Linnæus University
Sweden

Master Thesis

SIMULATED ANNEALING FOR VEHICULAR AD-HOC NETWORKS (VANETs)

Author: Ethish Venumbakka

Supervisor: Sven Nordebo

Examiner: Sven-Erik Sandström

Date: 06.11.2023

Course Code: 5ED36E

Level: Master’s

Department of Physics and Electrical Engineering, Faculty of Technology.

Abstract

In this thesis, we tackle a significant optimization challenge within Vehicular Ad Hoc Networks (VANETs) by employing a simulated annealing approach. We focus on developing an efficient Vehicle Routing Problem (VRP) algorithm to sift through numerous potential solutions and identify the best one.

Our VANET scenario revolves around four distinct vehicles traversing four unique routes. The primary objective is to minimize the total distance covered by these vehicles while ensuring that they visit all designated waypoints. We implement this problem using MATLAB to establish initial routes for each simulation uniquely.

Simulated annealing proves to be a valuable tool in optimizing VANETs. The gradual cooling process reduces the likelihood of accepting suboptimal solutions over time, allowing the algorithm to escape local optima and converge towards nearly optimal solutions.

Regarding routing protocol parameter configuration, simulated annealing is the technique of choice for identifying the most influential parameters. It evaluates the cost and creates new routes based on neighboring nodes, calculating the cost function for these new routes. Starting from an initial configuration, the algorithm iteratively refines it by introducing random changes, retaining only those that enhance the objective function. Our objective function defines the Quality of Service (QoS) and communication efficiency of the routing protocol. The gradual reduction in the acceptance of less favorable configurations over time is called the annealing schedule, enabling the algorithm to escape local optima and approach nearly optimal designs.

Table of Contents

1. Introduction

A well-functioning transportation system is crucial for the efficient movement of people, goods, and services. Ideally, we aim to minimize transportation activities. However, it is tough to avoid transportation entirely. Consequently, humans continuously look for methods to reduce the need for transportation. One aspect of transportation involves the distribution of goods within a network, where a single vehicle is tasked with delivering multiple items to various destinations efficiently. A fundamental model for this goods transportation is the Traveling Salesman Problem (TSP), which has been extensively studied mathematically. The Traveling Salesman Problem has led to the development of more advanced transportation optimization models, such as the Vehicle Routing Problem (VRP). In VRP, multiple vehicles are engaged in transportation, making it a valuable algorithm for Vehicular Ad Hoc Networks (VANETs) applications.

1.1 Why VANETs

Vehicle Ad hoc Networks (VANETs) are an essential technology that could change the transportation industry [1].

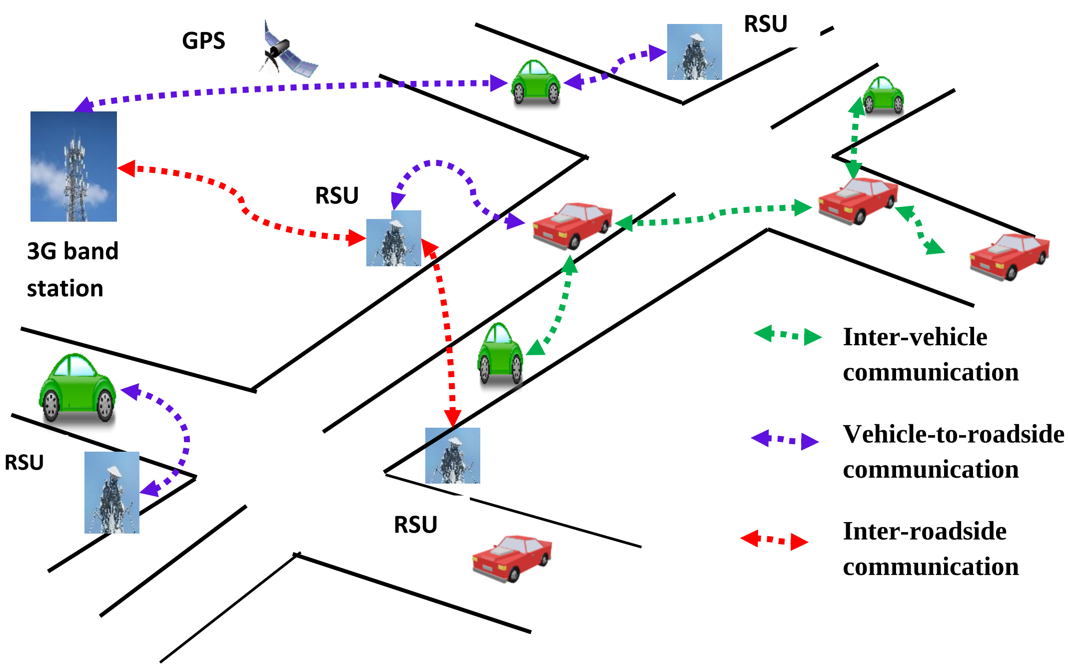


Fig 1 VANET [69]

Improving road safety, traffic efficiency, emergency Services, Driver Assistance, etc.... is one of the main reasons for using VANETs. Vehicles can connect with roadside unit (RSU) infrastructure, and VANETs are crucial in reducing the risk of accidents. Cars could recognize and avoid the risk of collision by exchanging real-time data based on their location, speed, direction, and other important information. This could help significantly drop collisions, injuries, and casualties on the roads. [2].

1.2 History of VANETs

In the late 1990s to early 2000s, VANETs emerged, focusing on vehicle-to-vehicle communication for collision avoidance and traffic information dissemination. 2002 to 2006: CAR-2-CAR Communication Consortium was formed to promote developing and standardizing cooperative intelligent transportation systems (C-ITS). Large-scale field tests and experiments were conducted. 2006 to 2010: Research projects like the U.S. DOT's Vehicle Infrastructure Integration (VII) and European COMeSafety contribute to VANET advancement. Integration of vehicles with roadside infrastructure for improved safety and efficiency. 2010 to 2015: Focus shifts to scalability, security, and privacy. Standardization efforts like IEEE 802.11p and DSRC standards provide a common framework for VANET communication. 2015 to 2021: The rise of connected and autonomous vehicles (CAVs) brings increased attention to VANETs. Exploration of new communication technologies like cellular networks (LTE, 5G) and V2X communication. Integration with edge computing and AI for intelligent transportation systems [5-10]. Beyond 2021: Ongoing research emphasizes security, privacy, communication reliability, and novel applications. Regional variations in development and deployment exist due to infrastructure, regulations, and technology differences. VANETs continue to evolve rapidly, with new stories shaping their future.

In summary, the history of VANETs started with early research and evolved through standardization efforts, field tests, and the development of communication technologies. The focus shifted from basic V2V and V2I communication to more comprehensive C2X communication, leading to the exploration of cooperative intelligent transport systems and the potential integration of emerging technologies.

1.3 Classification VANETs

Vehicular Ad hoc Networks (VANETs) can be classified based on various factors [15]. Based on their communication characteristics, network topology, and deployment scenarios, VANETs are classified into,

a) Communication Characteristics:

Vehicle-to-Vehicle (V2V) – In this type, VANET focuses on communication between the vehicles, which allows the vehicle's information, such as position, speed, and acceleration, to be exchanged directly.

Vehicle-to-Infrastructure (V2I) – In this type of communication, the vehicle exchanges information between vehicles and Roadside Unit (RSU) infrastructure, which leads cars to access the infrastructure’s services, such as traffic signal information, road condition updates, and parking availability.

Vehicle-to-Pedestrian (V2P) – In this type of communication, the vehicle exchange information with pedestrians and ensure safety by providing warnings to drivers when the pedestrian is nearby, especially in areas where there are many people like metropolitan area and school zones.

Vehicle-to-Network (V2N)- In this type of communication, the vehicles connect to external networks, such as cellular networks or cloud services, where cars can access real-time traffic information.

b) Network Topology:

Infrastructure-based VANETs facilitate communication between vehicles. Roadside Units (RSUs) are strategically placed to help cars communicate at intersections and highway exits.

