Supply Chain and Operations Decision Making

www.scodm.reapress.com

Sup. Chain Oper. Decis. Mak. Vol. 2, No. 3 (2025) 115–126.

Paper Type: Original Article

IoT-Driven Intelligent Traffic Systems for Congestion Management

Nikhil Singh[1,*], Seyyed Ahmad Edalatpanah[2,3,4], Mahmoud Alimoradi[5]

[1] School of Computer Science Engineering, KIIT University, Bhubaneswar, India; 2229131@kiit.ac.in.
[2] Department of Applied Mathematics, Ayandegan University, Tonekabon, Iran.
[3] Department of Mathematics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai-602105, India.
[4] University College, Korea University, Seoul 02841, South Korea; saedalatpanah@gmail.com.
[5] Department of Computer Engineering, Ayandegan Instite of Higher Education, Tonekabon, Iran; Mahmoud.alimoradi@aihe.ac.ir.

Citation:

Received: 17 May 2025 *Revised: 21 July 2025* *Accepted: 01 September 2025*	Singh, N., Edalatpanah., S. A., & Alimoradi, M. (2025). IoT-driven intelligent traffic systems for congestion management. *Supply chain and operations decision making*, 2(3), 115-126.

Abstract

Utilization of Artificial Intelligence (AI) techniques in intelligent transportation systems opens up new dimensions in choreographing sustainable urban mobility. However, one of the main issues concerns the appropriate context or situation where such techniques ought to be adopted. They have several alternatives, including the utilization of cloud computing, fog computing, edge computing, or even their mobile devices. A smart traffic management system based on the Internet of Things (IoT) concept is proposed in this paper. We optimize the use of evolutionary algorithms, starting with the Lightweight Random Early Detention (LRED) for Vehicles Dynamic (VD) mechanism. LREDfor VDs can be employed in controlled junctions to clear oncoming traffic and optimize the cycle and phases of the traffic lights. Then the authors explain that after LRED for VDs has been successfully optimized in a non-real-time environment, it is possible to deploy the approach to an unknown traffic situation without the need to involve AI in edge IoT devices. The versatility of this mechanism is extensively assessed using the traffic simulation package, SUMO. iREDVD outperforms all other competing designs since it minimizes the waiting time of vehicles, average travel time, fuel usage, and emission of solid and gaseous pollution, among other benefits.

Keywords: Intelligent transportation systems, Smart traffic management, Internet of Things, Evolutionary algorithms, Traffic optimization.

1 | Introduction

The Internet of Things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over

Corresponding Author: 2229131@kiit.ac.in
https://doi.org/10.48313/scodm.v2i3.37

the Internet. IoT enables companies to automate processes and reduce labour costs. It further alleviates wastage and enhances service provision, making manufacturing and delivery of goods less expensive, and also bringing accountability to the dealings of customers [1].

AI, which stands for artificial intelligence, refers to systems or machines that perform continuous tasks and can improve themselves based on the information they collect. AI is considered a human machine because it collects its own data and operates without human interference. For instance, Siri on our phone once claimed that this number was for my mom, then immediately added it to the phone directory and continued to use it when we asked Siri to call mom [2].

The interplay between IoT, Big Data, and Cloud Computing presents companies with rich possibilities for business expansion. In simple terms, IoT is the data reservoir, Big Data is the data processing or analytics, and Cloud Computing handles storage, scaling, and access speed [3].

2 | Related Works

Two main branches exist within evolutionary computation. One utilizes genetic algorithms for the representation of traffic light phases and timing. These are represented as a set of chromosomes. Optimisation is direct on these chromosomes. It is possible that they can be included within the fuzzy logic.

Furthermore, a second approach utilises GA as an optimisation algorithm to enhance the traffic light control system [4]. This system has a large number of parameters. It would otherwise be unfeasible to optimise. The main advantages of these algorithms are two. They require a significant computational load for both training and parameter adjustment. However, this disadvantage can be mitigated. Working in the cloud during the training phase makes it easier to update the algorithm. This update can be carried out once the training phase is complete. In the first group, an extraordinary piece of work stands out. GA was used for encoding a fuzzy logic controller in chromosomes.

The optimal parameters are determined based on the number of cars waiting. Similar works on optimizing traffic light timing are presented. These works utilized GA. An approach linking GAs with device communication has been identified. This D2D approach is functional [4]. It collects Information from Sensors and actuators. The aim is also to reduce response time. Concerning the Second group, our work is encompassed within this one. We use a GA The GA optimizes a traffic light control algorithm. The method has an advantage. It shifts the complexity of the search to a training scenario. Parameters of the control Algorithm are learned with great ease. This allows the adaptation of the control algorithm. It works well in multiple scenarios [5]. Changes in the environment are accounted for with ease. The control algorithm is capable of acting in advance. The aim is to study the scenario beforehand. Other works utilize optimisation algorithms, which bear a similarity to GAs. For example, this includes Ant Colony Optimization (ACO) algorithms. An ACO algorithm is probabilistic.

It is a technique for solving computational problems. It can be used to find good paths through graphs. Within this category, work done by Rehman et al. [6] and Jerry et al. [7] stand out. Both solve the traffic congestion Problem using ACO. In contrast to Previous works in the existing literature, our focus is on optimizing an adaptive traffic light control algorithm [5]. This algorithm is REDVD. We use a GA to achieve optimisation. Our improved Version of the algorithm is called iREDVD. The novelty is the use of GAs for the optimisation process.

This use makes it easier to set the best values for configuration parameters. To our knowledge, contrary to earlier works, most related literature includes GAs in traffic Management procedures themselves. We use it only as an offline step. Later, we demonstrate something. After the Parameters are set, Iredvd works well. It can be used under unknown traffic scenarios. It maintains a lightness feature. Also, it avoids the burden of AIT. This means it can be used in IoT devices [5].

3|Random Early Detection for Vehicles

Upon reflection of the preceding text, we find various techniques for controlling traffic light intersections. This section now displays the foundation for ITS. This proposition is in document [5]. REDVD, the Basis for this piece is its basis. REDVD operates on the principle of Random Early Detection or RED [8], [9]. This algorithm is well-known for controlling congestion in communication networks. It's a technique for managing congestion in packet-switching networks. RED focuses on early detection and avoidance of congestion. By doing so, it can elevate the throughput rate. Congestion is avoided by implementing a strategy of randomly discarding packets. This depends on the number of Packets waiting to be processed in the router. Some packets can potentially be dropped. This can happen if the packet queue that is ready to be processed surpasses a fixed threshold. The probability of dropping packets increases as the Queue size goes up. Thus, congestion can be bypassed even before it happens. The original red for traffic control in urban landscapes was introduced. This was in document [5].

It was named RED for vehicles or REDV. The REDV model modifies the RED algorithm. It caters to the peculiarities of vehicles and signaled intersections. The idea behind it is as follows. In REDV, the number of vehicles waiting to cross at each intersection branch is taken as equivalent. This is the number of packets waiting to be processed in a router. Instead of dropping a packet, REDV perceives the probability of packet loss.

This probability depends on the number of packets to be processed. REDV adjusts the green time of every intersection branch. This probability is intersection-specific and depends on the Number of cars waiting. When many cars use a branch, the green Time frequently increases. In contrast, when there are only a few cars to be processed, the green time is usually reduced. The total cycle of intersections is a fixed value. REDV demonstrated strong performance. It was tested as an adaptive control system. It proactively managed incipient Congestion at isolated intersections.

Handling alterations in traffic flow Rate was the primary focus. The traffic flow rate includes the number of vehicles that cross the intersection per unit of time. It was vital to Remodel the total cycle. The goal was to provide long cycles to accommodate a large volume of traffic. Conversely, short cycles were provided when the volume of Traffic was low. The same authors also released REDV Dynamic, also known as REDVD. This was noted in [10], where the Traffic metric Performance Improved with this version. The downside was the addition of new adjustable parameters. These Parameters required proper tuning for Peak algorithm performance. Every REDVD configuration parameter is included in *Table 1*. These new parameters contribute to traffic control flexibility. The parameters are adaptive to changing traffic conditions. Optimizing the large set is not a trivial task. The [5] values were empirically obtained for configuring REDVD. Due to this aspect, this work has A key objective. The goal is to get the optimal configuration.

The configuration should provide the best performance by reducing the average waiting time per intersection for every vehicle. The optimization known as EA will be carried out. The EA optimizes using GA, which is the most popular type of EA. The iREDVD is an optimized version. The optimization process is outlined next. The EA focuses on optimizing parameters. The parameters are present in the REDVD. These parameters aim to increase the potential of the algorithm. Empirical data from [5] provides a foundation for the optimization. The data involves each parameter's range and correlation. The iREDVD variation stands as the optimal algorithm. The optimization process ensures iREDVD has the best possible performance. The results confirm the effectiveness of the optimization process. The current section provides an overview of this process.

Table 1. REDVD parameters.

Parameter	Description
Minth	The minimum threshold of vehicles to increase the green-time phases probabilistically.
Maxth	The maximum threshold of vehicles to increase the green-time phases probabilistically.
Delta	The increase/decrease of the green-time phase, either because it is above/below the maxth/minth thresholds, respectively, or due to a probabilistic increase when it falls between these thresholds.
Min_greentime	The minimum time for a green-time phase.
Max_greentime	The maximum time for a green-time phase.
Liminc	Number of consecutive times the green-time phase must be increased, taking into account both arteries to increase the cycle for all phases.
Limdec	Number of consecutive times the green-time phase must be decreased, taking into account both arteries to decrease the cycle for all phases.
Delta_cycle	The increase/decrease of the cycle for all phases.
Min_cycle	The minimum cycle for all phases.
Max_cycle	The maximum cycle for all phases.
Wq	A factor that determines the weight of the historical value versus the current value in the queue length.
Maxp	Factor used in the RED algorithm.

4 | Optimisation Process

A genetic algorithm is a series of mechanisms that hark back to the principles of evolution. It's employed to optimize a set of parameters inherent in a problem. It enables optimization of that problem. This Problem is by nature complex. It is carried out using procedures that are Easy to perform. Also, these procedures have a low computation load [11], [9]. Thus, this results in a rapid parameter optimization. The general procedure comprises four phases. These are the initialization of the population, fitness calculation, selection, and crossover. These phases create a generation. The complete optimization process will involve the necessary number of generations. A population constitutes a set of individuals. Each individual contains an enormous number of genes. The exact number is determined by the number of parameters that need optimization. In our case, these parameters are listed in *Table 1*. Each gene carries the coded information of each parameter. For instance, Maxp Parameter is allotted a float number between 0 and 1 in our case. To illustrate further,

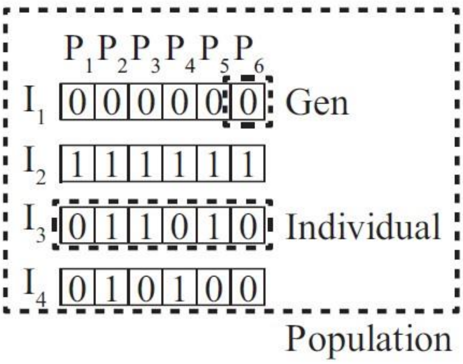

Fig. 1. Example of a population of a GA, which consists of four individuals and six binary parameters to be optimised.

if we're considering a problem with six binary parameters to optimize, such as the knapsack problem [12], an example of a population is possible. This is shown in *Fig. 1*. In this figure, you can see a population of four individuals. These Individuals are referred to as I1 to I4. Each of these individuals has six parameters. These parameters are P1 to P6. Each parameter is coded in binary. The code is either 0 or 1. Each of these parameters is represented in each gene. Best-adapted individuals to the environment Are Selected for each generation. This is so that the next generation Individuals can contain their genes. The term "better adapted to the environment" is defined by the fitness function, which provides a numerical measure.

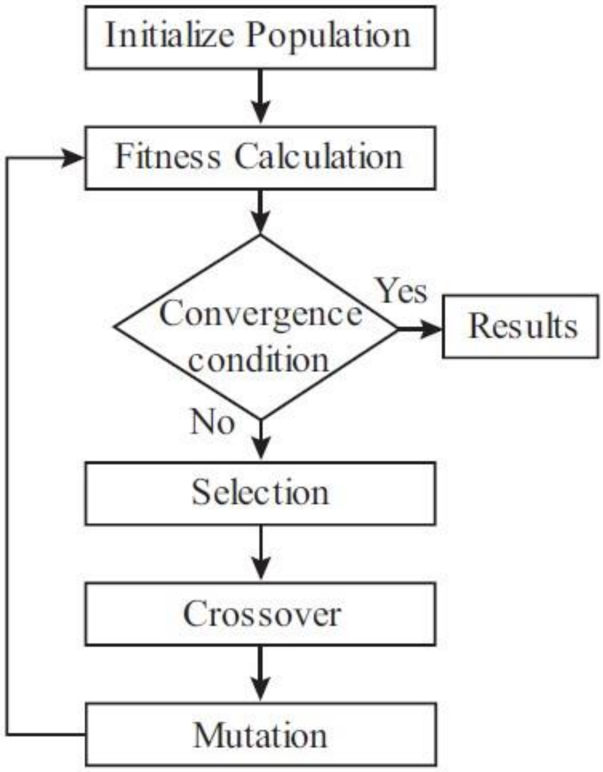

Fig. 2. A general GA procedure. The convergence condition can be different, either several entire generations or a fitness value improvement after a few generations.

It gauges how well an individual can adjust to the Problem being optimized. Imagine tasks of humanoid robots learning to walk. In this case, the genes encode the movement of joints and muscles. The fitness function will become the distance traveled. Individuals who have journeyed the farthest are likely to be chosen.

Finally, we will delve into the four phases of the general optimization procedure (see *Fig. 2*).

4.1|Initialize Population

Process initiation based on Random genes significantly reduces convergence time. Alternative methods exist To Shorten the process. These methods are essential to explore. They can be used to follow a different pattern. Despite this, they may not necessarily be optimal [13], [14]. This warrants a comparison. Population size plays a massive role in the GA. It guarantees More diversity. It ensures a vast section of the parameter space is probed. However, there's A trade-off. A significant population easily allows for wide Genetic diversity. It allows for a broad parameter space to be explored randomly. However, there are negatives to consider. Achieving fitness for all individuals can take a substantial amount of time. The focal problem significantly influences population size. To cover a broad parameter range, we may need to increase the population size exponentially. A key suggestion is to have twice as many individuals as parameters under study. Typically, population sizes range from 20 to 100 people [11], [15], [16].

4.2 | Fitness Calculation

Every single member of the population gets reviewed. What do we find? In its parameter settings, we find the fitness level. Imagine we are making a robot. This robot should walk. Here is what the fitness of each member means. It's the distance it can walk. Or in problems such as the knapsack problem, the concept is similar. In the knapsack problem [17]. The benefit obtained by objects inside the knapsack that minimize weight is the adjustment. Upon considering our case, the scenario is slightly different. The concept is similar. However, it will be the average wait time for each vehicle. This wait time is at every intersection. For all vehicles used, this figure is needed. This will aid in assessing the congestion levels. This would not be possible if individuals were not evaluated. The evaluation gives an insight into the fitness level. Individuals are judged based on their ability to carry out specific tasks. These are serious considerations when designing a transportation system. There must be a smooth, efficient flow of traffic. Evaluation of every individual in the population is a crucial aspect of this process. Ultimately, we have a metric to assess the performance of the system [12]. This is the average waiting time at each intersection. The waiting time at every intersection is a critical consideration. It's an essential piece of information to determine the efficiency of traffic flow. Vehicles must not come to a standstill. Waiting times must be kept to a minimum. Our evaluation of the population provides a clear picture of this scenario. By utilizing these parameters, we can design an improved transportation system that effectively addresses congestion.

4.3 | Selection

After all the individuals constituting the population have had their abilities examined, the next step is to select the group that exhibits the best adaptation to its surroundings. The Methods to pick the crème de la crème range widely. Roulette wheel selection is one such mechanism; Boltzmann selection is another. Tournament selection and rank selection are more. We also find steady-state selection and elitist selection. There are several others as well. One of the simplest and most effective Strategies is as follows [18–20]. We select a certain percentage of the individuals. They possess the highest level of fitness. With this mechanism, it is possible to gather the best-adapted individuals. They fit the environment of the current generation. Parents are what we call these Selected individuals. They will be the ones generating the children. The percentage of selective individuals plays a crucial role in the correct functioning of the GA. A very high value will hinder rapid convergence. However, a very low value will also lead to low genetic diversity. A high probability of converging towards a local minimum exists [11], [15], [16]. Typical values for this percentage occur in a Range of 5 to 10. These values vary with population size. The remaining individuals are known as children. These parents will produce them. The total of the population will be obtained this way. For this, we employ crossover and mutation mechanisms. Although sophisticated methods exist for the Genetic Algorithm's selection process, rank selection is preferable in initial generations. This is to get quick preliminary results. More sophisticated techniques are found elsewhere. [20].

4.4 | Crossover

After all the individuals constituting the population have had their abilities examined, the next step is to select the group that exhibits the best adaptation to its surroundings. The Methods to pick the crème de la crème range widely. Roulette wheel selection is one such mechanism; Boltzmann selection is another. Tournament selection and rank selection are more. We also find steady-state selection and elitist selection. There are several others as well. One of the simplest and most effective Strategies is as follows. We select a certain percentage of the individuals. They possess the highest level of fitness. With this mechanism, it is possible to gather the best-adapted individuals. They fit the environment of the current generation. Parents are what we call these Selected individuals. They will be the ones generating the children [21–23]. The percentage selectivity of individuals plays a crucial role in the correct functioning of GA. GA: A very high value will hinder rapid convergence. However, a very low value will also lead to low genetic diversity. A high probability of converging towards a local minimum exists [24]. Typical values for this percentage occur in a Range of 5 to 10. These values vary with population size. The remaining individuals are known as children. These parents will produce them. The total population will be obtained this way. For this, we employ crossover and mutation mechanisms. Although sophisticated methods exist for the Genetic Algorithm's selection process, rank selection is preferable in initial generations. This is to get quick preliminary results. More sophisticated techniques are found elsewhere [11], [15], [16].

4.5 | Mutation

To achieve high genetic diversity, apply mutation mechanisms to a minority of genes. This is in the population. With this mutation, the value of every gene may fluctuate randomly with low probability. It prevents the GA from steadily moving towards a local best outcome. Then, it opens up other gene combinations to be randomly examined. Not too frequently, however, as it would turn into a random hunt. The mutation rate indicates how often a gene is altered. When it's 0% the genes of offspring are stable. If the rate is 100%, all Offspring genes Are Modified within A particular range. This range is less than 20% of the permissible range for each gene. The rate of mutation must be between 0.1% and 1%. It allows exploration. This exploration leads to a wider parameter space. This Way, genetic diversity is achieved, and at the same time, random [48], [49]. Parameter space exploration is avoided. The mutation rate must decrease. The margin of change in parameters must also decrease as the generations are formed. The goal is to achieve a swift convergence. This process is shown in *Fig. 2*. It depicts the Procedure followed by the GA up to the Convergence point. For iREDVD, we settle on a population size of 256 individuals [11], [15], [16]. We have 16 selected parents. The Crossover rate is at 80%. The Mutation Rate is 0.1%. Generation count can go up to 20.

5 | Performance Evaluation Results

5.1 | Training Scenario

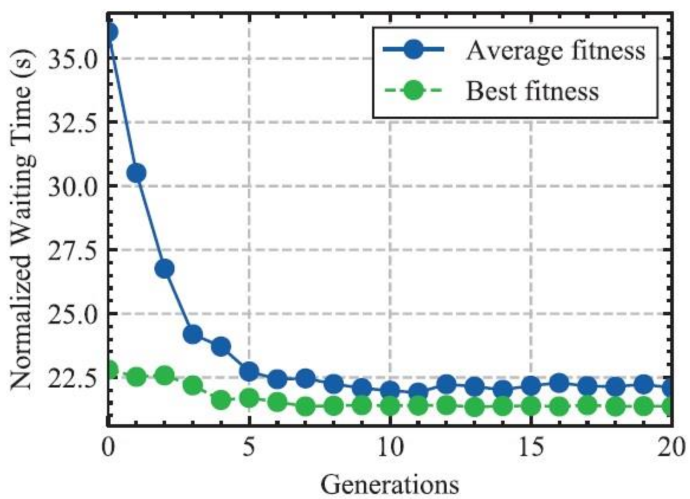

Fig. 3. Optimization process. "Best fitness" is the individual with the lowest normalised waiting time, and "Average fitness" is the average normalised waiting time.

Waiting time for the entire population. The X-axis represents the generations.

Fig. 3 sketches a picture. It shows the value of the normalized Waiting time. This is for vehicles per intersection. The figures are for the optimization process. The GA executes it in the training phase. As can be seen, after around five Generations, the GA hits promising values. This is for almost all simulated individuals. The simulation could have been concluded at that point. But it was left running for more. This was done to find a strong set of parameters. These are abilities that, with Time, can prove their superiority over others. The time needed for each simulation was recorded. This was part of the optimization Process for the test scenario. It was roughly 1 min. This indicates simulating 256 individuals for 21 generations on a

Server with 16 processing cores [5]. This required about 48 hours. Each simulation was repeated 10 times. After that, the average value and standard deviation of each variable were determined. Post that, the best parameter configuration was implemented. Execution of the Traffic light control algorithm is nearly instant. It no longer requires an artificial intelligence algorithm. It's based on A light method. A fast method. A quick method! Low-power IoT devices can swiftly execute this method.

For further information, one can look at [5]. Now, the values of the optimized parameters after training are displayed in *Table 4*. This reveals some interesting numbers during the optimization process. The parameter limdec denotes The Number of times the green time of Traffic lights is reduced. The purpose is to decrease the total cycle time at the intersection. If limdec is 1, even a minor decrease in traffic prompts a reduction in total cycle time. This ultimately also reduces the green time. The opposite holds for an Increase in Total cycle time. The parameter liminc controls the increase.

The total cycle time is that of the traffic lights. If the traffic increases, this cycle Time also increases. A high value indicates a cautious approach [5]. It points to the fact that Iredvd must be certain before an increase in total cycle time. So, when traffic is levied, the total cycle time of the traffic lights is only to be incremented when the certainty of iREDVD is at a high level. Data from the training scenario is tabulated. Refer to it as *Table 5*. It supports the idea of an improvement in iREDVD. This is more than 50%. The improvement is in the average waiting time. The vehicles at every intersection experience the same issue.

Let us compare it to the traditional algorithms, specifically Fixed 30, Fixed 45, and Fixed 60. Additionally, GreenWave 30, GreenWave 45, and GreenWave 60 are considered for comparison. Our improvement over the original REDV algorithm proposed in [5] remains visible. It underlines the need for proper Functionality of the algorithm to adjust parameters. Compared to the original REDVD variant found in [5], there is a 27% improvement. An improvement means a reduction of over 8 seconds in average normalized waiting time. Regarding other Key performance metrics, such as average travel time and speed, we can see further improvement. Average CO emissions have decreased by 13% to 46%.

Average fuel consumption decreases 7% to 18%. All of these KPI show improvements with iREDVD. It ranges from 7% to 45%. Notably, speed increases by 16 to 26%. There are decreases in other metrics. The average CO emissions and average fuel consumption have decreased by 7% to 18%. You see these compared to traditional algorithms [5].

Table 2. Optimised parameters and values.

Parameter	Optimal Value
Minth	10
Maxth	25
Delta	10
Min_greentime	15
Max greentime	60
Liminc	5
Limdec	1
Delta_cycle	15
Min_cycle	45
Max_cycle	130
Wq	0.7
Maxp	0.85

5.2|Testing Scenarios

Upon completion of the training, we applied stress to the algorithm. This happened during testing scenarios. We did use the optimized values.

We used optimized values for the configuration parameters. Results in Tables 6 and 7 indicate improvements. Improvements are very close to those obtained during the training phase. This confirms something. We confirmed the iREDVD's ability. iREDVD can adjust to unfamiliar conditions. These are conditions it was not trained for.

It maintains its robustness. It can be used in real deployments. Consider a case in point. Consider the Software's performance on two tests. In Test 1, iREDVD achieves a reduction in waiting time. They achieve 34% to 49% reduction compared to other control techniques. In Test 2, the results are more impressive. The iREDVD achieves a reduction of 78% to 82%. How massive is this decrease? Think about a few things. Think about the original waiting time. Now imagine a 6 to 100-second drop in that waiting time at every intersection. That's the equivalent. The improvements are not Limited to reductions. Other metrics reflect improvement. Consider the reduction of carbon monoxide emissions. They go down by around 25%. Then there is the reduction in average fuel. It goes down between 6 to 17%. The iREDVD software not only offers time savings but also makes a significant contribution to overall environmental health. It does so by improving vehicular efficiency. Its impact is seen not just in time saved, but also in Reduced emissions.

That impact is felt in lowered fuel consumption. Implications of the iREDVD software are wide-ranging. They're not limited to a single sphere. The algorithm can be applied to many scenarios. This robustness is a quality that sets it apart. It's a quality that makes it a candidate for use in real-world deployments

5.3|Final Remarks

The exceptional performance of fine-tuned iREDVD is due to its optimized design. The ability to adjust to road Conditions is behind it. We can find it in *Fig. 10.a* And Also in *Fig. 10.b*. There, you Can See how iREDVD proactively adjusts the cycle time in *Fig. 10.a* during training. There's also an adaptation during unknown Circumstances in testing *Fig. 10.b*. The cycle lengthens when more vehicles pass through the intersection, and conversely, it shortens.

Further insight is provided in *Fig. 10.a*. And there's also more in *Fig. 10.b*. The time for each branch is separate. Traffic doesn't matter. It's independent. iREDVD adjusts to traffic conditions. It accounts for asymmetric traffic. See *Fig. 10.b* at Hour 4. At that Point, iREDVD acquires a different time for each branch. This is key. It self-adjusts to traffic conditions. Independence drives iREDVD to a more efficient time distribution. This also leads to a reduction in waiting time

6|Conclusion

In the Present era, the vehicle population is rapidly on the rise. Large urban zones are tending towards massification. The development of the IoT is gaining ground. Hence, the need for effective orchestration in smart city constructions intensifies. The application of ITS ensures smooth traffic management. This occurs when it is used at signal-controlled intersections. It affects the improved flow of vehicles. When used in this manner, it leads to reduced pollutants and Fuel consumption. In this work, we have detailed the optimization and Function of the iREDVD ITS method.

In this method, an EA is employed as the optimization mechanism. iREDVD is a lightweight, fine-tuned, and quick algorithm that can Be Utilized in IoT devices. These devices have lower requirements compared to traditional devices. It controls intersections, which means traffic must be managed. The process is carried out in a highly effective manner. Our work has shown that iREDVD is better than tried-and-trusted traffic light control methods. It has also surpassed its previous versions. If we make a comparison with traditional control methods, there is a significant difference. The time vehicles spend waiting at each intersection is reduced. It's been reduced by 24 to 100 seconds.

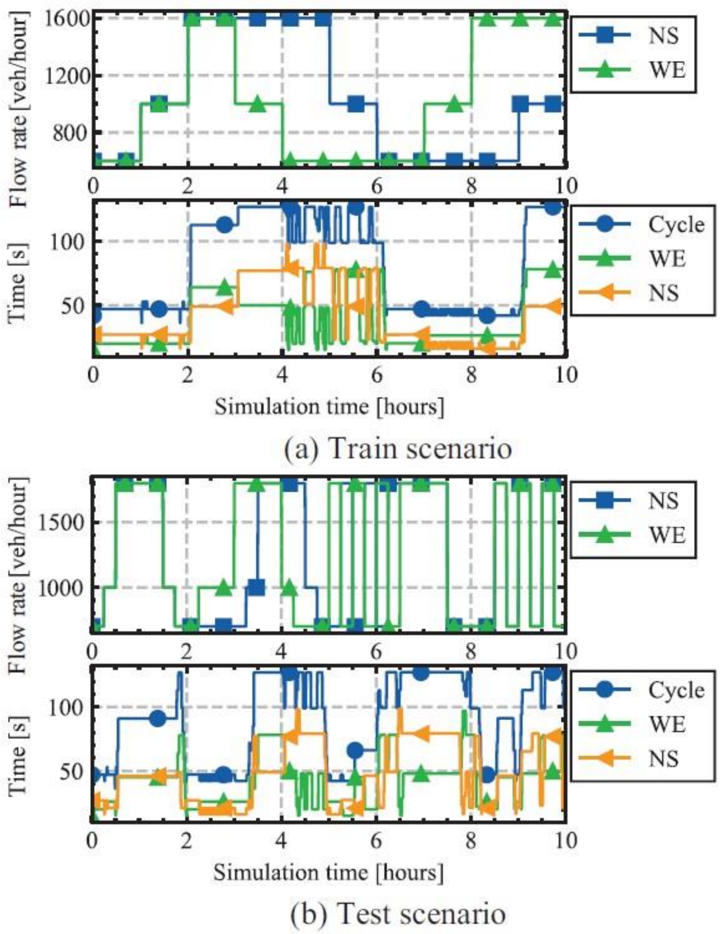

Fig. 4. Vehicular flow rate simulated, time for each branch, and total cycle for intersection 1 vs. simulation time: a. train scenario, b. test scenario

This is observed when a vehicle passes through it. It's equivalent to a 50% to 80% reduction in the waiting time of vehicles at traffic lights. Original REDVD has Long waiting times at every intersection. It's reduced by more than 27% in iREDVD. iREDVD has reduced emissions of pollutants. The reduction is 7%–38%. It also has reduced fuel consumption. This reduction is 7%–14%.

In conclusion, the use of these ITS will allow Future smart cities to coordinate different actors better. This will result in an increase in Pollution, which will be diminished. And, transportation Systems will be fine-tuned. Future work is being conducted on the integration of ITS with 5G technology. The purpose is to create a completely connected traffic management ecosystem.

Acknowledgments

Acknowledgements enable you to thank all those who have helped in carrying out the research. Careful thought needs to be given concerning those whose help should be acknowledged and in what order. The general advice is to express your appreciation in a concise manner and to avoid strong emotive language.

Author Contributions

Nikhil Singh: Integrating insights on conceptualization, methodology, data analysis, research validation, and limitations.

Funding

This research received no external funding.

Data Availability

The data analyzed in this study can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

These sections should be tailored to reflect the specific details and contributions, if necessary.

References

[1] Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. *IEEE transactions on intelligent transportation systems*, *12*(4), 1624–1639. https://doi.org/10.1109/TITS.2011.2158001

[2] Murray, R. M. (2002). Panel on future directions in control, dynamics, and systems. *2*(02). http://www.cds.caltech.edu/~murray/cdspanel/

[3] Cano, M. D., Sanchez-Iborra, R., Garcia-Sanchez, F., Garcia-Sanchez, A. J., & Garcia-Haro, J. (2016, July). Coordination and agreement among traffic signal controllers in urban areas. *2016 18th international conference on transparent optical networks (ICTON)* (pp. 1-4). IEEE. https://doi.org/10.1109/ICTON.2016.7550368

[4] Floyd, S., & Jacobson, V. (2002). Random early detection gateways for congestion avoidance. *IEEE/acm transactions on networking*, *1*(4), 397–413. https://doi.org/10.1109/90.251892

[5] Sanchez-Iborra, R., & Cano, M.-D. (2017). On the similarities between urban traffic management and communication networks: Application of the random early detection algorithm for self-regulating intersections. *IEEE intelligent transportation systems magazine*, *9*(4), 48–61. https://doi.org/10.1109/MITS.2017.2743202

[6] Rehman, A., Mazhar Rathore, M., Paul, A., Saeed, F., & Ahmad, R. W. (2018). Vehicular traffic optimisation and even distribution using ant colony in smart city environment. *IET intelligent transport systems*, *12*(7), 594–601. https://doi.org/10.1049/iet-its.2017.0308

[7] Jerry, K., Yujun, K., Kwasi, O., Enzhan, Z., & Parfait, T. (2015). NetLogo implementation of an ant colony optimisation solution to the traffic problem. *IET intelligent transport systems*, *9*(9), 862–869. https://doi.org/10.1049/iet-its.2014.0285

[8] