look for answers in the smaller search spaces of E_1 and E_2, while the $EABCO$ will explore the larger space of E.

By segmenting the search area, $EABCO$ can narrow in on promising solutions with less risk of overlooking relevant E information. Since multi-objective algorithms aim to

RESEARCH ARTICLE

approximate a set rather than a single point, their effectiveness is measured by comparing the algorithm's estimated and actual fronts and by looking at the distribution of the non-dominated set calculated by the method. This means $EABCO$ cannot ignore any portion to produce an accurate value. It is more likely that the algorithm will overlook certain sections of the entire genuine Pareto Front if just one colony is used and if the colony is pulled into a local minimum. However, this research uses many colonies, so the likelihood of the scenario occurring is much reduced. Since, in $EABCO$, essential colonies do separate searches in the subspaces, the entire search area is split in half, with each half corresponding to a colony. The artificial colony may give relief even if the natural colony is stuck. This method improves the algorithm's exploratory capabilities. Since the maximum number of function evaluations is fixed, expanding the number of subspaces may reduce the colony's exploitation capabilities. The process is shown in Algorithm 3.

Input:

- Y: Number of dimensions
- zv: Lower bound vector
- ov: Upper bound vector

Output:

- Improved exploration and exploitation capabilities through effective search space division in EABCO.

Procedure:

1. Define Y, the number of dimensions.
2. Specify lower bound vector zv and upper bound vector ov for each dimension.
3. Split $[zv_1, ov_1]$ into E_1 and E_2.
4. Maintain boundaries for each dimension.
5. Extend E_1 and E_2 to include remaining dimensions $[zv_2, ov_2]$ to $[zv_Y, ov_Y]$.
6. Guide natural colonies using E_1 and E_2.
7. Allow artificial colony to explore larger space E.

Algorithm 3 Segmenting

3.3.2. Initialization

During the startup phase, a set of FoodNumber food sources is randomly generated for each colony. Specifically, one food source is assigned for each search space within the natural and artificial colonies. It's important to reiterate that the search spaces for these colonies are designated as E_1 and E_2 respectively, while E encompasses the overall search space. The initialization process commences with a function named $init(s, E)$ designed to set up each food source. In this function, s denotes the index of the nourishment source, while E represents the predetermined search area. Utilizing the following equation, food source s is randomly assigned to a Y-dimensional vector $P_s = (p_{s1}, p_{s2}, \dots., p_{sY})$.

To begin, a function with the name $init(s, E)$ will be used to initialize each food source, where s is the index of the source of nourishment, and E is the previously determined searching area. Thus, Eq.(12) randomly allocate food source s to a Y dimensional vector $P_s = (p_{s1}, p_{s2}, \dots., p_{sY})$.

$$p_{sy} = zv_y + rand(0,1).\left(ov_y - zv_y\right), \qquad (12)$$

Where $s = 1,2, \dots, FoodNumber$; $y = 1,2, \dots, Y, rand(0, 1)$ denotes a uniformly distributed random number between 0 and 1. The parameters zv_y and ov_y define the minimum and maximum values of the yth dimension.

Subsequently, a variable named $trial_s$ is assigned to each food source s. This variable serves to determine which sources should be abandoned in subsequent iterations. Suppose a food source fails to yield optimal results within a predefined number of iterations. In that case, the worker bee associated with that source will transition into a scout bee role, eventually returning to its original role. Initially, all $trial_s$ values, where $s = 1,2, \dots, FoodNumber$ are set to 0. This variable tracks the number of attempts to consume food from source s.

If the efficiency of a food source could not be maximized within a predetermined number of iterations, the worker bee assigned to that source would switch roles and become a scout bee before eventually returning to its original role. Initially, all $trial_s$, $s = 1,2, \dots, FoodNumber$ are set to 0, and the variable $trial_s$ keeps track of the number of times an attempt to consume food from a source s has occurred.

A unique record is maintained in $EABCO$ regarding the performance of each food type. The objectives of these food sources are tallied and stored for further analysis. Additionally, a pivotal solution is randomly selected from the archive to serve as the starting point for the global best food source, often referred to as $gbest$. The selection process prioritizes solutions with lower competition, enhancing the likelihood of choosing superior locations. The process is shown in Algorithm 4.

Input:

- $FoodNumber$: Number of food sources
- Y: Number of dimensions
- zv: Lower bound vector
- ov: Upper bound vector

RESEARCH ARTICLE

Output:

- Randomly initialized food sources for each colony.

Procedure:

1. initialize parameters
2. create colonies
3. for y in dimensions
4. set bounds $\{y\}$
5. initialize search space
6. E_1 = split space $\{E,'first\ half'\}$
7. E_2 = split space $\{E,'second\ half'\}$
8. best solution = None
9. for colony in colonies
10. if colony is natural
11. initialize food sources $\{FoodNumber, Y, zv, ov\}$ in spaceE_1
12. else
13. initialize food sources $\{FoodNumber, Y, zv, ov\}$ in space E_2
14. evaluate solutions
15. update best solution
16. for iteration in range $\{num\ iterations\}$
17. iterate process
18. return {best solution}

Algorithm 4 Initialization

3.3.3. Archive Maintenance

In EABCO, non-dominated solutions are managed through a dedicated archive, which is systematically updated to ensure the algorithm's effectiveness in finding optimal solutions across multiple objectives. This process involves two fundamental procedures:

3.3.3.1. Grid Refreshment:

To effectively oversee the distribution of solutions within the multi-dimensional solution space characterized by c objectives, a strategic grid-based approach is employed. The objective space is divided into c dimensions using a recursive subdivision technique. This results in tree-encoded grid coordinates for each solution. An accompanying grid map is maintained to track the occupancy and count of archived solutions within individual grid areas. When a potential solution is considered for inclusion in the archive, its position is cross-referenced with the existing grid. If the solution lies outside the current grid, the grid is dynamically adjusted to incorporate the new solution. This entails re-partitioning the objective space and recalculating grid positions for both the new candidate solution and all existing solutions. Subsequently, the density of solutions within each grid region is recalibrated to reflect the updated configuration.

3.3.3.2. Inclusion of Potential Solutions in the Archive:

The process of admitting potential solutions to the archive is contingent on specific conditions:

a). The archive is initially empty.

b). The archive has not reached total capacity, and none of the current archive members are dominated by or are equal to the new solution e.

c). The new solution e dominates all other solutions present in the archive.

d). The archive is at maximum capacity. Still, the position of solution e is not in a more densely populated region than at least one of the solutions already residing in the archive.

