Multi-Agent Coordination in Autonomous Vehicle Routing: A Simulation-Based Study

KM Khalid Saifullah
Department of Computer Science
The College of Wooster
Wooster, USA
tsaifullah25@gmail.com

***Abstract*—This study explores how multi-agent interaction enhances autonomous vehicle (AV) decision-making in dynamic traffic environments. While traditional AV models focus on individual autonomy, real-world traffic scenarios often require collective behavior through inter-agent communication and coordination. To investigate this, we developed a graph-based simulation environment that enables vehicle agents to exchange information and reroute in real time in response to road obstacles. Our findings demonstrate that communication and adaptive rerouting significantly reduce average wait times and improve travel efficiency. Furthermore, we introduce a lightweight memory mechanism—Object Memory Management (OMM)—which allows agents to retain knowledge of previously encountered obstacles. This feature proved critical in avoiding routing loops and redundant decisions. Together, these results highlight the potential of communication- and memory-enhanced agents in creating resilient, cooperative AV systems capable of navigating complex and unpredictable traffic networks.**

***Index Terms*—Autonomous Vehicles, Multi-Agent Systems, V2V Communication, Dynamic Rerouting, Traffic Simulation, Obstacle Avoidance.**

I. Introduction

Autonomous vehicles (AVs) are poised to revolutionize transportation by promising increased safety, reduced congestion, and enhanced mobility. Modern AVs rely on sophisticated sensors, machine learning algorithms, and decision-making systems to perceive their surroundings and plan efficient routes. However, most current implementations operate under the assumption of individual autonomy, where each vehicle navigates independently. This assumption breaks down in real-world scenarios, where coordinated behavior among vehicles can significantly improve system-level efficiency and safety.

As AV technology advances toward higher levels of automation, as defined in the SAE J3016 standard, the ability for vehicles to share information and collaborate in real-time becomes critical. Multi-agent systems (MAS)—where vehicles act not as isolated units but as part of a coordinated, intelligent network—offer a promising approach to addressing challenges such as dynamic obstacle avoidance, traffic rerouting, and congestion management.

To operate effectively in such dynamic environments, AVs must be capable of both reactive and proactive behaviors. This includes exchanging route and obstacle data (Vehicle-to-Vehicle or V2V communication), responding to infrastructure signals (Vehicle-to-Infrastructure or V2I), and making decisions based not only on current sensor inputs but also on shared environmental knowledge.

This study investigates how communication and decentralized decision-making affect routing performance in multi-agent AV systems. We propose a graph-based simulation framework that enables agents to dynamically reroute, exchange localized information, and adapt to obstacles in real-time. Our research is guided by the following core question: *To what extent does inter-agent communication improve travel efficiency and reduce delay in multi-agent autonomous vehicle systems navigating obstacle-rich environments?*

By introducing controlled complexity into the simulation—such as variable obstacle placements and communication configurations—we evaluate the impact of coordination and rerouting logic under realistic yet repeatable conditions. Our findings contribute to the growing body of research on cooperative autonomy by highlighting the importance of real-time interaction and strategic rerouting in next-generation AV deployments.

II. Background and Related Work

A. *Multi-Agent Systems in Autonomous Driving*

Multi-agent systems (MAS) provide a framework in which multiple AVs operate as intelligent agents interacting within a shared environment. In this context, each vehicle (agent) observes the road network, makes decisions, and communicates with others to achieve individual and collective goals. By coordinating actions, MAS enable cooperative driving behaviors that optimize traffic flow, reduce congestion, and improve safety [13]. This paradigm contrasts with traditional single-agent autonomy, where an AV plans in isolation.

The significant impact of even a single AV on traffic flow has been demonstrated in field experiments, where one self-driving car with an intelligent control policy dissipated stop-and-go waves in human-driven traffic [19]. As the proportion of AVs on the road increases, the potential benefits of multi-agent coordination become even greater. Research in autonomous driving MAS has explored vehicles sharing intentions, routes, and observations to negotiate maneuvers and avoid conflicts. Multi-agent interaction is now viewed as key to decentralized decision-making in traffic networks, shifting the core challenge in fully autonomous traffic from unpredictable human behavior to communication and coordination problems [6].