Infrastructure-less VANETs, the vehicle connects directly with each other without any particular infrastructure. These networks are self-organized and are very useful for setting up in places like rural areas with high infrastructure costs.

c) Deployment Scenarios:

Urban VANETs are designed for crowded places like the urban area where the vehicle is high in volume and has a complex road network. It helps cars tackle traffic jams, prevent accidents, and find the best route within urban areas. Highway VANETs, on the other hand, are designed for long-distance and high-speed scenarios. The network prioritization application focuses on applications such as smart merging into lanes and real-time monitoring or boosting highway safety and efficiency. These networks concentrate on providing connectivity, sharing information, and offering emergency services in rural locations, where the condition of the road risks and differs much from urban areas.

1.4 Challenges, risks, and limitations of VANETs

Vehicular Ad hoc Networks (VANETs) face several challenges, risks, and limitations that must be addressed for successful implementation and widespread use. Here is a detailed explanation of the main challenges associated with VANETs:

Dynamic and High-Speed Environment: The vehicles can rush and change their positions. This creates difficulties in maintaining dependable and robust communication links between the cars.

Scalability: VANETs should be able to support many vehicles communicating simultaneously within a specific geographic area. As the number of cars increases, the network scalability becomes a challenge.

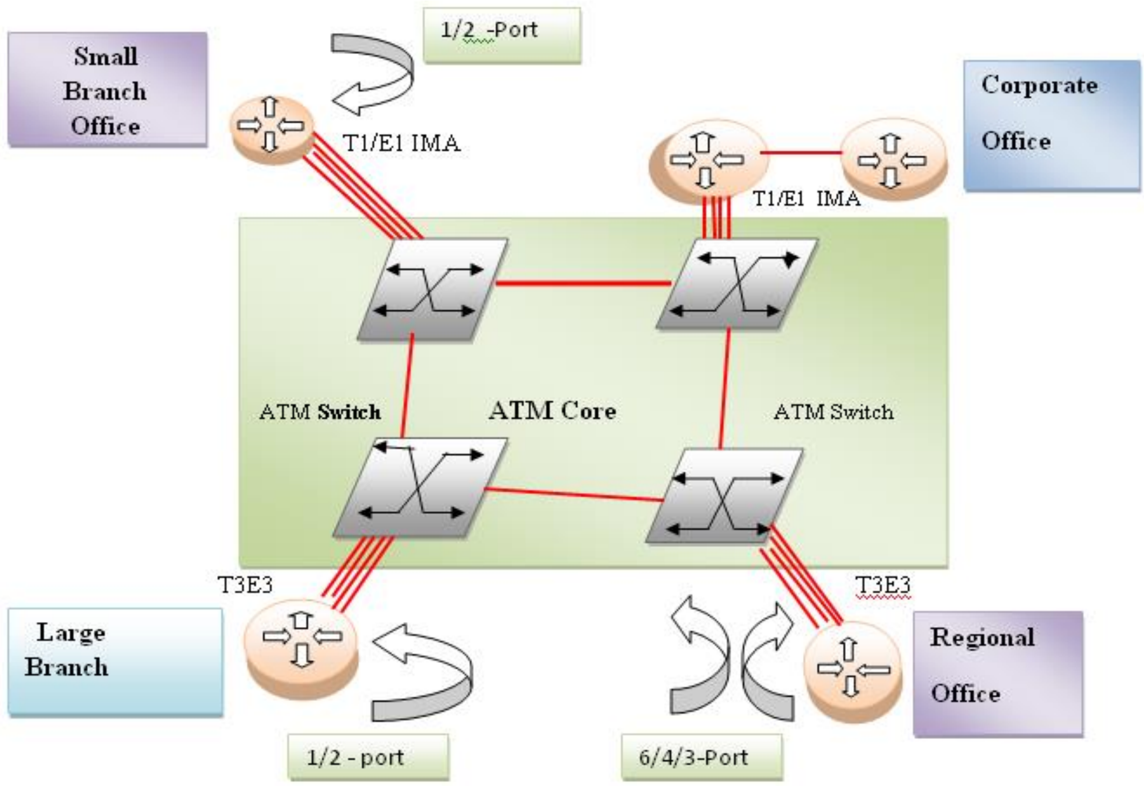
Network Management and Control: Managing the network efficiently in a decentralized and self-organized manner with sophisticated algorithms and protocols that can adapt to the changing network condition to optimize the performance is needed.

In short, VANETs have several challenges regarding communication reliability, scalability, security, privacy, quality of service, network management, and cost-effectiveness. Overcoming these challenges needs robust technical solutions, collaboration between all parties concerned, standardization efforts, and careful consideration of the social, ethical, and economic implications of VANET deployment.

2. Routing algorithms

2.1 What are routing algorithms?

Routing algorithms play an essential role in forwarding data packets within a network. They are fundamental components of network protocols and are responsible for efficient and reliable packet delivery. Routing algorithms are designed to address the challenges of finding the optimal or near-optimal paths, avoiding network congestion, handling network failures, and achieving scalability. Here are some routing algorithms:



ATM – Asynchronous Transfer Mode, IMA – Inverse Multiplexing for ATM

Fig 2 Routing

Distance Vector Routing algorithms, such as the Bellman-Ford algorithm, calculate the shortest path based on the number of hops or the sum of link costs between nodes. Each node maintains a routing table containing the distances to all other nodes in the network. Distance Vector Routing algorithms iteratively exchange routing information with neighboring nodes to update their routing tables and converge on the optimal paths. This algorithm is relatively simple but may suffer from slow convergence and routing loops.

Link State Routing algorithms, such as the Shortest Path First (SPF) or the Dijkstra algorithm, build a detailed network map by flooding information about all network links to every node. Each node then calculates the shortest path based on this complete network map. Link State Routing algorithms provide faster convergence, more accurate routing decisions, and better fault tolerance than Distance Vector Routing. However, they require more computational resources and generate larger routing tables.

Path Vector Routing algorithms, such as the Border Gateway Protocol (BGP) used in the Internet, are designed for large-scale networks and inter-domain routing. Path Vector Routing algorithms consider multiple routing attributes, such as path length, policy rules, and autonomous system (AS) information, to make routing decisions. They ensure that routes are selected based on policy preferences, avoid routing loops, and provide a scalable routing solution for complex networks.

Routing algorithms are essential for efficient and reliable packet delivery in various network environments. The choice of routing algorithm depends on factors such as network size, traffic patterns, scalability requirements, and network dynamics.

2.2 Why routing is essential in VANETs.

Routing plays a crucial role in Vehicular Ad hoc Networks (VANETs) due to the dynamic nature of the network and its unique challenges. Vehicular ad-hoc networks (VANETs) enable vehicle-to-vehicle and vehicle-to-roadside communication for road safety, efficient traffic flow, and diverse applications [17].

VANETs transmit various data types, including real-time traffic updates, emergency notifications, multimedia content, and location-based services. Efficient routing ensures that data packets are delivered promptly to their intended destinations. By finding the optimal or near-optimal paths, routing algorithms help minimize delays and ensure that time-sensitive information reaches its destination quickly. Routing in VANETs plays a crucial role in managing traffic effectively. By disseminating information about traffic conditions, roadblocks, accidents, and congestion, routing algorithms facilitate traffic coordination and help drivers make informed decisions. Efficient routing helps vehicles avoid congested areas, find alternative routes, and contribute to overall traffic flow optimization.

VANETs encompass many vehicles that move at high speeds and frequently change their positions. Routing algorithms ensure the network remains scalable and can handle increasing numbers of cars. Routing protocols optimize bandwidth utilization and prevent congestion by efficiently managing network resources.

In summary, routing is vital in VANETs to ensure efficient data delivery, manage traffic, enhance road safety, optimize network resources, adapt to dynamic network conditions, provide fault tolerance, accommodate resource constraints, and meet diverse QoS requirements. Practical routing algorithms are crucial for the successful operation of VANETs and the realization of their numerous applications and benefits.