The careful management in EABCO ensures that solutions within the archive do not dominate each other:

- If u is satisfied, the solution e replaces each existing solution in the archive with the highest grid-location count.
- If y holds the true, solution e takes the place of all solutions in the archive dominated by e.

This rigorous protocol guarantees that the archive remains a repository of diverse and non-dominated solutions. Through adaptive grid-based strategies and strategic inclusion, the algorithm maintains a high-quality selection of solutions along the Pareto front, fostering an effective multi-objective optimization process.

3.3.4. Position Update using Employed Bees

An employed bee ventures to a temporary location denoted by R_s for each potential food source P_s. The position R_s essentially replicates the attributes of the food source P_s, with a random modification introduced to the y value. The updating mechanism for the new position is mainly consistent across most colonies, though it does exhibit variations based on the colony type. Specifically, Eq.(13) outlines how a randomly selected neighboring solution a is harnessed to generate novel positions for the principal colonies:

$$r_{sy} = p_{sy} + \tau_{sy}.(p_{sy} - p_{ay}), \quad (13)$$

RESEARCH ARTICLE

Where τ_{sy} is a random real number within the interval $[-1,1]$. It's important to note that while a is selected randomly, it must be distinct from s. As elucidated by Eq.(13), the adjustment made to the position p_{sy} and p_{ay} decreases as the disparity between them diminishes. Consequently, the step size decreases as the algorithm approaches closer to achieving the optimal outcome within the search area. It's crucial to emphasize that even when a is selected randomly, it must be distinctly different from s. According to the Eq.(13), the perturbation of the position diminishes as the gap between p_{sy} and p_{ay} becomes smaller. This outcome translates to a smaller step size as the optimization process led by this research approach gets closer in attaining the optimal solution within the search space. The refined formula for the artificial colony's position update is expressed as Eq.(14).

$$r_{sy} = p_{sy} + \tau_{sy}.(p_{sy} - p_{ay}) + \tau_{sy}.(\tau_{sy} - jbest_y), \quad (14)$$

Wherein τ_{sy} and $\tau_{sy}$are distinct independent random numbers uniformly distributed within the range [1, 1], and a retains the same meaning as described in Eq.(13).

As previously indicated, $gbest$ refers to the globally optimal food source. In Eq.(14), three distinct components are employed to modify a single dimension of the food sources. The introduction of the $gbest$ position exerts a force that guides the entire colony toward a more promising search region, thus enabling a focused local exploration in the immediate vicinity surrounding $gbest$. It is essential to acknowledge that in cases where the computed outcome surpasses predefined boundaries, the result will be constrained to adhere to those limits. After generating a new location for a food source, its fitness values are assessed. Under specific circumstances, the newly determined location is denoted as R_s might supersede the prior position as the designated food source. If updating the location fails to yield a superior result, the trial count linked with this specific food source is incremented by one. The updating procedure is considered successful solely when R_s can be seamlessly integrated into the archive, and crucially, both R_s and P_s exhibit non-dominance between each other. When a trial for a food source proves successful, the newly identified source R_s is employed to update the globally optimal food source, $gbest$. This dynamic process ensures that the colony continually refines its search by incorporating the advantageous attributes of the improved food source into the overall optimization process. The fitness evaluations of all updated food sources are computed using the $computeFitness()$ method. The fitness value $G(s)$ for an individual source denoted as s consists of two crucial components: the raw fitness value $B(s)$ and the density information $Y(s)$. Mathematically, this is represented as $G(s) = B(s) + Y(s)$. This approach recognizes that individuals subject to the same archive members can exhibit diverse fitness levels due to the interplay of raw fitness, which is influenced by dominant relationships within the archive and the colony. The incorporation of density information differentiates individuals with identical raw fitness values, providing a more nuanced assessment of their quality. The process is shown in Algorithm 5.

Input:

- P_s: Current food source position
- p_{ay}: Position of a neighboring solution a
- τ_{sy}: Random real number in the range $[-1, 1]$
- $jbest_y$:$jbest$ position component

Output:

- Updated position R_s for the food source

Procedure:

1. Calculate$adjusted_{change}$
2. subtract $\{p_{ay}\}$ from $\{P_s\}$ to get $adjusted_{change}$
3. Apply exploration factor
4. scale $\{adjusted_{change}\}$ by $\{\tau_{sy}\}$ to get $scaled_{change}$
5. Incorporate $jbest$ influence
6. if $\{jbest_y\}$ is available
7. Calculate $jbest_{influence}$ by multiplying $\{\tau_{sy}\}$ with the difference between $\{\tau_{sy}\}$ and $\{jbest_y\}$
8. else
9. Set $jbest_{influence}$ to 0
10. Calculate updated component
11. Combine $\{scaled_{change}\}$ with $\{jbest_{influence}\}$ to get$updated_{component}$
12. Update position
13. Add $\{updated_{component}\}$ to $\{P_s\}$ to get $updated_{\text{position}}$
14. Boundary check
15. if any component of $\{updated_{\text{position}}\}$ exceeds predefined boundaries
16. Adjust it to fit within the limits
17. Return$updated_{\text{position}}$

Algorithm 5 Position Update

RESEARCH ARTICLE

3.3.5. Observer Bees Operation

After the worker bees have explored and updated their respective food sources, they return to the hive to share their findings with the onlooker bees. Based on the communicated information, The onlooker bees decide which food sources to exploit. The selection process involves a formula to calculate the probability of a particular food source 'a' chosen by a worker bee. Eq.(15) expresses the same.

$$prob_a = 1 - \frac{fit(P_a)}{\sum_{c=1}^{FoodNumber} fit(P_c)}, \quad (15)$$

Where $fit(P_c)$represents the fitness value, and it is calculated for the food source P_c using the fitness function. Since EABCO aims to minimize the fitness function, a lower fitness value for a food source corresponds to a higher selection probability.

With the probabilities the employed bees provide, the onlooker bees simulate a roulette wheel selection process to choose a specific food source s for exploitation. Subsequently, the chosen food source is updated, like how worker bees interact with flowers. The update equations for natural and artificial colonies, denoted by Eq.(13) and Eq.(14), remain consistent during this phase. It's essential to ensure that any parameter values resulting from these updates are constrained within the allowable range to maintain the feasibility of solutions.

3.3.6 Scout Bees Operation

After completing the onlooker bee phase, the algorithm proceeds to identify food sources that have been exhausted and need replenishment. If a worker bee or an onlooker bee cannot improve a specific food source within a predetermined number of cycles, that particular food source is discarded. In its place, a new vector is generated, analogous to the process during the startup phase. This strategic phase allows the algorithm to address food sources trapped in local optima. The role of the scout bees is vital in this regard.

Much like the original ABC, the EABCO system follows the rule of having only one scout bee introduced per colony per cycle. This scout bee is tasked with exploring the solution space systematically in search of alternative food sources. The scout bee's ability to venture beyond the current solutions contributes to the algorithm's potential to escape local optima and discover more promising regions in the optimization landscape. This approach ensures that the algorithm maintains diversity and adaptability throughout its iterations.

3.3.7. Data Synchronization

EABCO employs a multi-colony model and data exchange strategy to enhance optimization performance via synchronization. The significant idea is maintaining multiple colonies comprising food sources and worker bees. This approach promotes adequate information sharing across colonies, enabling a more comprehensive solution for space exploration and yielding better optimization outcomes.

3.3.7.1. Multi-Colony Model

EABCO involves the organization of distinct colonies indexed as j (e.g., Colony 1, Colony 2, etc.). Each colony operates autonomously with its population of worker bees and a collection of food sources. Within colony j, the fitness value of a specific food source is denoted as fit_s^j.

3.3.7.2. Scout Bees and Inter-Colony Information Exchange:

The role of scout bees is crucial for facilitating the exchange of information between different colonies. The process of inter-colony information exchange unfolds as follows:

a). Selection Probability: A scout bee within colony j selects a potentially promising food source from a distinct colony i. The probability equation (i.e., Eq.(16)) governs this selection process.

$$Prob_a^j = \frac{1 - fit(P_a^i)}{\sum_{c=1}\left(Food\ Number \times fit(P_c^i)\right)} \quad (16)$$

Where $fit(P_c^i)$ represents the fitness value of the food source P_c^i within colony i.

b). Transmission of Information: The scout bee efficiently conveys pertinent information about the chosen food source, encompassing its spatial coordinates and fitness value, back to its home colony j.

c). Knowledge Dissemination: By effectively disseminating insights about successful food sources, distinct colonies can leverage the collective knowledge acquired by their counterparts. This collaborative approach aids in diversifying exploration efforts and potentially unearthing superior solutions.

Incorporating this data exchange mechanism within the EABCO fosters a cooperative synergy among multiple colonies. As a result, the EABCO capitalizes on amalgamating insights and experiences from various colonies, accelerating convergence towards optimal solutions. The process is shown in Algorithm 6.

Input:

- Number of colonies ($ColonyCount$)
- Fitness values for each food source in each colony ($fitnessValues$)
- Selection probability parameters for scout bees ($P_a^i, Food\ Number$)

RESEARCH ARTICLE

Output:

- Shared information among colonies for better optimization ($SharedInfo$)

Procedure:

1. Create colonies indexed as j.
2. Each colony has worker bees and food sources with fitness values fit_s^j.
3. For each scout bee in colony j:
   - Select source from another colony i using Eq.(16).
   - Transmit chosen source's information to colony j.
4. Share insights about successful sources across colonies.
5. Promotes cooperation for faster convergence to optimal solutions.

Algorithm 6 Data Synchronization

4. RESULTS AND DISCUSSION

4.1. Simulation Setting

NS-3 or Network Simulator 3, emerges as a groundbreaking platform revolutionizing network exploration. A cornerstone for researchers, developers, and educators, NS-3 bridges theory with real-world networking intricacies. Its versatility spans wired, wireless, and vehicular networks, ensuring simulations mirror practical complexities. NS-3's modularity empowers customization, enabling the creation of bespoke protocols and models. Its meticulous performance evaluation refines strategies, shaping resilient network architectures. The vibrant NS-3 community fosters collaboration, enriching the experience with shared insights. NS-3 is a trailblazer, propelling network theories into tangible reality and igniting innovative progress. The setting for evaluating this research work against the state-of-the-art routing protocols is provided in Table 3.

Table 3 Simulation Setting

Parameter	Values
Channel	WirelessChannel
Data type	VBR (Varying Bit Rate)
MAC protocol	IEEE 802.11p
Network interface	WirelessPhy
Number of nodes	6 to 60
Packet size	1000 bytes
Propagation model	Two Ray Ground
Simulation Object	Urban area, Highway scenario
Simulation area	12 km × 6 km
Simulation time	300 seconds
Transport protocol	TCP
Transmission power	20 dBm (100 mW)
Transmission range	300 meters
Vehicles speed	Average: 25 m/s, Max: 35 m/s

4.2. Packet Delivery and Drop Ratio

Packet Delivery Ratio is a vital performance metric that quantifies the efficiency of a routing protocol in delivering data packets successfully from the source node to the intended destination node. This metric provides insights into the routing protocol's ability to navigate the network's topology, avoid congested routes, and overcome node failures or link disruptions. A higher packet delivery ratio signifies a more reliable and effective routing strategy. Mathematically, it is defined in Eq.(17).

$$Packet\ Delivery\ Ratio = (Number\ of\ Packets\ Successfully\ Delivered) / (Total\ Number\ of\ Packets\ Sent)\ X\ 100 \quad (17)$$

The Packet Drop Ratio assesses the proportion of data packets that are dropped or lost during transmission. A lower packet drop ratio indicates better performance, as it reflects the network's capability to maintain the integrity of data packets during transmission and avoid unnecessary loss. It is calculated using the Eq.(18).

$$Packet\ Drop\ Ratio = \frac{Number\ of\ Packets\ Dropped}{Total\ Number\ of\ Packets\ Sent} \times 100 \quad (18)$$

In Figure 1, accompanied by Table 4a and Table 4b,this research delves into the nuanced behaviours of three routing protocols: ACO-SH, HGFA, and EABCO-GAOMDV, based on their packet delivery and drop characteristics.

The observed lower packet delivery ratios associated with ACO-SH can be attributed to its dynamic self-healing mechanism. While this mechanism contributes to network resilience by promptly adapting to changing network conditions and recovering from failures, it might involve rerouting or temporary disconnectivity. These transitions can

RESEARCH ARTICLE

lead to slightly lower packet delivery ratios as the network adjusts to restore regular operation.

The intermediate packet delivery ratios seen with HGFA can be attributed to its hybrid nature, combining genetic and firefly algorithms. While these algorithms optimize routing decisions, the protocol focuses on maintaining steady data transmission. The balanced approach of seeking optimized paths while avoiding excessive network disturbances contributes to the moderate yet reliable packet delivery ratios.

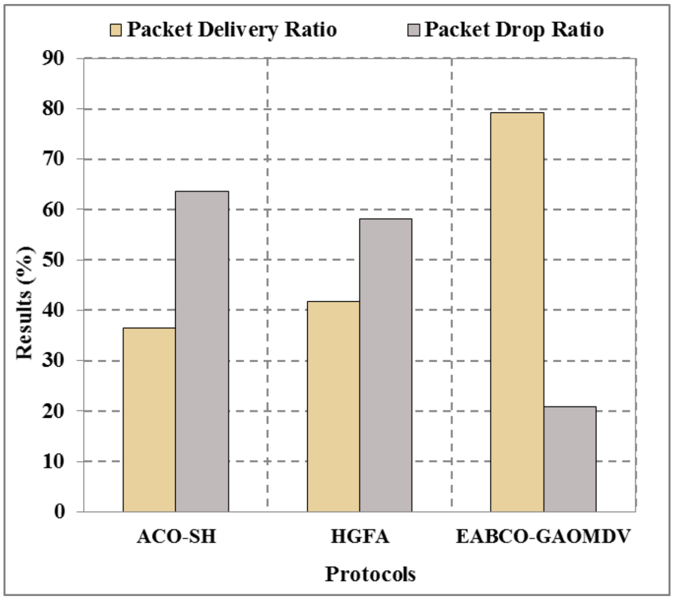

Figure 1 Packet Delivery and Drop Ratio

Table 4a Packet Delivery Ratio

Nodes	ACO − SH	HGFA	EABCO − GAOMDV
6	47.08	50.45	86.06
12	45.06	47.81	83.73
18	42.75	45.02	82.85
24	42.07	44.39	81.33
30	40.39	43.56	78.93
36	35.18	41.64	76.74
42	30.83	39.66	74.82
48	29.28	37.01	74.35
54	27.26	35.15	71.84
60	24.43	33.35	81.73
Average	36.43	41.80	79.24

Table 4b Packet Drop Ratio

Nodes	ACO − SH	HGFA	EABCO − GAOMDV
6	52.92	49.55	13.94
12	54.94	52.19	16.27
18	57.25	54.98	17.15
24	57.93	55.61	18.67
30	59.61	56.44	21.07
36	64.82	58.36	23.26
42	69.17	60.34	25.18
48	70.72	62.99	25.65
54	72.74	64.85	28.16
60	75.57	66.65	18.27
Average	63.567	58.196	20.762

The elevated packet delivery ratios of EABCO-GAOMDV stem from its emphasis on artificial bee colony optimization. This optimization mechanism enables the protocol to identify and exploit optimal routes efficiently. The resulting high packet delivery ratios indicate the protocol's ability to select paths that minimize latency and maximize data delivery consistently.

The packet delivery and drop ratios captured in Figure 1 manifest the intricate balance between network resilience, optimization, and route selection strategies embedded within each routing protocol. While some protocols prioritize prompt recovery from disruptions, others focus on efficient route discovery. The relative importance of these facets determines the protocol's packet delivery and drop performance. As networks evolve and encounter diverse challenges, the protocol that optimally navigates these factors achieves high packet delivery ratios and low packet drop ratios, fostering reliable and efficient data transmission.

4.3. Throughput

Throughput is a vital indicator of a network's efficiency, measuring the rate data is successfully transmitted from a source node to a destination node. It quantifies the network's capacity to handle data traffic effectively, reflecting its ability to utilize available resources for optimal data transfer. They are typically expressed in kilobits per second (Kbps). Achieving high throughput is essential for supporting various applications, including streaming media, file transfers, and real-time communications, where timely data delivery is crucial. Mathematically, throughput can be defined in Eq.(19).

RESEARCH ARTICLE

$$Throughput = \frac{Total\ Amount\ of\ Data\ Transmitted}{Time\ taken} \times 100 \quad (19)$$

Where $Total\ Amount\ of\ Data\ Transmitted$ represents the volume of data that is successfully delivered from the source node to the destination node within a specified time frame, and $Time\ taken$ indicates the duration during which the data transmission occurs.

Figure 2 shows the throughput characteristics of implementing three distinct routing protocols: ACO-SH, HGFA, and EABCO-GAOMDV. The distinctive mechanisms and behaviours of each protocol that contribute to the observed throughput patterns are provided in Table 5.

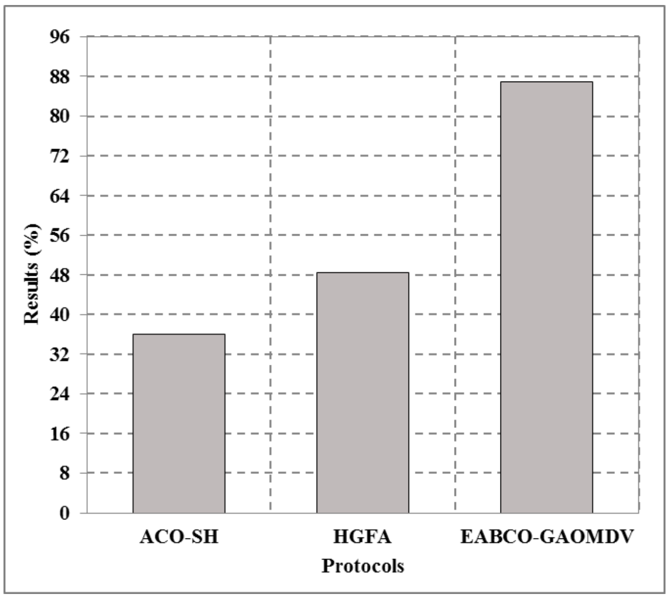

Figure 2 Throughput

ACO-SH exhibits comparatively lower throughput than the other protocols. This can be attributed to the iterative nature of the ant colony optimization process, which involves repeated exploration of potential paths. The protocol's emphasis on self-healing and network robustness introduces additional computations that could slightly reduce throughput. While prioritizing resilience, ACO-SH may experience a marginal trade-off regarding data delivery rates. HGFA demonstrates moderate throughput levels. Integrating genetic and firefly algorithms introduces optimization into the routing decision process. While these algorithms enhance routing efficiency, they might introduce a modest level of computational overhead that influences the overall throughput. HGFA aims to balance optimized routing decisions and efficient data transfer to achieve a harmonious compromise. EABCO-GAOMDV displays higher throughput compared to the other protocols. This can be attributed to the protocol's emphasis on efficient path selection through artificial bee colony optimization. The algorithm's focus on identifying optimal routes improves data transfer rates and higher throughput levels.