B. V2V and V2I Communication

For AVs to function as a coordinated MAS, they must exchange information in real-time. This is enabled by Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, collectively known as V2X. V2V allows AVs to form decentralized vehicular ad hoc networks (VANETs) to share data like location, speed, and road hazards. V2I connects vehicles with infrastructure like traffic signals or cloud servers for broader traffic information.

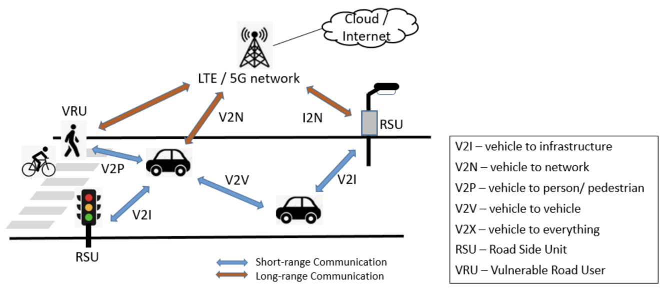

Fig. 1. V2X communication architecture showing interactions among vehicles, infrastructure, and networks [22].

Dedicated Short Range Communications (DSRC), based on the IEEE 802.11p standard, is a pioneering V2V technology offering low latency suitable for safety-critical applications like forward collision warnings [20]. Newer cellular-based V2X (C-V2X) leverages 4G/5G networks, promising extended range but facing challenges in meeting strict latency requirements for safety applications [2]. Regardless of the technology, these protocols enable cooperative perception, where an obstacle detected by one agent is virtually sensed by all agents in communication range, reducing reliance on individual onboard sensors. However, the security and integrity of V2X communication are paramount, as threat actors could spoof messages or jam signals, leading to incorrect routing decisions [15].

C. Distributed Coordination and Decision-Making

Multi-agent coordination can be centralized or decentralized. Centralized approaches, where a single entity computes global solutions, suffer from scalability issues and single points of failure, making them impractical for large AV fleets [6]. Decentralized coordination, where each agent makes its own choices based on local information, is considered more realistic and robust [23]. The challenge lies in ensuring that independent decisions produce desirable global behavior.

Recent research has shown decentralized approaches to be highly effective. Studies have demonstrated that decentralized deep reinforcement learning can eliminate traffic bottlenecks, with each vehicle making decisions based on local observations [23]. Scalable multi-agent driving policies have achieved improved congestion metrics in complex open road networks without a central coordinator [4]. However, purely independent decision-making without communication can lead to inefficient behavior, as each AV's optimal action might conflict with others.

Successful decentralized strategies often incorporate consensus-building or negotiation. In multi-agent reinforcement learning (MARL), a popular paradigm is centralized training with decentralized execution (CTDE), where agents learn cooperative policies offline with full state information but execute them at runtime using only local data [6]. Rule-based distributed coordination has also been explored, such as cooperative rerouting algorithms where vehicles use iterative message passing to converge on a traffic dispersion strategy [21].

A key ingredient in many distributed strategies is memory. Since there is no central database, each vehicle must serve as a sensor and a data relay. Our implementation of a lightweight memory mechanism, Object Memory Management (OMM), allows each vehicle to track obstacles it has learned about. The importance of memory in multi-agent decision-making has been highlighted in literature, as it helps agents avoid repeating mistakes in non-stationary environments [6].

D. Dynamic Routing and Rerouting

In a static environment, route planning reduces to finding an optimal path on a graph using algorithms like Dijkstra's or A*. However, real-world driving is dynamic, with events like accidents or congestion rendering planned routes suboptimal. In a multi-agent context, vehicles can orchestrate path re-planning in real-time through V2V communication.

Coordination ensures that vehicles do not all swarm to the same detour or interfere with each other's new routes. Empirical studies have shown that adaptive vehicle rerouting can mitigate congestion by dynamically redistributing traffic [7]. This can be framed as an optimization problem (balancing loads on alternate routes) or a game-theoretic one (vehicles choosing routes to reach a Nash equilibrium). Roncoli et al. proposed an approach where each vehicle's route choice incorporates a cost reflecting both its own travel time and collective congestion, encouraging a balanced traffic distribution [21].

An important consideration is when to reroute. Constant path switching can lead to system instability. Agents typically employ thresholds; for instance, a vehicle may wait a predetermined time before initiating a reroute. This was implemented in our study, where vehicles wait 8 seconds before self-rerouting.