2.3 Algorithms in VANETs.

To improve the effectiveness and dependability of communication among cars in Vehicular Ad hoc Networks (VANETs), several routing algorithms have been developed. These algorithms are designed to address vehicle contexts' unique characteristics and difficulties. Here are some examples of routing algorithms used often in VANETs:

a. Greedy Perimeter Stateless Routing (GPSR) utilizes location information to determine the next hop for packet forwarding. It uses greedy forwarding, forwarding packets to the nearest neighbor to the destination. It activates perimeter mode to find a way around obstacles when it encounters a local maximum.
b. Ad hoc On-demand Distance Vector, is a routing algorithm that operates on a demand-driven basis and relies on positional data to create routes between source and destination vehicles. It employs processes for discovering routes as needed and for sustaining them through ongoing maintenance as necessary.
c. The Cluster-Based Routing Protocol (CBRP) arranges vehicles into clusters, with each cluster having a designated cluster head responsible for managing routing within that cluster. These cluster heads establish communication among themselves to exchange routing information and enable inter-cluster communication.
d. The Probabilistic Routing Protocol for Intermittently Connected Networks (PRoPHET) is tailored for routing in intermittently connected networks, including VANETs. It employs a probabilistic forwarding mechanism, where nodes share data regarding their past interactions with other nodes. This shared information is then utilized to estimate the likelihood of future encounters and determine the optimal path for transmitting data.

e. Roadside Infrastructure-Aided Routing (RIAR) makes use of the existence of roadside infrastructure elements, like roadside units (RSUs), to enhance routing capabilities. RSUs serve as fixed reference points and provide guidance to vehicles when determining routes and making decisions regarding data forwarding [20].

These are just a few examples of the routing algorithms used in VANETs. Each algorithm has strengths and limitations, and its suitability depends on network size, mobility patterns, QoS requirements, and available infrastructure. Ongoing research and development in VANETs continue to explore new and improved routing algorithms to address the evolving needs and challenges of vehicular communication.

3. Simulated Annealing

Annealing is a heat treatment process used to modify the properties of metals or alloys by heating them to a specific temperature and allowing them to cool slowly.

Likewise, Simulated Annealing (SA) is an optimization technique inspired by the annealing process. It is often used to tackle problems where you must find the best solution from many options. Simulated Annealing works by slowly cooling down the system, just like in annealing, to help it avoid getting stuck in less-than-optimal solutions and explore better ones [31].

The Boltzmann distribution is a fundamental concept within statistical physics and thermodynamics. It depicts how particles or energy states become distributed within a system at a specific temperature. The mathematical representation of the Boltzmann distribution takes the form:

$P(E) = (1/Z) * \exp(-E / kT)$

In this equation:

$P(E)$ symbolizes the probability of locating a particle within an energy state denoted as E.

Z refers to the partition function, acting as a normalization constant. Its role is to ensure that the probabilities add up to 1 across all conceivable energy states.

E represents the energy characteristic of the state in question.

k is Boltzmann's constant, serving as the bridge between energy and temperature.

T signifies the temperature measured in Kelvin.

The algorithm starts with an initial solution and iteratively explores neighbouring solutions. It accepts worse solutions based on a probability that decreases as the temperature drops. This probabilistic acceptance allows the algorithm to escape local optima and potentially converge towards a globally optimal solution.

The core of Simulated Annealing lies in the acceptance criterion, which is based on the Metropolis criterion. The probability of accepting a worse solution depends on the objective function value difference between the current solution (S) and the candidate solution (S'). The probability is calculated using the Boltzmann distribution:

$P(S \rightarrow S') = \exp[- (f(S) - f(S')) / T]$

Where:

- $P(S \rightarrow S')$ is the probability of accepting the transition from S to S'.
- f(S) is the function value of the current solution S.
- f(S') is the function value of the candidate solution S.'
- T is the current temperature.

The above equation is used in simulated annealing and other optimization algorithms that employ a probabilistic acceptance criterion. This equation can be used for both maximization and minimization problems, depending on how we set up the objective function and the associated optimization task.

Here, we could look for both scenarios:

Maximization: If the goal is to maximize the objective function f(S), we could prefer to move toward solutions with higher values of f(S). In this case, we should accept a new solution S', if it has a higher objective function value ($f(S') > f(S)$). In terms of the equation, a higher f(S') results in a smaller exponent inside the exponential, which increases the probability of accepting the new solution S.' So, the equation can be used effectively for maximization by selecting S' if $f(S') > f(S)$.

Minimization: If the goal is to minimize the objective function f(S), we could aim to move toward solutions with lower values of f(S). In this scenario, we should accept a new solution S', if it has a lower objective function value (f(S') < f(S)). In the equation, a lower f(S') results in a more prominent exponent inside the exponential, which still increases the probability of accepting the new solution S.' So, the equation can also be applied to minimization problems by selecting S' if f(S') < f(S).

At the start of the algorithm, the temperature is set to a high value, allowing for more exploratory moves. As the algorithm progresses, the temperature decreases according to a cooling schedule. The cooling schedule determines the rate at which the temperature drops and influences the balance between exploration and exploitation. A typical cooling program is the exponential cooling:

$T(k+1) = \alpha * T(k)$

Where:

- $T(k+1)$ is the temperature at iteration $k+1$.
- $T(k)$ is the temperature at iteration k.
- α is the cooling rate ($0 < \alpha < 1$).

The cooling rate determines the speed of temperature reduction. A slower decline allows for more exploration, while a faster reduction promotes exploitation.

The Simulated Annealing algorithm follows the following steps:

1. Initialize the temperature T and the initial solution S.
2. Repeat until the stopping criterion is met:

   a. Generate a candidate solution S' by applying a perturbation to the current solution S.
   b. Calculate the objective function values f(S) and f(S').

   c. If f(S') < f(S), accept S' as the new current solution.

   d. If f(S') ≥ f(S), calculate the acceptance probability P(S → S') using the Boltzmann distribution.

e. Generate a random number r between 0 and 1. f. If r <P(S → S'), accept S' as the current solution.

g. Update the temperature using the cooling schedule.

n. The best solution found during the iterations. $\text{Open}(S) = \begin{cases} f(\mathrm{S}'),\ \mathrm{f}(\mathrm{S}') < \mathrm{f}(\mathrm{S}) \\ f(\mathrm{S}'),\ r < \mathrm{P}(\mathrm{S} \to \mathrm{S}'), \\ f(S),\ r > \mathrm{P}(\mathrm{S} \to \mathrm{S}') \end{cases}$

Simulated Annealing provides a trade-off between exploration and exploitation by allowing the algorithm to escape local optima while converging towards an optimal solution. The acceptance probability based on the Boltzmann distribution and the cooling schedule controls the algorithm's behavior. Adjusting these parameters allows the algorithm's performance to be fine-tuned for different optimization problems [32-33].

It's important to note that the detailed derivations of the equations and the specific design choices for Simulated Annealing can vary depending on the problem being solved and the implementation approach. The presented equations and steps provide a general understanding of the algorithm's principles and core components.

3.1 How to choose the acceptance rate for simulated annealing.

Choosing the acceptance rate or probability for Simulated Annealing involves finding a balance between exploration and exploitation. The acceptance probability determines the likelihood of accepting a worse solution as the algorithm progresses. While there is no universally optimal acceptance rate, the following approach can be used to guide the selection:

Start by defining a function that calculates the acceptance probability based on the difference in objective function values between the current solution (S) and the candidate solution (S').

At the beginning of the Simulated Annealing process, when the temperature is high, you want a relatively high acceptance rate to encourage exploration. The cooling schedule determines how the temperature decreases over time. It controls the speed of exploration and exploitation throughout the algorithm. Different cooling schedules are available, such as linear, logarithmic, or geometric. The choice of cooling schedule depends on the problem and desired behavior. As the algorithm progresses and the temperature decreases, the acceptance rate should be reduced to promote exploitation and focus on refining the current solution. The cooling schedule defines the rate at which the temperature decreases and can also influence how the acceptance rate is updated. For example, you can use a linear cooling schedule that decreases the acceptance rate linearly with the temperature.