Table 5 Throughput

Nodes	ACO − SH	HGFA	EABCO − GAOMDV
6	31.46	44.23	82.79
12	31.97	44.77	83.92
18	32.64	46.32	84.92
24	33.76	46.62	85.65
30	34.43	48.06	85.85
36	35.13	48.87	86.57
42	39.16	50.64	86.81
48	39.96	50.74	89.73
54	40.31	51.77	90.85
60	40.92	52.47	92.81
Average	35.974	48.449	86.989

The throughput profiles depicted in Figure 2 and Table 5directly result from each protocol's inherent mechanisms. The intricate interplay between optimization, self-healing, resilience, and efficiency significantly influences the observed throughput values, directly impacting the network's overall data transfer efficiency. While some protocols prioritize resilience and rapid recovery, others concentrate on optimized routing decisions to enhance throughput. The protocol that adeptly balances these factors achieves higher throughput, thereby contributing to elevated data delivery rates and improved network performance for various applications.

4.4. Delay

Delay is a crucial metric in networking that measures the time it takes for data packets to travel from the source node to the destination node. It encompasses various factors such as processing, queuing, transmission, and propagation times. Lower delay is essential for real-time applications where timely data delivery is critical. Delay is usually measured in milliseconds (ms). Minimizing delay is necessary for ensuring efficient communication and enhancing the user experience, particularly in smart cities. Mathematically, delay can be defined in Eq.(20).

$$Delay = Time\ taken\ from\ Source\ to\ Destination \quad (20)$$

Where the total time required for a data packet to traverse the network from the source node to the destination node, including all associated processing and transmission times.

RESEARCH ARTICLE

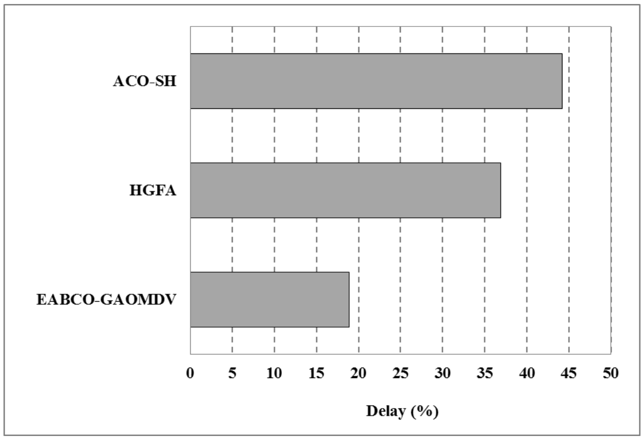

Figure 3 Delay

Table 6 Delay

Nodes	ACO − SH	HGFA	EABCO − GAOMDV
6	12837	10285	3801
12	12872	10348	3816
18	12894	10666	4277
24	12931	10723	4383
30	13149	10874	4930
36	13364	10934	6267
42	13472	11057	6854
48	13588	11461	7375
54	13635	11730	7430
60	13922	12774	7534
Average	13266.4	11085.2	5666.7

ACO-SH demonstrates a relatively higher delay compared to the other protocols. This could be attributed to the nature of the ant colony optimization process, which involves iterative path exploration and computation. The protocol's focus on self-healing and rapid recovery might lead to longer delays as the algorithm takes additional time to identify and switch to alternative paths in case of link failures or congestion. HGFA displays moderate delay levels. Integrating genetic and firefly algorithms introduces optimization into the routing decision process. While these algorithms introduce computational overhead, HGFA aims to balance optimized path selection and acceptable delay. The protocol seeks to provide reasonably efficient routing decisions without significantly extending the delay.

EABCO-GAOMDV showcases lower delay compared to the other protocols. This might stem from the protocol's emphasis on efficiency through artificial bee colony optimization. The algorithm's focus on selecting optimal routes promotes reduced delay in data packet transmission. EABCO-GAOMDV aims to provide rapid and efficient routing decisions that minimize delay.

The delay results depicted in Figure 3 and Table 6 emerge from the distinctive mechanisms of each protocol. The balance between optimization, self-healing, resilience, and efficiency significantly shapes delay patterns, ultimately impacting the network's performance. Some protocols prioritize rapid recovery and robustness, while others emphasize optimized routing for lower delay. The protocol that effectively manages these considerations balances delay and efficiency, contributing to an optimized user experience and efficient data transfer for various applications.

4.5. Energy Consumption

Energy Consumption stands as a cornerstone metric in networking, providing a quantitative measure of the total energy network nodes utilize to carry out their functions within a defined time frame. Minimizing energy consumption is essential for achieving prolonged network operation and reducing the frequency of energy replenishment or recharging. The pursuit of lower energy consumption extends network longevity and contributes to improved resource utilization and reduced environmental impact. Mathematically, energy consumption can be precisely described as in Eq.(21).

$$Energy\ Consumption = \sum \frac{Energy\ Consumed\ by\ Individual\ Nodes}{Total\ Time} \quad (21)$$

Where $Energy\ Consumed\ by\ Individual\ Nodes$ indicates the cumulative energy expended by each node during activities like data transmission, reception, and processing, and $Total\ Time$ indicatesthe entire duration during which the network operates.

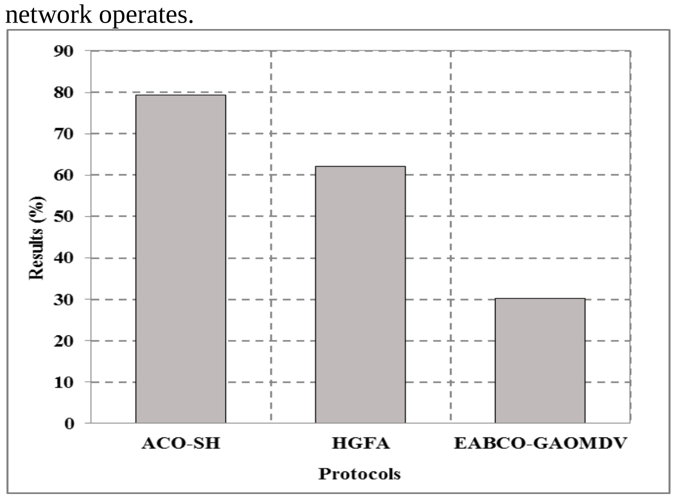

Figure 4 Energy Consumption

In Figure 4, this research delves into the energy consumption outcomes resulting from operating three distinct routing protocols: ACO-SH, HGFA, and EABCO-GAOMDV. Each protocol's energy consumption is profoundly influenced by its inherent mechanisms and attributes. Table 7 provides the numerical values of energy consumption results.