E. Simulation Platforms

Research on multi-agent AV interaction relies heavily on simulation. High-fidelity 3D simulators like CARLA offer realistic physics and environments but are computationally expensive and have significant hardware requirements [8]. Microscopic traffic simulators like SUMO can handle large road networks but require complex scripting to implement specific multi-agent communication and memory behaviors [14].

Given these constraints and our focus on high-level decision-making logic, we developed a custom, lightweight, graph-based simulator. This approach abstracts away low-level vehicle dynamics to focus solely on routing decisions and waiting times caused by blockages. It provides a clean, deterministic environment to implement and analyze the effects of communication, self-rerouting, and memory. This aligns with the principle of using a model that is "as simple as possible, but no simpler," allowing us to isolate cause and effect.

III. Methodology

To methodically assess multi-agent coordination, we adopted a graph-based simulation environment designed explicitly for evaluating rerouting and inter-vehicle communication effectiveness under controlled, reproducible conditions.

A. Graph-Based Simulation Environment

Our simulation environment models a road network as a directed graph $G = (V, E)$, where nodes $v \in V$ represent intersections and edges $e \in E$ represent road segments. For this study, the graph contains 86 nodes and 161 edges. Vehicles, conceptualized as autonomous agents, navigate this graph from defined start nodes to predetermined destinations. The primary route-planning method employed is Dijkstra's shortest path algorithm [5], chosen for its efficiency and suitability for road networks where edge weights (e.g., travel time, distance) are non-negative.

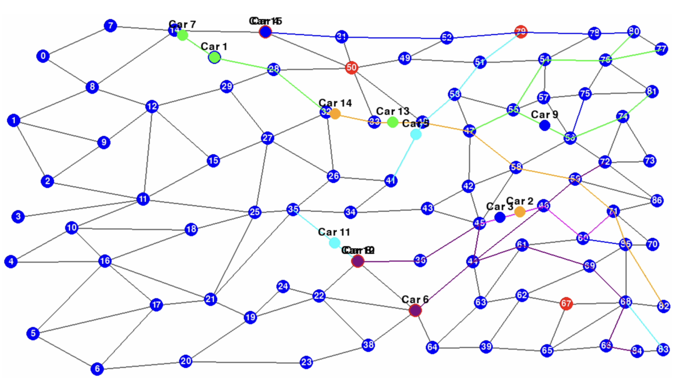

Fig. 2. The graph-based road network used in the simulation, consisting of 86 nodes and 161 edges.

Obstacles can be dynamically placed on any node, temporarily disabling it and requiring vehicles to adapt. Each vehicle agent operates independently, initialized with a randomly assigned origin and destination.

B. Experimental Design

We structured our evaluation around 12 distinct scenarios to investigate the performance implications of varying traffic densities (15, 35, and 55 cars) and obstacle frequencies (6 and 20 obstacles). For each combination, we tested two movement patterns: a structured left-to-right traversal and a randomized origin-destination pairing.

Within each of these 12 scenarios, we tested six distinct agent capability configurations to quantify the benefits of incremental technological advancements:

1) **No Obstacles (Baseline):** A control condition to establish idealized travel time.
2) **Obstacles, No Communication (Worst-Case):** Vehicles encountering an obstacle wait for a fixed duration (10 seconds) for it to clear.
3) **Communication Enabled:** Vehicles broadcast obstacle information upon detection but do not proactively reroute. Waiting vehicles still wait 10 seconds.
4) **Self-Rerouting Enabled:** Vehicles encountering an obstacle wait 8 seconds before autonomously recalculating their routes. Communication is enabled.
5) **Object Memory Management (OMM) Enabled:** Vehicles maintain a memory of broadcasted obstacle locations and use this information to proactively avoid known problematic nodes. Communication is enabled, but self-rerouting (the 8s wait trigger) is disabled.
6) **Self-Rerouting with OMM:** The most advanced configuration, combining communication, an 8-second wait trigger for rerouting, and a memory of known obstacles to inform path recalculation.

This structured approach ensures that variations in performance outcomes can be directly attributed to the specific technologies being evaluated.

C. Data Collection and Simulation Logic

The simulation was implemented in Python using the PyGame library for visualization. The road network was parsed from a GraphML file. Global variables controlled the activation of obstacles, communication, self-rerouting, and OMM for each experimental run.