Run Simulated Annealing with different acceptance rates and observe the algorithm's behavior and performance. Evaluate the quality of the solutions found, convergence speed, and solution diversity. Adjust the acceptance rate and cooling schedule parameters iteratively until satisfactory results are achieved.

It's important to note that the selection and fine-tuning of the acceptance rate involve experimentation, problem-specific considerations, and understanding the landscape of the objective function. The chosen acceptance rate should balance exploration and exploitation, allowing the algorithm to escape local optima while converging towards the global optimum.

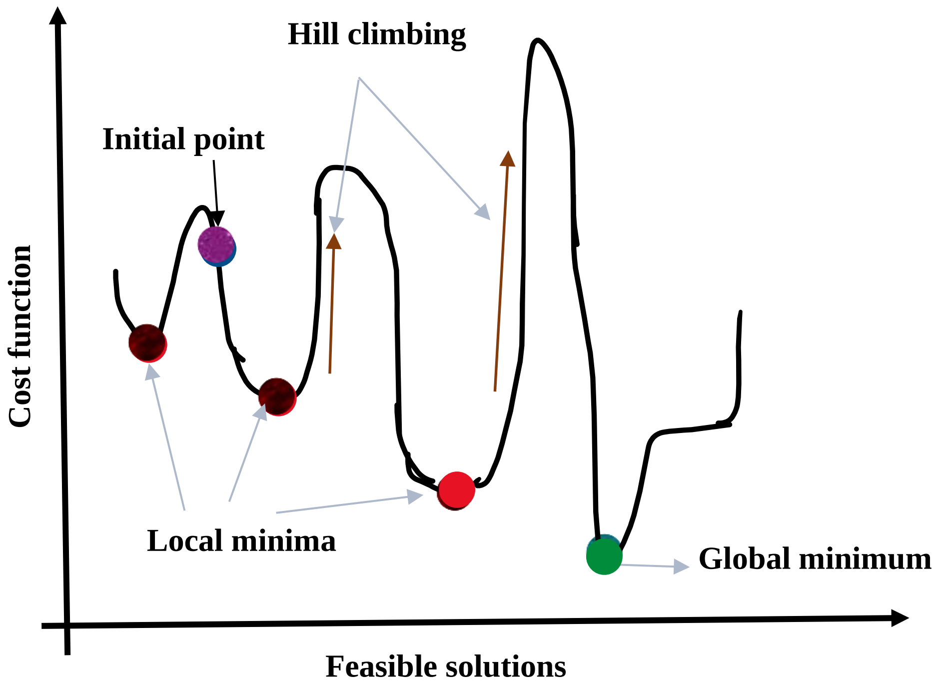


Fig 3 simulated annealing (SA)

The SA is a stochastic algorithm involving asymptotic convergence and allowing random movements in the searched neighborhood to escape local minima [31]. Derivations for the acceptance probability function and the impact of the cooling schedule on the acceptance rate are intricate and highly problem-dependent. The presented acceptance probability equation and steps provide a general understanding of the concept and guidelines for selecting and updating the acceptance rate during Simulated Annealing. In practice, empirical testing and domain-specific knowledge often determine the acceptance rate to achieve optimal results for a particular problem.

3.2 Problem Formulation

This thesis addresses a Vehicle Ad-Hoc Network (VANET) problem involving four vehicles traveling along four routes. The primary goal is to minimize the cumulative distance these four vehicles cover by identifying the optimal route that encompasses all designated waypoints. The problem is implemented using MATLAB, where random integer generation functions are leveraged to secure unique initial ways for each simulation.

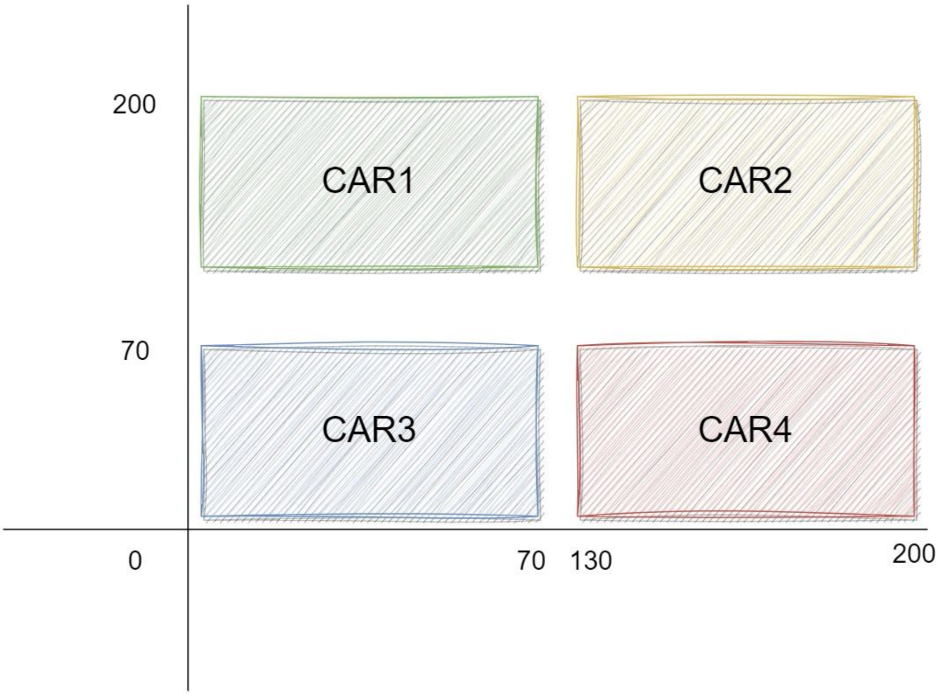


Fig 3.1 2D Map of four Cars in 200*200 Grid [70].

Initialization Function: In MATLAB, depicted in Figure 3.1, a 2D map with a maximum length of 200 units along both axes is illustrated. The map is partitioned into four equally sized quadrants, with the centre designated as (100,100). As the figure indicates, each of the four cars is assigned a specific quadrant to traverse. Utilizing the randi() function in MATLAB, 25 distinct random points within each quadrant are selected. Additionally, this function computes the distances between these various points and stores them in an array. Each quadrant has 25 distinct attributes for each car to visit. The randi() function is employed again to establish the sequence in which these points are traversed. An essential condition is imposed to guarantee that the final destination for all cars is the map's central point. Furthermore, the function ensures the uniqueness of these 25 points for each vehicle and prevents any car from occupying the exact location.

Following the application of these specified conditions, the initial routes for the cars are determined. Subsequently, a cost is computed, representing the cumulative distance covered as the cars navigate these four quadrants.

Cost = Σ d_ij

where d_ij is the distance between two points i and j.

The overarching objective of the Simulated Annealing (SA) problem is to minimize this cost function.

Minimize: Σ d_ijss

Table 1- Values used in problem formulation

INITIAL TEMPERATURE	500
DAMPING RATE	0,997
STOP TEMPERATURE	1
ITERATION	300
MAX ITERATION	2100

Solution Overview: In the context of the Simulated Annealing (SA) algorithm, the initial temperature is set to 500, and the iterations continue until the temperature reaches 1. For each temperature level, a new route is randomly generated using one of three distinct functions. These functions serve to diversify the solution and facilitate the escape from local optima during the optimization process.

1. **Swap Function**: The Swap function generates a new route by swapping the positions of two distinct points within the given initial route. For instance, if the initial route is

[1,2,3,4,5,6,7,8], the Swap function may produce a new route like [1,2,3,8,5,6,7,4]. It is important to note that the resulting route is subject to constraints, as outlined in Section 3.2.

2. **Reversion Function**: In the Reversion function, two distinct points are randomly selected from the initial route, and the sequence of points between them is reversed to create a new route. For example, if the initial route is [1,2,3,4,5,6,7,8], the Reversion function could yield a new route like [1,2,8,7,6,5,4,3]. The randomness in the selection of points ensures diversity in the generated routes.

3. **Insertion Function**: The Insertion function involves removing a point from its current position in the initial route and inserting it elsewhere to form a new route. For instance, if the initial route is [1,2,3,4,5,6,7,8], the Insertion function might result in a new way like [1,2,3,8,4,5,6,7]. This process continues until various possible avenues are explored.