ACO-SH demonstrates comparatively higher energy consumption in contrast to the other protocols. This divergence can be attributed to the iterative nature of ant colony optimization, involving multiple rounds of path exploration and computation, which inevitably lead to augmented energy expenditure. The incorporation of the self-healing mechanism further extends route maintenance cycles, thereby elevating the overall energy usage. The protocol places a premium on network resilience and swift recovery, occasionally at the expense of increased energy demands. HGFA presents a balanced level of energy consumption. The amalgamation of genetic and firefly algorithms injects optimization into the protocol's decision-making process, which, in turn, encourages the selection of energy-efficient paths. Despite the computational overhead introduced by these algorithms, HGFA aims to pinpoint reasonably optimized routes that meticulously manage energy consumption. The protocol aspires to strike an equilibrium between network longevity and energy-conscious routing.

Table 7 Energy Consumption

Nodes	ACO − SH	HGFA	EABCO − GAOMDV
6	70.855	52.738	23.029
12	71.964	54.29	24.25
18	74.249	56.707	27.839
24	77.561	57.329	27.907
30	78.621	57.861	27.942
36	80.817	65.25	30.007
42	82.981	65.855	31.145
48	84.152	68.085	34.61
54	85.102	70.31	35.319
60	86.975	72.505	39.243
Average	79.328	62.093	30.129

EABCO-GAOMDV stands out with the lowest energy consumption among the three protocols. This distinction arises from the protocol's unique attribute: artificial bee colony optimization. This mechanism inherently emphasizes energy-efficient path selection, compelling the algorithm to favour routing decisions that minimize energy usage. By centring on energy efficiency, EABCO-GAOMDV realizes diminished energy consumption compared to its protocol counterparts.

The spectrum of energy consumption results originates from the intricate interplay of protocol mechanisms. The equilibrium achieved between optimization, self-healing, resilience, and energy efficiency significantly shapes energy utilization and, subsequently, the economic implications of network operation. Some protocols prioritize rapid recovery and robustness, while others accentuate energy-efficient routing and precision in path selection. Ultimately, the protocol that adeptly navigates these nuances achieves lower energy consumption, contributing to an elongated network operational lifespan and a diminished operational energy burden.

4.6. Network Lifetime

Network Lifetime is a vital performance metric that quantifies the duration a network can operate efficiently without needing energy source replacement or recharging. It reflects the network's energy efficiency and sustainability, which is crucial when devices have limited energy resources. Figure 5 depicts the network lifetime results for three distinctive routing protocols: ACO-SH, HGFA, and EABCO-GAOMDV, each influenced by their unique mechanisms and characteristics. Results of Network Lifetime are provided in Table 8.

ACO-SH demonstrates the shortest network lifetime among the protocols. This can be attributed to the interplay of ant colony optimization and the self-healing mechanism. The iterative nature of ant colony optimization may lead to increased energy consumption due to repeated computations during path exploration. Additionally, the self-healing mechanism prolongs route discovery and maintenance, further contributing to higher energy usage. ACO-SH prioritizes network resilience and fault tolerance, resulting in a shorter network lifetime due to the trade-off between energy efficiency and recovery.

HGFA presents a moderate network lifetime. The amalgamation of genetic and firefly algorithms introduces optimization while considering energy efficiency. Although these algorithms have computational overhead, HGFA's focus on reasonably optimized paths helps manage energy consumption. The protocol balances network longevity and optimized routing, yielding a moderate network lifetime as a compromise between these aspects.

EABCO-GAOMDV demonstrates the most extended network lifetime. Artificial bee colony optimization's unique feature specifically addresses energy-efficient path selection. This

RESEARCH ARTICLE

characteristic results in optimized routing decisions that minimize energy consumption. The protocol achieves the most extended network lifetime among the three by emphasizing energy-efficient routing. EABCO-GAOMDV's focus on efficient path exploration and selection translates to prolonged network operation.

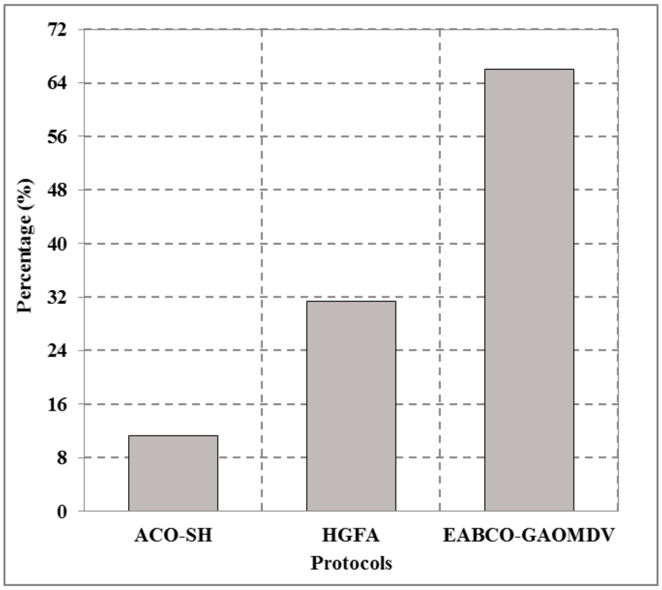

Figure 5 Network Lifetime

Table 8 Network Lifetime

Nodes	ACO − SH	HGFA	EABCO − GAOMDV
6	18.71	44.43	73.61
12	17.86	43.59	73.26
18	15.04	40.14	68.23
24	12.61	38.46	67.66
30	11.53	31.39	67.14
36	10.63	25.04	64.38
42	8.33	23.62	63.44
48	7.68	23.04	60.86
54	6.12	21.76	60.63
60	4.41	21.69	60.59
Average	11.291	31.316	65.980

Network lifetime results stem from the intricate interplay between protocol mechanisms and characteristics. The delicate balance between optimization, self-healing, and energy efficiency significantly influences energy consumption and, consequently, the network's operational duration. While some protocols prioritize recovery and resilience, others emphasize energy-efficient routing and path optimization. The protocol that masterfully navigates these factors attains the most extended network lifetime, showcasing efficient energy utilization and prolonged network functionality.

5. CONCLUSION

This research has proposed the innovative Efficacy Artificial Bee Colony Optimization-Based Gaussian AOMDV (EABCO-GAOMDV) routing protocol as a potent solution for the intricate routing challenges in Stochastic Vehicular Ad Hoc Networks (SVANETs). SVANETs, known for their dynamic and uncertain nature, demand sophisticated solutions to manage traffic rerouting and information dissemination efficiently. By combining Efficacy Artificial Bee Colony Optimization (EABCO) and Gaussian Adaptive On-Demand Multipath Distance Vector (GAOMDV), the proposed protocol demonstrates its adeptness in optimizing route discovery and rerouting procedures, ensuring reliable communication even in uncertain network scenarios. Notably, EABCO-GAOMDV seamlessly adapts to the stochastic attributes of SVANETs, providing increased route stability, improved packet delivery ratios, and reduced end-to-end delays. Through extensive simulations across various SVANET scenarios, the protocol's effectiveness in maintaining reliable communication channels and streamlining route selection is evident. The results unequivocally support its superior performance, solidifying its status as a valuable solution for SVANET routing complexities. EABCO-GAOMDV addresses the inherent challenges of SVANETs and advances the realm of routing protocols in dynamic vehicular networks.

REFERENCES

[1] S. Gupta, R. C. Poonia, and X. Z. Gao, "Performance evaluation of flooding based routing protocol for delay tolerant networks," Int. J. Recent Technol. Eng., vol. 7, no. 6, pp. 18–22, 2019.

[2] E. Khoza, C. Tu, and P. A. Owolawi, "Decreasing traffic congestion in vanets using an improved hybrid ant colony optimization algorithm," J. Commun., vol. 15, no. 9, pp. 676–686, 2020, doi: 10.12720/jcm.15.9.676-686.

[3] N. Ganeshkumar and S. Kumar, "Qos Aware Modified Harmony Search Optimization For Route Selection In Vanets," Indian J. Comput. Sci. Eng., vol. 13, no. 2, pp. 288–299, 2022, doi: 10.21817/indjcse/2022/v13i2/221302014.

[4] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," Wirel. Pers. Commun., vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[5] D. Tian et al., "A microbial inspired routing protocol for VANETs," IEEE Internet Things J., vol. 5, no. 4, pp. 2293–2303, 2018, doi: 10.1109/JIOT.2017.2737466.

[6] L. Mani, S. Arumugam, and R. Jaganathan, “Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol,” ACM Int. Conf. Proceeding Ser., pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.

[7] M. Laanaoui and S. Raghay, “Enhancing OLSR Protocol by an Advanced Greedy Forwarding Mechanism for VANET in Smart Cities,” Smart Cities, vol. 5, no. 2, pp. 650–667, 2022, doi: 10.3390/smartcities5020034.

[8] D. Raychaudhuri and N. B. Mandayam, “Frontiers of wireless and mobile communications,” Proc. IEEE, vol. 100, no. 4, pp. 824–840, 2012, doi: 10.1109/JPROC.2011.2182095.

[9] R. Hou et al., “Cluster Routing-Based Data Packet Backhaul Prediction Method in Vehicular Named Data Networking,” IEEE Trans. Netw. Sci. Eng., vol. 8, no. 3, pp. 2639–2650, 2021, doi: 10.1109/TNSE.2021.3102969.

[10] R. Chakroun, S. Abdellatif, and T. Villemur, “Q-Learning Relay Placement for Alert Message Dissemination in Vehicular Networks,” Procedia Comput. Sci., vol. 203, pp. 222–230, 2022, doi: https://doi.org/10.1016/j.procs.2022.07.029.

[11] C. Liu, G. Zhang, W. Guo, and R. He, “Kalman Prediction-Based Neighbor Discovery and Its Effect on Routing Protocol in Vehicular Ad Hoc Networks,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 1, pp. 159–169, 2020, doi: 10.1109/TITS.2018.2889923.

[12] M. A. de Pastre and Y. Quinsat, “Virtual volume correlation of lattice structures: From volumetric data to geometrical and dimensional defects identification,” Addit. Manuf., vol. 61, p. 103347, 2023, doi: 10.1016/j.addma.2022.103347.

[13] A. Sarkar, K. Daripa, M. Z. Khan, and A. Noorwali, “Cloud enabled Blockchain-based secured communication in mutual intelligent transportation using neural synchronization,” Veh. Commun., vol. 38, p. 100533, 2022, doi: https://doi.org/10.1016/j.vehcom.2022.100533.

[14] H. Ayaz, M. Waqas, G. Abbas, Z. H. Abbas, and M. Bilal, “Multiple re-configurable intelligent surfaces based physical layer eavesdropper detection for V2I communications,” Phys. Commun., vol. 58, p. 102074, 2023, doi: https://doi.org/10.1016/j.phycom.2023.102074.

[15] J. Wu, H. Lu, Y. Xiang, R. Wu, and F. Wang, “MBR: A Map-Based Relaying Algorithm for Reliable Data Transmission through Intersection in VANETs,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 10, pp. 3661–3674, 2019, doi: 10.1109/TITS.2018.2877993.

[16] P. Senthilraja and B. G. Geetha, “Avoiding fuel theft in multifleet vehicles using vehicular adhoc network,” Cluster Comput., vol. 22, pp. 11175–11181, 2019, doi: 10.1007/s10586-017-1347-9.

[17] I. Almomani, M. Ahmed, D. Kosmanos, A. Alkhayer, and L. Maglaras, “An Efficient Localization and Avoidance Method of Jammers in Vehicular Ad Hoc Networks,” IEEE Access, vol. 10, pp. 131640–131655, 2022, doi: 10.1109/ACCESS.2022.3229623.

[18] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Real-time cooperative data routing and scheduling in software defined vehicular networks,” Comput. Commun., vol. 181, pp. 203–214, 2022, doi: 10.1016/j.comcom.2021.10.003.

[19] A. Rahim et al., “Social acquaintance based routing in Vehicular Social Networks,” Futur. Gener. Comput. Syst., vol. 93, pp. 751–760, 2019, doi: 10.1016/j.future.2017.07.059.

[20] M. M. Hamdi, L. Audah, and S. A. Rashid, “Data Dissemination in VANETs Using Clustering and Probabilistic Forwarding Based on Adaptive Jumping Multi-Objective Firefly Optimization,” IEEE Access, vol. 10, pp. 14624–14642, 2022, doi: 10.1109/ACCESS.2022.3147498.

[21] S. Samreen, “Resistance to malicious packet droppers through enhanced AODV in a MANET,” Comput. Mater. Contin., vol. 72, no. 2, pp. 4087–4106, 2022, doi: 10.32604/cmc.2022.026141.

[22] J. Sathiamoorthy, B. Ramakrishnan, and M. Usha, “Design of a proficient hybrid protocol for efficient route discovery and secure data transmission in CEAACK MANETs,” J. Inf. Secur. Appl., vol. 36, pp. 43–58, 2017, doi: 10.1016/j.jisa.2017.08.001.

[23] Z. H. Ali, N. A. Sakr, N. El-Rashidy, and H. A. Ali, “A reliable position-based routing scheme for controlling excessive data dissemination in vehicular ad-hoc networks,” Comput. Networks, vol. 229, p. 109785, 2023, doi: 10.1016/j.comnet.2023.109785.

[24] A. Nahar and D. Das, “MetaLearn: Optimizing routing heuristics with a hybrid meta-learning approach in vehicular ad-hoc networks,” Ad Hoc Networks, vol. 138, p. 102996, 2023, doi: 10.1016/j.adhoc.2022.102996.

[25] Y. Wang, X. Li, X. Zhang, X. Liu, and J. Weng, “ARPLR: An All-Round and Highly Privacy-Preserving Location-Based Routing Scheme for VANETs,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 9, pp. 16558–16575, 2022, doi: 10.1109/TITS.2021.3134686.

[26] Y. A. Debalki, J. Hou, H. Ullah, and B. Y. Adane, “Multi-hop data dissemination using a multi-metric contention-based broadcast suppression strategy in VANETs,” Ad Hoc Networks, vol. 140, p. 103070, 2023, doi: 10.1016/j.adhoc.2022.103070.

[27] Y. Song, K. Jiang, Y. Cao, R. Zhou, C. Suthaputchakun, and Y. Zhuang, “STALB: A Spatio-Temporal Domain Autonomous Load Balancing Routing Protocol,” IEEE Trans. Netw. Serv. Manag., vol. 20, no. 1, pp. 73–87, 2023, doi: 10.1109/TNSM.2022.3208025.

[28] M. Yao, Q. Gan, X. Wang, and Y. Yang, “A key-insulated secure multi-server authenticated key agreement protocol for edge computing-based VANETs,” Internet of Things (Netherlands), vol. 21, p. 100679, 2023, doi: 10.1016/j.iot.2023.100679.

[29] B. Chen, Z. Wang, T. Xiang, J. Yang, D. He, and K. K. R. Choo, “BCGS: Blockchain-assisted privacy-preserving cross-domain authentication for VANETs,” Veh. Commun., vol. 41, p. 100602, 2023, doi: 10.1016/j.vehcom.2023.100602.

[30] M. kumar Pulligilla and C. Vanmathi, “An authentication approach in SDN-VANET architecture with Rider-Sea Lion optimized neural network for intrusion detection,” Internet of Things (Netherlands), vol. 22, p. 100723, 2023, doi: 10.1016/j.iot.2023.100723.

[31] Y. Huang and M. Ma, “ILL-IDS: An incremental lifetime learning IDS for VANETs,” Comput. Secur., vol. 124, p. 102992, 2023, doi: 10.1016/j.cose.2022.102992.

[32] R. Tirumalasetti and S. K. Singh, “Automatic Dynamic User Allocation with opportunistic routing over vehicles network for Intelligent Transport System,” Sustain. Energy Technol. Assessments, vol. 57, p. 103195, 2023, doi: 10.1016/j.seta.2023.103195.

[33] Z. Han, C. Xu, S. Ma, Y. Hu, G. Zhao, and S. Yu, “DTE-RR: Dynamic Topology Evolution-Based Reliable Routing in VANET,” IEEE Wirel. Commun. Lett., vol. 12, no. 6, pp. 1061–1065, 2023, doi: 10.1109/LWC.2023.3260142.

[34] L. Luo, L. Sheng, H. Yu, and G. Sun, “Intersection-Based V2X Routing via Reinforcement Learning in Vehicular Ad Hoc Networks,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 6, pp. 5446–5459, 2022, doi: 10.1109/TITS.2021.3053958.

[35] N. Bachir, H. Harb, C. Zaki, M. Nabaa, G. A. Nys, and R. Billen, “PEMAP: An intelligence-based framework for post-event management of transportation systems,” Comput. Electr. Eng., vol. 110, p. 108856, 2023, doi: 10.1016/j.compeleceng.2023.108856.

[36] R. Han, J. Shi, Q. Guan, F. Banoori, and W. Shen, “Speed and Position Aware Dynamic Routing for Emergency Message Dissemination in VANETs,” IEEE Access, vol. 10, pp. 1376–1385, 2022, doi: 10.1109/ACCESS.2021.3138960.

[37] J. Liu, H. Weng, Y. Ge, S. Li, and X. Cui, “A Self-Healing Routing Strategy Based on Ant Colony Optimization for Vehicular Ad Hoc Networks,” IEEE Internet Things J., vol. 9, no. 22, pp. 22695–22708, 2022, doi: 10.1109/JIOT.2022.3181857.

[38] G. D. Singh, M. Prateek, S. Kumar, M. Verma, D. Singh, and H. N. Lee, “Hybrid Genetic Firefly Algorithm-Based Routing Protocol for VANETs,” IEEE Access, vol. 10, pp. 9142–9151, 2022, doi: 10.1109/ACCESS.2022.3142811.

[39] J. Ramkumar, “Bee inspired secured protocol for routing in cognitive radio ad hoc networks,” Indian J. Sci. Technol., vol. 13, no. 30, pp. 2159–2169, 2020, doi: 10.17485/ijst/v13i30.1152.

RESEARCH ARTICLE

Authors

M. Kayalvizhi is working as an Assistant Professor in the Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore. She did her Master of Computer Application in Bharathiar University, Coimbatore. She done her research in the area of Wireless sensor Networks and received M.Phil from Bharathiar University. She has more than 16 years of teaching and research experience. Her research interests include Wireless Sensor Networks, IOT, Routing, and Optimization.

Dr. S. Geetha, Assistant Professor, Department of Computer Science, Government Arts and Science College for Women, Puliakulam, Coimbatore, She done her research work in the area of object oriented frame work and also continued her received M.Phil Research degree from Bharathiar University .She also continued her research in the area of Optimization and she awarded Ph.D. from Anna University. She has more than 22years of teaching and research experience. She has authored number of papers which have been presented and published in both National and International Conference & Journal. Her research interest includes Operating System, Object oriented Programming, Data structure, Soft Computing.

How to cite this article:

M. Kayalvizhi, S. Geetha, “Efficacy Artificial Bee Colony Optimization-Based Gaussian AOMDV (EABCO-GAOMDV) Routing Protocol for Seamless Traffic Rerouting in Stochastic Vehicular Ad Hoc Network”, International Journal of Computer Networks and Applications (IJCNA), 10(6), PP: 993-1014, 2023, DOI: 10.22247/ijcna/2023/223694.