Vehicles are represented as objects, each with attributes for its current node, destination, speed, route, and a list of known obstacles (for OMM). When an obstacle is encountered, a vehicle broadcasts its location to all other agents. The `broadcast_obstacle` function ensures each obstacle is broadcast only once per detection to maintain communication efficiency. Receiving vehicles with OMM enabled update their internal list of known obstacles.

During path recalculation, Dijkstra's algorithm is run on a modified graph where known obstacle nodes are temporarily removed, preventing the agent from routing through them. This decentralized approach mimics how an AV would use shared data to make independent but coordinated decisions.

Extensive logging mechanisms recorded critical events for each vehicle, including start/finish times, total travel time, wait time at obstacles, and the number of route recalculations. A global communication log tracked all inter-vehicle messages. Post-simulation reports summarized these key metrics, facilitating rigorous comparative analysis.

IV. Results

The experiments were designed to evaluate how different levels of coordination affect routing efficiency. Key performance metrics collected include Total Travel Time, Wait Time at obstacles, and the Number of Route Recalculations. A summary of the average results across all 12 scenarios is presented in Table I.

TABLE I
SUMMARY OF AVERAGE TRAVEL/WAIT TIMES AND RECALCULATIONS ACROSS ALL SCENARIOS

Scenario	Avg. Travel Time (s)	Avg. Wait Time (s)	Avg. Recalc. per car
No Obstacle (Baseline)	19.35	0.00	0
Obstacle, No Reroute	36.16	16.59	0
Communication Only	36.79	14.56	3.08
OMM Only	32.19	9.43	1.58
Self-Reroute, No OMM	104.99	64.91	9.83
Self-Reroute with OMM	**32.55**	**7.81**	**1.67**

A. Impact of Obstacles and Communication

The introduction of obstacles consistently resulted in significantly increased travel times. Across all vehicle and obstacle densities, the average travel time nearly doubled from the baseline of 19.35s to 36.16s in the worst-case scenario. This highlights the severe disruptive effect of unexpected blockages on uncoordinated traffic flow.

Enabling communication without proactive rerouting offered minimal benefit, with average travel times remaining high (36.79s). Although wait times saw a slight reduction as some vehicles could anticipate blockages, the inability to change paths meant most vehicles were still delayed. This suggests that information without action is insufficient for mitigating congestion.

B. Effectiveness of OMM and Self-Rerouting

The most significant findings emerged from the comparison of advanced features. The self-rerouting mechanism, when activated *without* OMM, led to a catastrophic degradation in performance. Average travel times surged to 104.99s, nearly three times worse than the scenario with no rerouting at all. This counterintuitive result was caused by vehicles becoming trapped in recalculation loops. Without a memory of past obstacles, an agent would reroute away from one obstacle only to be directed back towards it after encountering a second, creating an inefficient cycle. This phenomenon is a critical finding of our study, demonstrating that naive, reactive rerouting can be more detrimental than simply waiting.

In stark contrast, the introduction of Object Memory Management (OMM) resolved this issue and proved to be the most effective strategy. When combined with self-rerouting, OMM reduced average travel times to 32.55s and average wait times to just 7.81s—a 53% reduction in waiting compared to the worst-case scenario. By retaining a memory of known obstacles, agents made more intelligent and efficient routing decisions, effectively circumventing blocked paths and avoiding loops.

Interestingly, enabling OMM without the self-rerouting trigger (i.e., vehicles rerouted based on broadcasted information