Route Evaluation and Acceptance: For each new route generated, its cost is calculated. If the cost of the new course is superior to the previously established trial, the new way is accepted. Otherwise, the iteration continues. This step allows the algorithm to optimize the solution gradually.

Handling Local Minima: To escape local minima, the algorithm may accept a worse solution than the previous one under specific conditions. This acceptance is determined by evaluating the probability, denoted as P. If P is less than a randomly generated value within the range of 0 to 1, where P is defined as:

$$P = \exp(-\Delta /T)$$

Where:

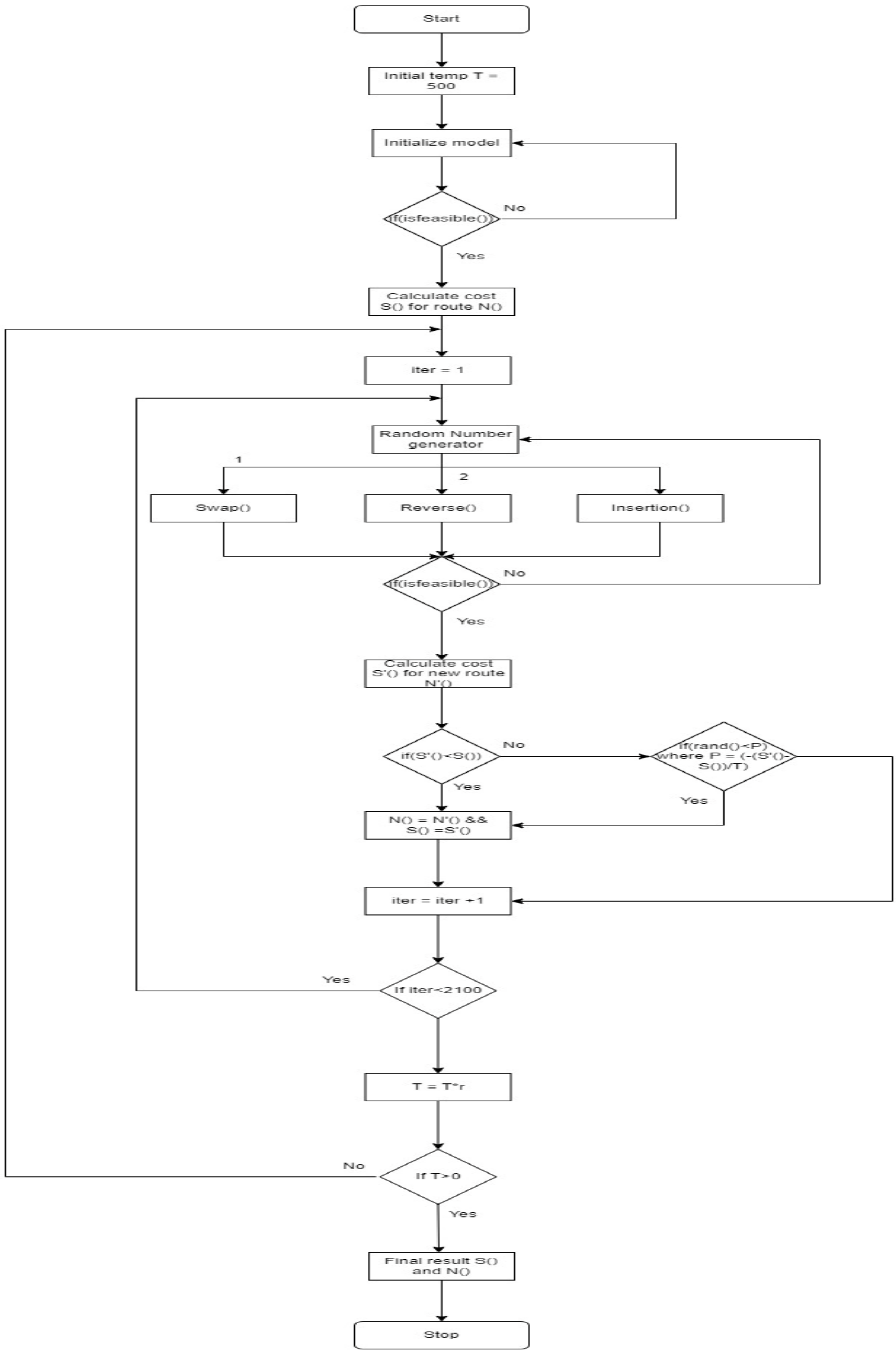
P is the probability of accepting a worse solution.

Δ (delta) represents the difference between the new cost and the current cost (Δ= new cost − cost).

T is the temperature parameter in the Simulated Annealing algorithm.

This equation is used in the Simulated Annealing algorithm to calculate the probability of accepting a worse solution (higher cost) based on the difference in price (Δ) and the current temperature (T).

3.3 Flow chart of Simulated Annealing


Start
Initial temp T = 500
Initialize model
if(isfeasible())
No
Yes
Calculate cost S() for route N()
iter = 1
Random Number generator
1
2
Swap()
Reverse()
Insertion()
if(isfeasible())
No
Yes
Calculate cost S'() for new route N'()
if(S'()<S())
No
Yes
if(rand()<P) where P = (-(S'()-S())/T)
Yes
N() = N'() && S() =S'()
iter = iter +1
If iter<2100
Yes
T = T*r
If T>0
No
Yes
Final result S() and N()
Stop

4. Plots and Results

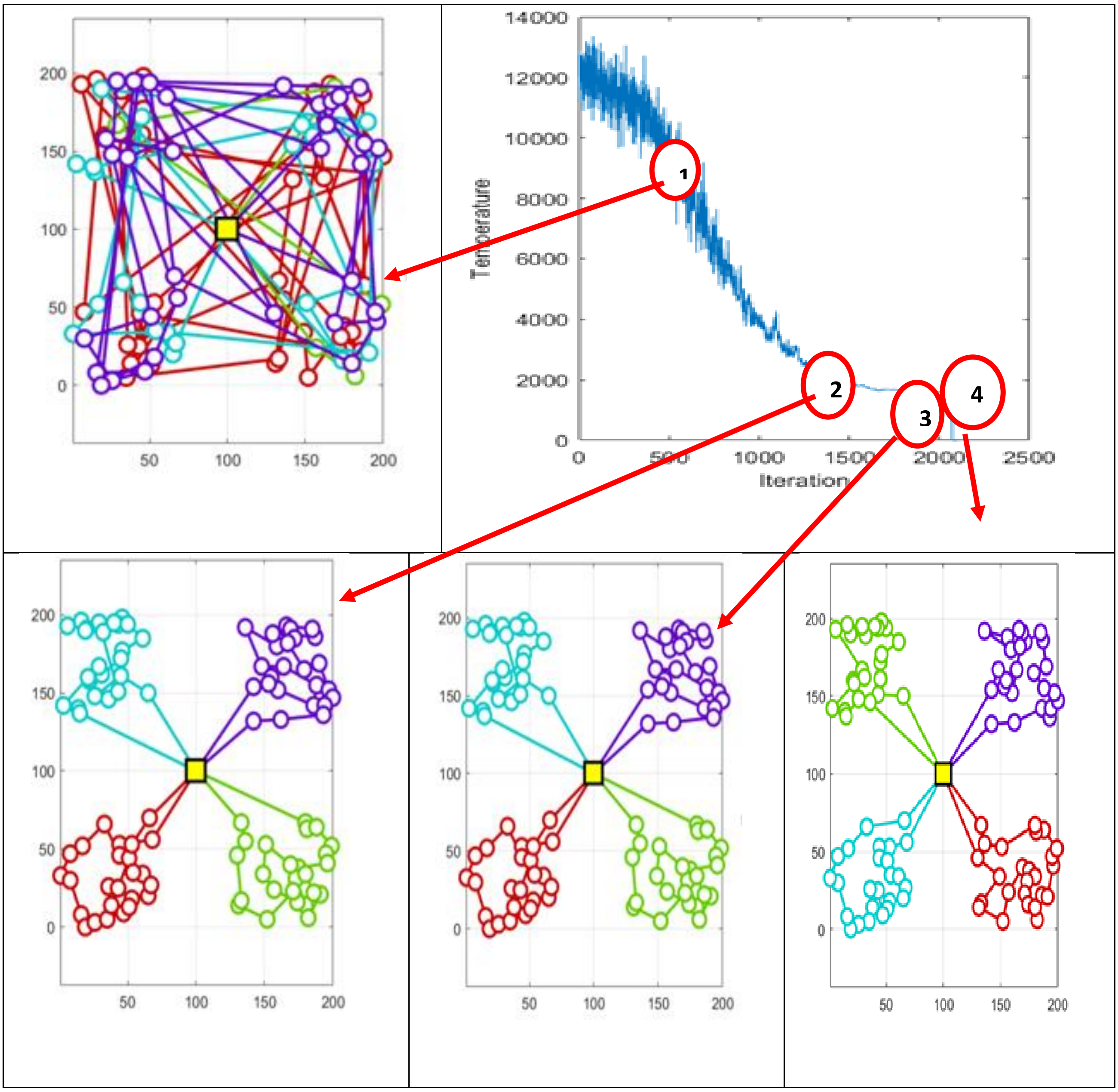


Fig 4 output

When applying simulated annealing to VANET, the goal is to optimize the performance or efficiency of the network in some way. This can involve minimizing communication delays, maximizing network coverage, optimizing route selection, or improving overall network throughput. Simulated annealing can be used as an optimization technique to find an optimal configuration or arrangement of vehicles and network parameters.

1. Start with an initial configuration of the VANET system, including the placement of vehicles and network parameters.

2. Evaluation: Evaluate the current configuration based on the defined optimization objective or fitness function. This function quantifies the performance or efficiency of the network.
3. Neighbouring Solution Generation: Generate a neighboring solution by modifying the current configuration. These modifications can involve changing the positions of vehicles, adjusting communication parameters, or modifying network settings.
4. Acceptance Criteria: Determine whether to accept the neighboring solution or stick with the current one. Simulated annealing introduces a probabilistic criterion for accepting worse solutions early in the optimization process, allowing exploration of different regions of the search space.
5. Cooling Schedule: Adjust the temperature parameter, which controls the probability of accepting worse solutions, to decrease the acceptance probability slowly over time. This gradual cooling process helps the algorithm converge toward the optimal solution.
6. Iteration: Repeat steps 3-5 until a termination condition is met. This condition can be a maximum number of iterations, reaching a predefined fitness threshold, or running for a specific time.
7. Output: Once the termination condition is met, output the best configuration found during the optimization process, representing the optimal solution according to the defined objective or fitness function.

By applying simulated annealing to VANET, the algorithm explores different configurations and gradually converges toward an optimal solution that improves the performance or efficiency of the network.

In Figure 4, the algorithm iteratively explores the search space by generating neighboring solutions and evaluating their fitness from 1 to 4. It accepts worse solutions early in the optimization process to avoid getting trapped in local optima, allowing it to find better solutions in other regions of the search space.

The acceptance of worse solutions is controlled by a temperature parameter that decreases gradually, following a cooling schedule. Initially, the algorithm is more likely to accept worse solutions, but as the temperature drops, the probability of taking worse answers decreases. This cooling process allows the algorithm to converge toward better solutions as it progresses.

The termination condition for the algorithm can be based on a maximum number of iterations, reaching a predefined fitness threshold, or running for a specific amount of time. Once the termination condition is met, the algorithm outputs the best solution found during the optimization process, representing a good solution but not necessarily the optimal one.

The simulated annealing algorithm explores the solution space by generating different vehicle routing configurations and gradually improving them. The process begins by developing an initial solution, representing a set of routes for the vehicles. An evaluation function is defined to calculate the objective value of each solution, considering factors like travel time, congestion levels, or communication overhead. Simulated annealing then explores neighboring solutions by making minor modifications, such as reassigning vehicles to different routes or rearranging waypoints. The evaluation function is applied to each new key, and a probabilistic acceptance criterion determines whether to accept the modification or not, allowing for exploration of the search space. As the algorithm progresses, the acceptance probability decreases, and the investigation is gradually reduced, focusing on exploiting promising solutions. The temperature parameter controls this exploration-exploitation trade-off and is slowly lowered over time. By repeating these steps for a specified number of iterations, simulated annealing converges toward an optimal or near-optimal vehicular routing configuration, effectively finding routes that optimize the desired objectives, such as minimizing travel time or maximizing network throughput.

5. Conclusion

To summarize, our investigation showcases the immense promise of VANETs in revolutionizing road safety, optimizing traffic flow, and facilitating intelligent transportation systems. Overcoming challenges related to communication reliability, security, and scalability remains crucial for the successful implementation of VANETs. However, given the continuous technological advancements and the growing infrastructure support, VANETs offer an unprecedented opportunity to transform the transportation sector, paving the way for a safer and more efficient road network.

REFERENCES

1. Arain, Q. A., Memon, I., Deng, Z., Memon, M. H., Mangi, F., &Zubedi, A. A. (2017). Location monitoring approach: Multiple mix-zones with location privacy protection based on traffic flow over road networks. *Multimedia Tools and Applications*, *77*, 5563–5607.

2. Arain, Q. A., Zhongliang, D., Memon, I., Arain, S., Shaikh, F. K., Zubedi, A., et al. (2016). Privacy-preserving dynamic pseudonym-based multiple mix-zones authentication protocol over road networks. *Wireless Personal Communications*, *95*, 1–17.

3. Barr, R., & Haas, Z. J. (2005). *JiST/SWANS, wireless networks laboratory*. Ithaca: Cornell University.

4. Bayat, M., Barmshoory, M., Rahimi, M., &Aref, M. R. (2015). A secure authentication scheme for VANETs with batch verification. *Wireless Networks*, *21*(Num 5), 1733–1743.

5. Bazzi, A., Masini, B. M., Zanella, A., & Thibault, I. (2017). On the performance of IEEE 802.11 p and LTE-V2V for the cooperative awareness of connected vehicles. *IEEE Transactions on Vehicular Technology*, *66*(11), 10419–10432.

6. Bellare, M., &Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. *Proceedings of the CCS*, *93*, 67–73.

7. Biswas, S. M., Haque, M., &Misic, J. V. (2010). Privacy and anonymity in VANETs: A contemporary study. *Ad Hoc Sensor Wireless Networks*, *10*(Num 2–3), 177–192.

8. Blau, J. (2008). Car talk-[update], spectrum. *IEEE*, *45*(Num 10), 16-16.

9. Boneh, D., Lynn, B., &Shacham, H. (2001). Short signatures from the weil pairing. *Proceedings of Asiacrypt*, *2248*, 514–532.

10. Boyen, X., & Waters, B. (2006). Compact group signatures without random oracles. In *Advances in cryptology-EUROCRYPT 2006* (pp. 427-444).

11. CAMP Vehicle Safety Communications Consortium. (2005). Vehicle safety communications project: Task 3 final report: Identify intelligent vehicle safety applications enabled by DSRC. National Highway Traffic Safety Administration, US Department of Transportation, Washington DC

12. Chim, T. W., Yiu, S. M., Hui, L. C., & Li, V. O. K. (2012). "MLAS: Multiple-level authentication scheme for VANETs" in Ad Hoc Networks, 10(7), 1445–1456.

13. "Dedicated Short Range Communications (5.9 GHz DSRC)" can be found at http://www.leearmstrong.com/DSRC/DSRCHomeset.htm.

14. Dror, E. (2011). "Vehicular ad-hoc networks VANETs simulator and hierarchical clustering algorithm." Beersheba: Ben-Gurion University of the Negev.

15. Fall, K., Varadhan, K., (2011). "Ns Notes and documents" is accessible at http://www.isi.edu/nsnam/ns/ns-documentation.html.

16. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., & Rabin, T. (2000). "Security arguments for digital and blind signatures" in the Journal of Cryptology, 13(3), 361–396.

17. Guo, J., Baugh, J. P., & Wang, S. (2007). "A group signature-based secure and privacy-preserving vehicular communication framework" in Proceedings of the mobile networking for vehicular environments (pp. 103–108).

18. Horng, S.J., Tzeng, S. F., Pan, Y., Fan, P., Wang, X., Li, T., & Khan, M. K., (2013). "b-SPECS+: Batch verification for secure pseudonymous authentication in VANET" in IEEE Transactions on Information Forensics and Security (pp. 1860–1875).

19. Horng, S. J., Tzeng, S. F., Li, T., Wang, X., Huang, P. H., & Khan, M. K. (2017). "Enhancing security and privacy for identity-based batch verification scheme in VANET" in IEEE Transactions on Vehicular Technology, 66(4), 3235–3248.

20. Hsin-Te, W., Li, W.-S. ,Tung-Shih, S., &Hsiehz, W.-S. (2010). "A novel RSU-based message authentication scheme for VANET" in the Fifth International conference on systems and networks communications, IEEE (pp. 111–116)

21. Huang, D., Misra, S., Verma, M., &Xue, G. (2011). "PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs" in IEEE Transactions on Intelligent Transportation Systems, 12(3), 736–746.

22. Hu, C., Chim, T. W., Yiu, S.-M., Hui, L. C. K., & Li, V. O. K. (2012). "Efficient HMAC-based secure communication for VANETs" in Computer Networks, 56(9), 2292–2303..

23. Hu, H., Lu, R., Huang, C., & Zhang, Z. (2016). "PTRS: A privacy-preserving trust-based relay selection scheme in VANETs" in Peer-to-Peer Networking and Applications, 10, 1204–1218.

24. "IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006) (2013)" can be accessed at https://doi.org/10.1109/IEEESTD.2013.6509896.

25. Lee, C. C., & Lai, Y. Y. (2013). "Toward a secure batch verification with group testing for VANET" in Wireless Networks, 1–9, 1441–1449.

26. Li, J., Lu, H., & Guizani, M. (2015). "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs" in IEEE Transactions on Parallel and Distributed Systems, 26(4), 938–948.

27. Lin, X., Sun, X., Ho, P. H., & Shen, X. (2007). "GSIS: A secure and privacy-preserving protocol for vehicular communications" in IEEE Transactions on Vehicular Technology, 56(6), 3442–3456.

28. Liu, Y., Guo, W., Zhong, Q., & Yao, G. (2017). "LVAP: Lightweight V2I authentication protocol using group communication in VANETs" in International Journal of Communication Systems, 30, e3317.

29. Liu, Y., Wang, L., & Chen, H. H. (2015). "Message authentication using proxy vehicles in vehicular ad hoc networks" in IEEE Transactions on Vehicular Technology, 64(8), 3697–3710.

30. Lu, R., Lin, X., Liang, X., & Shen, X. (2012). "A dynamic privacy-preserving key management scheme for location-based services in VANETs" in IEEE Transactions on Intelligent Transportation Systems, 13(1), 127–139.

31. Kirkpatrick S, Gelatt CD Jr, Vecchi MP. "Optimization by simulated annealing" in Science. 1983 May 13;220(4598):671-80. doi: 10.1126/science.220.4598.671. PMID: 17813860.

32. Basu A, Frazer LN. "Rapid determination of the critical temperature in simulated annealing inversion" in Science. 1990 Sep 21;249(4975):1409-12. doi: 10.1126/science.249.4975.1409. PMID: 17812169.Zheng W. Simulated annealing: an effective stochastic optimization approach to computational library design. Methods Mol Biol. 2004;275:379-98. doi: 10.1385/1-59259-802-1:379. PMID: 15141122.

33. Hasegawa M. (2011). "Verification and correction of the physical analogy of simulated annealing for solving the traveling salesperson problem" in Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, 83(3 Pt 2), 036708. doi: 10.1103/PhysRevE.83.036708. PMID: 21517626.

34. Lu, R., Lin, X., Zhu, H., Ho, P. H., & Shen, X. (2008). "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications" in Proceedings INFOCOM, 2008, 1229–1237.

35. Mao, W. (2003). "Modern cryptography: Theory and practice." Bengaluru: Pearson Education India.

36. Masini, B., Bazzi, A., & Zanella, A. A. (2018). "Survey on the roadmap to mandate onboard connectivity and enable V2V-based vehicular sensor networks" in Sensors, 18(7), 2207.

37. Memon, I. (2015). "A secure and efficient communication scheme with an authenticated key establishment protocol for road networks" in Wireless Personal Communications, 85(3), 1167–1191.

38. Memon, I., Ali, Q., Zubedi, A., & Mangi, F. A. (2016). "DPMM: Dynamic pseudonym-based multiple mix-zones generation for mobile travelers" in Multimedia Tools and Applications, 76, 24359–24388.

39. Memon, I., Arain, Q. A., Memon, H., & Mangi, F. A. (2017). "Efficient user-based authentication protocol for location-based services discovery over road networks" in Wireless Personal Communications, 95, 1–20.

40. Memon, I., Arain, Q. A., Memon, M. H., Mangi, F. A., & Akhtar, R. (2017). "Search me if you can: Multiple mix zones with location privacy protection for mapping services" in the International Journal of Communication Systems, 30, e3312.

41. Memon, I., Hussain, I., Akhtar, R., & Chen, G. (2015). "Enhanced privacy and authentication: An efficient and secure anonymous communication for location-based service using an asymmetric cryptography scheme" in Wireless Personal Communications, 84(2), 1487–1508.Memon, I., Mohammed, M. R., Akhtar, R., Memon, H., Memon, M. H., & Shaikh, R. A. (2014). Design and implement authentication over a GSM system using certificate-less public key cryptography (CL-PKC). *Wireless Personal Communications*, *79*(Num 1), 661–686.

42. Nagel, K., & Schleicher, A. (1994). "Microscopic traffic modeling on parallel high-performance computers" in Parallel Computing, 20, 125–146. https://doi.org/10.1016/0167-8191(94)90117-1.

43. Pointcheval, D., & Stern, J. (2000). "Security arguments for digital and blind signatures" in the Journal of Cryptology, 13(3), 361–396.

44. Pournaghi, S. M., Zahednejad, B., Bayat, M., & Farjami, Y. (2018). "NECPPA: A novel and efficient conditional privacy-preserving authentication scheme for VANET" in Computer Networks, 134, 78–92.

45. Qu, F., Wu, Z., Wang, F., & Cho, W. (2015). "A security and privacy review of VANETs" in IEEE Transactions on Intelligent Transportation Systems, 16(6), 2985–2996.

46. Raya, M., & Hubaux, J. P. (2005). "The security of vehicular ad hoc networks" in the 3rd ACM workshop on Security of ad hoc and sensor networks (pp. 11–21).

47. Raya, M., & Hubaux, J. P. (2007). "Securing vehicular ad hoc networks" in the Journal of Computer Security, 15(1), 39–68.

48. Seo, H., Lee, K.-D., Yasukawa, S., Peng, Y., & Sartori, P. (2016). "LTE evolution for vehicle-to-everything services" in IEEE Communications Magazine, 54(6), 22–28.

49. Shacham, H., & Waters, B. (2007). "Efficient ring signatures without random oracles" in PKC.

50. Shao, J., Lin, X., Lu, R., & Zuo, C. (2016). "A threshold anonymous authentication protocol for VANETs" in IEEE Transactions on Vehicular Technology, 65(3), 1711–1720.

51. Sun, X., Lin, X., & Ho, P. H. (2007). "Secure vehicular communications based on group signature and ID-based signature scheme" in ICC 2007, Moscow, Russia (pp. 1539–1545).

52. Toghi, B., Saifuddin, M.D., Mahjoub, H. N., Mughal, M. O., Fallah, Y. P., Rao, J., Das, S. (2018). "Multiple access in cellular V2X: Performance analysis in highly congested vehicular networks" in IEEE Vehicular Networking Conference (VNC) (pp. 1–8).Vahedi, E., Bayat, M., Pakravan, M. R., & Aref, M. R. (2017). A secure ECC-based privacy-preserving data aggregation scheme for smart grids. *Computer Networks*, *129*, 28–36.

53. Walsh, K., & Sirer, E. (2003). "A staged network simulator (SNS)." Ithaca: Computer Science Department, Cornell University.

54. Wang, F., Xu, Y., Zhang, H., Zhang, Y., & Zhu, L. (2016). "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET" in IEEE Transactions on Vehicular Technology, 65(2), 896–911.

55. Wang, Y., Zhong, H., Xu, Y., & Cui, J. (2016). "ECPB: Efficient conditional privacy-preserving authentication scheme supporting batch verification for VANETs" in the International Journal of Network Security, 18(2), 374–382.

56. Wang, Y., Zhong, H., Xu, Y., Cui, J., & Guo, F. (2017). "Efficient extensible conditional privacy-preserving authentication scheme supporting batch verification for VANETs" in Security and Communication Networks, 9, 5460–5471.

57. Wasef, A., Jiang, Y., & Shen, X. (2010). "An efficient distributed certificate service scheme for vehicular networks" in IEEE Transactions on Vehicular Technology, 59(2), 553.

58. Wu, H.-T., & Hsieh, W.-S. (2013). "RSU-based message authentication for vehicular ad-hoc networks" in Multimedia Tools and Applications, 66(2), 215–227.

59. Xiong, H., Beznosov, K., Qin, Z., & Ripeanu, M. (2010). "Efficient and spontaneous privacy-preserving protocol for secure vehicular communication" in IEEE-ICC, Cape Town, South Africa (pp. 1–6).

60. Xiong, H., Chen, Z., & Li, F. (2012). "Efficient and multi-level privacy-preserving communication protocol for VANET" in Computers and Electrical Engineering, 38(3), 573–581.

61. Xue, X., & Ding, J. (2012). "LPA: A new location-based privacy-preserving authentication protocol in VANET" in Security and Communication Networks, 5, 69–78.

62. Zeng, X., Bagrodia, R., & Gerla, M. (1998). "GloMoSim: A library for parallel simulation of large-scale wireless networks" in Twelfth workshop on parallel and distributed simulation, 1998. PADS 98. Proceedings. IEEE (pp. 154–161).

63. Zhang, C., Lin, X., Lu, R., & Ho, P.-H. (2008). "RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks" in IEEE International Conference on Communications (pp. 1451–1457).Zhang, J., Ma, L., Su, W., & Wang, Y.,

(2007). Privacy-preserving authentication based on short group signatures in vehicular networks. *The first International Symposium on data, privacy, and E-commerce, ISDPE 2007*.

64. Zhang, C., Ho Tapolcai J, P. H., &Tapolcai, J. (2011). On batch verification with group testing for vehicular communications. *Wireless Networks*, *17*(num 8), 1851–1865.

65. RamaGaspar, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons, attributes Kurt Kaiser, CC0, via Wikimedia Commons.

66. Image created by using https://app.diagrams.net/ online tool.

MATLAB Programs

Simulated Annealing

```
clc;
clear;
close all;

T0 = 500 ; % initial temperature
r = 0.997 ; % temperature damping rate
Ts = 1 ; % stop temperature
iter = 300;
plotminroute = zeros(5,103);
model = initModel();
% set(gcf,'unit','normalized','position',[0,0.35,1,0.7]);
set(gcf,'unit','normalized','position',[0.25,0.25,0.5,0.5])
cnt1 = 1;

flag = 0;

% initialization
while(1)
route = randomSol(model);
if(isFeasible(route,model))
    break;
end
end

cost = calculateCost(route,model);
T = T0;

cnt = 1;
minCost = cost;
minRoute = route;

maxIterate = 2100;
costArray = zeros(maxIterate,1);

% Simulated Anneling
while(T > Ts)
    for k = 1:iter
    mode = randi([1 3]);
    newRoute = createNeibor(route,model,mode);
    newCost = calculateCost(newRoute,model);
    delta = newCost - cost;
```

```
    if(delta < 0)
        cost = newCost;
        route = newRoute;
    else
        p=exp(-delta/T);
        if rand() <= p
             cost = newCost;
             route = newRoute;

        end
    end
    end

    costArray(cnt) = cost;
    if cost<minCost
        minCost = cost;
        minRoute = route;
        flag = 1;
    end

    T = T*r; %  annealing
  disp(['Iteration ' num2str(cnt) '
: BestCost = 'num2str(minCost)':CurrentCost = 'num2str(cost)' T = 'num2str(T)]);
    cnt = cnt+1;
   if((cnt==10) || (cnt==1400) || (cnt==1900) || (cnt==500) || (cnt==2))
       plotminroute(cnt1,:) = minRoute;
       cnt1 = cnt1+1;
   end
   figure(1);
   if(flag == 1)
   plotSolution(minRoute,model);
    flag = 0;
   end
%    figure(2);
   subplot(1,2,2)
   plot(costArray);
   xlabel('Iteration')
   ylabel('Temperature')
   pause(0.0001);
end
```

Create Neighbour

```
function newRoute = createNeibor(route,model,mode)
while(1)
    switch mode
        case 1
            newRoute = Swap(route);
        case 2
            % Do Reversion
            newRoute=Reversion(route);

        case 3
            % Do Insertion
            newRoute=Insertion(route);
    end

    if(isFeasible(newRoute,model))
        break;
    end
end
end

function newRoute = Swap(route)
    n = numel(route);

    i = randsample(n,2);
    i1 = i(1);
    i2 = i(2);
    newRoute = route;
    newRoute([i1 i2]) = route([i2 i1]);
end

function newRoute = Reversion(route)
    n=numel(route);

    i=randsample(n,2);
    i1=min(i(1),i(2));
    i2=max(i(1),i(2));

    newRoute=route;
    newRoute(i1:i2)=route(i2:-1:i1);

end
```

Init Model

```
function model = initModel()
city = 100 ; % city number except depot
veh = 4 ; % vehicle number
speed = [10 15 12 8];
model.city = city;
model.veh = veh;
model.speed = speed;
xmin=0;
xmax=200;
ymin=0;
ymax=200;
maps = zeros(city+veh,city+veh);
x=randi([xmin xmax],1,city);
y=randi([ymin ymax],1,city);
x0 = 100;
y0 = 100;
model.x0 = x0;
model.y0 = y0;
% for showing multiple vehicles
offset = 30;
x1=randi([xmin xmax/2-offset],1,city/4);
x2=randi([xmin xmax/2-offset],1,city/4);
x3=randi([xmax/2+offset xmax],1,city/4);
x4=randi([xmax/2+offset xmax],1,city/4);
y1=randi([ymin ymax/2-offset],1,city/4);
y2=randi([ymin ymax/2-offset],1,city/4);
y3=randi([ymax/2+offset ymax],1,city/4);
y4=randi([ymax/2+offset ymax],1,city/4);
x = [x1 x2 x3 x4];
y = [y1 y3 y2 y4];
for k = 1:veh
x = [x x0];
y = [y y0];
end
model.x = x;
model.y = y;
n = city+veh;
for i = 1:n
   for j = i:n
      maps(j,i) = sqrt((x(i)-x(j))^2+(y(i)-y(j))^2);
      maps(i,j) = maps(j,i);
   end
end
model.maps = maps;
end
```

Calculate Cost

```
function res = calculateCost(route,model)
    city = model.city;
    veh = model.veh;
    maps = model.maps;


    route = [city+veh route city+veh];
    res = 0;

    for i = 1:length(route)-1
        res = res + maps(route(i),route(i+1));
    end


end
```

Is Feasible

```
function res = isFeasible(route,model)
    len = length(route);
    if route(1)>model.city||route(len)>model.city
        res = 0; return
    end

    for i = 2:len
        if route(i)>model.city && route(i-1)>model.city
            res = 0; return
        end
    end

    res = 1;
end
```

Random solution

```
function res = randomSol(model)
    res = randperm(model.city+model.veh-1);
end
```