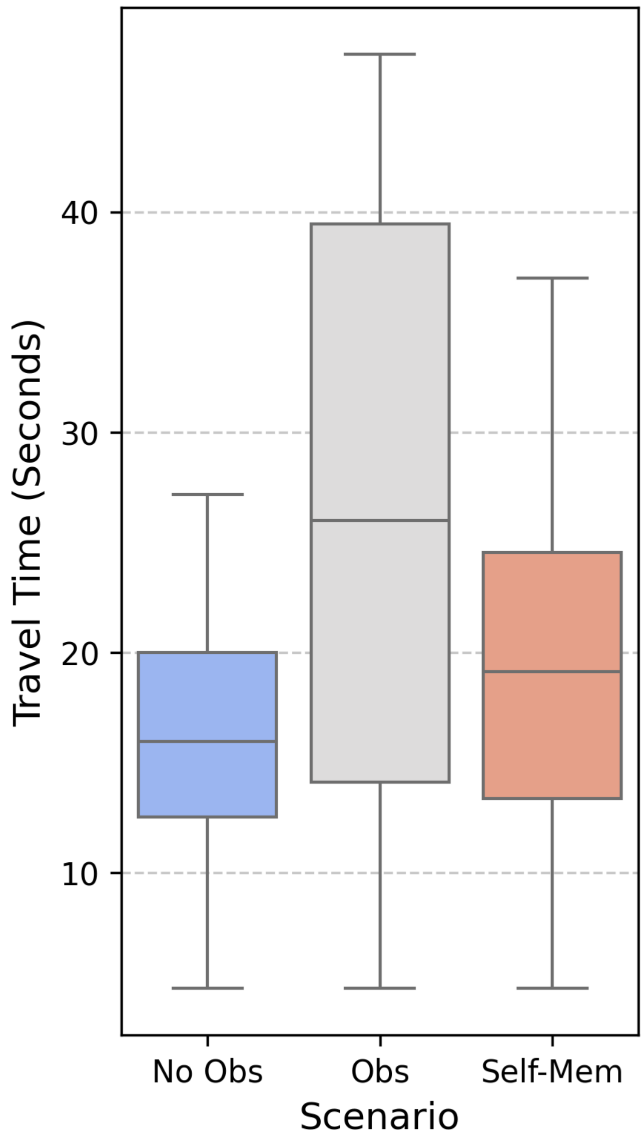

Fig. 3. Boxplot of travel times for a representative scenario, comparing No Obstacle (Baseline), Obstacle/No Reroute (Worst-Case), and Self-Reroute with OMM conditions.

but did not have the 8s wait-and-reroute behavior) yielded nearly identical performance (32.19s travel time). This indicates that the core benefit comes from the memory mechanism itself, which enables proactive avoidance, rather than the reactive self-rerouting trigger.

C. Scalability and Performance Under High Density

The effectiveness of the OMM-enabled algorithm was tested under increasing vehicle and obstacle density.

- **Vehicle Density:** As the number of cars increased from 15 to 55, baseline travel times rose due to natural congestion. However, the performance gains from the Self-Reroute with OMM algorithm remained consistent. For instance, with 6 obstacles, the algorithm reduced average travel time by 28% for 15 cars and 30% for 55 cars compared to the worst-case scenario. This demonstrates robust scalability.

- **Obstacle Density:** The algorithm's effectiveness diminished slightly as obstacle density increased. In the 15-car scenario, increasing obstacles from 6 to 20 reduced the travel time improvement from 28% to 24%. With 55 cars, the improvement dropped from 30% to 18%. This is due to the limited number of alternative paths available in a densely obstructed graph, forcing vehicles into longer detours.

Overall, the Self-Reroute with OMM algorithm proved to be robust and scalable, consistently outperforming other strategies across all tested conditions.

V. Discussion

Our findings provide strong empirical evidence that multi-agent coordination, specifically through V2V communication and a shared, persistent memory of the environment, significantly enhances the resilience of autonomous vehicle networks. The results underscore two critical insights for the design of cooperative AV systems.

First, information alone is not enough. Simply broadcasting obstacle locations without an effective mechanism for vehicles to act on that information yielded negligible improvements in traffic flow. This highlights that the value of V2X communication is contingent on the sophistication of the decentralized decision-making algorithms employed by the agents.

Second, and more critically, our study reveals the dangers of naive rerouting strategies. The failure of the memory-less self-rerouting algorithm, which created detrimental feedback loops, serves as a cautionary tale. It demonstrates that in a dynamic multi-agent environment, a vehicle's locally optimal decision (rerouting around an immediate obstacle) can lead to globally suboptimal—or even catastrophic—system behavior if it lacks broader spatial and temporal awareness. The agent, by forgetting a past obstacle, repeatedly made the same mistake, a pattern that could paralyze a real-world traffic grid.

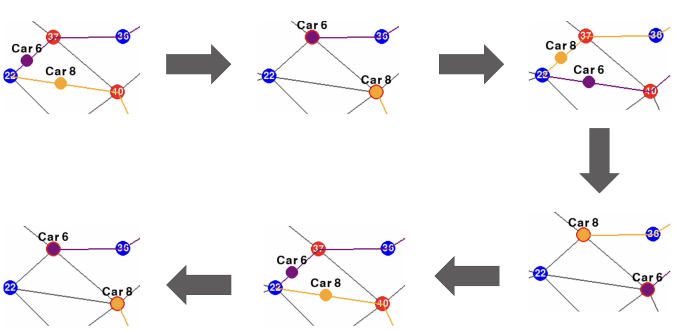

Fig. 4. Visualization of a vehicle trapped in a rerouting loop without OMM. The agent cycles between two obstacles after its path is recalculated.

The success of the Object Memory Management (OMM) feature directly addresses this flaw. By endowing each agent with a simple, lightweight memory of known obstacles, we transformed the system from one of reactive, myopic agents into a network of proactive, intelligent navigators. OMM essentially created a distributed, dynamic map of hazards that all agents could consult. This prevented routing loops and allowed the system to achieve a more efficient equilibrium. The fact that OMM's performance was consistent with or without the reactive 8-second self-rerouting trigger suggests that proactive planning based on shared knowledge is more powerful than reactive responses to immediate blockages.

The scalability analysis showed that while the OMM-based approach is robust, its effectiveness is ultimately constrained by the physical topology of the road network. In scenarios with high vehicle and obstacle density, the lack of viable alternative routes becomes the primary bottleneck, a problem that no rerouting algorithm can solve alone. This points to the need for higher-level traffic management strategies, possibly involving V2I communication, where infrastructure could suggest system-wide optimal routes or even manage traffic flow dynamically.

In summary, our work illustrates a clear progression in the intelligence of multi-agent AV systems: from isolated agents (no communication), to informed but passive agents (communication only), to dangerously naive reactive agents (memory-less rerouting), and finally to coordinated, proactive agents (OMM). The transition to this final stage, enabled by a simple memory mechanism, is what unlocks the true potential of cooperative autonomous driving.

VI. Future Work

Building on this study, several key areas warrant future exploration.

- **Complex Topologies:** Future experiments should utilize more realistic traffic networks, such as urban grids from real city maps, to validate the scalability and adaptability of the proposed coordination strategies.
- **High-Fidelity Simulation:** The next logical step is to implement and evaluate the OMM algorithm in a high-fidelity simulator like CARLA, which provides physics-based modeling, realistic sensor data, and stochastic agent behaviors.
- **Realistic Communication Models:** The current study assumes an ideal communication channel. Future work should integrate network simulation tools like NS-3 to model latency, packet loss, and bandwidth constraints, providing a more comprehensive understanding of the algorithm's reliability.
- **Mixed Traffic Scenarios:** Real-world traffic involves a mix of human-driven and autonomous vehicles. Extending the simulation to incorporate human-in-the-loop experiments would allow for an analysis of human-AV interactions and the design of more human-centric coordination strategies.
- **Agent Failure and Recovery:** Introducing random agent failures (e.g., an unresponsive AV) would allow for the evaluation of system resilience and the development of fault-tolerant coordination protocols.

VII. Conclusion

This study investigated the impact of V2V communication, self-rerouting algorithms, and Object Memory Management (OMM) on AV navigation in obstacle-rich environments. Our findings demonstrate that while obstacles significantly disrupt travel efficiency, the integration of intelligent, coordinated strategies can effectively mitigate these delays.

We found that self-rerouting without a persistent memory mechanism is highly inefficient and can paradoxically degrade traffic flow by trapping vehicles in decision loops. The most effective strategy emerged from integrating OMM with communication-enabled rerouting. Vehicles equipped with OMM exhibited substantial performance gains, reducing travel times by up to 75% in high-obstacle scenarios compared to naive rerouting. By enabling vehicles to store and share obstacle history, OMM facilitates proactive rerouting, minimizing redundant calculations and reducing wait times. These findings underscore the critical role of persistent, decentralized memory in optimizing autonomous vehicle navigation in complex and dynamic environments.

While this study provides strong evidence for the benefits of OMM and multi-agent communication, future work should focus on testing in large-scale, high-fidelity simulations and incorporating adaptive learning models. In summary, this research demonstrates that intelligent memory management, when combined with V2V communication, significantly enhances the resilience and efficiency of AV navigation, paving the way for more robust and optimized future mobility networks.

Acknowledgment

The author expresses gratitude to Dr. Daniel Palmer for his invaluable support and guidance throughout this project. Acknowledgment is also given to the generative AI tools that assisted in summarizing academic papers and generating experimental data. Finally, thanks to family and friends for their unwavering encouragement.

References

[1] R. Bellman, “On a routing problem,” *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[2] J. Billington, “C-V2X VS DSRC: Which technology is better for autonomous vehicles?” *Autonomous Vehicle International*, Oct. 2018. [Online]. Available: https://www.autonomousvehicleinternational.com/features/c-v2x.html

[3] R. Chen, J. Hu, and W. Xu, “An RRT-Dijkstra-based path planning strategy for autonomous vehicles,” *Applied Sciences*, vol. 12, no. 23, p. 11982, 2022.

[4] J. Cui et al., “Scalable multiagent driving policies for reducing traffic congestion,” *arXiv preprint arXiv:2103.00058*, 2021.

[5] E. W. Dijkstra, “A note on two problems in connexion with graphs,” in *Edsger Wybe Dijkstra: his life, work, and legacy*, 2022, pp. 287–290.

[6] J. Dinneweth et al., “Multi-agent reinforcement learning for autonomous vehicles: A survey,” *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 27, 2022.

[7] S. Djavadian et al., “Multi-objective eco-routing for dynamic control of connected & automated vehicles,” *Transportation Research Part D: Transport and Environment*, vol. 87, p. 102513, 2020.

[8] A. Dosovitskiy et al., “CARLA: An Open Urban Driving Simulator,” in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[9] R. W. Floyd, “Algorithm 97: shortest path,” *Communications of the ACM*, vol. 5, no. 6, pp. 345–345, 1962.

[10] FlyingTeller, “How to install Carla on macos with python 3.10?” *Stack Overflow*, May 2023. [Online]. Available: https://stackoverflow.com/questions/76172298/how-to-install-carla-on-macos-with-python-3-10

[11] Q. Gan et al., “Multi-Vehicle Cooperative Decision-Making in Merging Area Based on Deep Multi-Agent Reinforcement Learning,” *Sustainability*, vol. 16, no. 22, 2024.

[12] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[13] M. Kaufeld, R. Trauth, and J. Betz, “Investigating driving interactions: A robust multi-agent simulation framework for autonomous vehicles,” in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 803–810.

[14] P. A. Lopez et al., “Microscopic Traffic Simulation using SUMO,” in *The 21st IEEE International Conference on Intelligent Transportation Systems*, 2018.

[15] M. M. A. Muslam, “Enhancing security in vehicle-to-vehicle communication: A comprehensive review of protocols and techniques,” *Vehicles*, vol. 6, no. 1, pp. 450–467, 2024.

[16] M. Parulekar et al., “Automatic vehicle navigation using Dijkstra’s Algorithm,” in *2013 International Conference on Advances in Technology and Engineering (ICATE)*, 2013, pp. 1–5.

[17] A. Petrillo et al., “Adaptive multi-agents synchronization for collaborative driving of autonomous vehicles with multiple communication delays,” *Transportation research part C: emerging technologies*, vol. 86, pp. 372–392, 2018.

[18] I. Silva et al., “Realistic 3D simulators for automotive: a review of main applications and features,” *Sensors*, vol. 24, no. 18, p. 5880, 2024.

[19] R. E. Stern et al., “Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments,” *Transportation research part C: emerging technologies*, vol. 89, pp. 205–221, 2018.

[20] H. L. Tie, “V2X vehicle-to-everything communication – the future of autonomous connectivity,” *Keysight Blogs*, Oct. 2024. [Online]. Available: https://www.keysight.com/blogs/en/inds/auto/2024/10/03/v2x-post

[21] F. Vitale and C. Roncoli, “Cooperative Traffic Dispersion through Rerouting of Connected and Automated Vehicles in Urban Networks,” *Transportation Research Record*, p. 03611981241248648, 2024.

[22] T. Yoshizawa et al., “A survey of security and privacy issues in v2x communication systems,” *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–36, 2023.

[23] Y. Zhang et al., “Learning a robust multiagent driving policy for traffic congestion reduction,” *Neural Computing and Applications*, pp. 1–14, 2023.

[24] W. Zhou et al., “Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic,” *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 5, 2022.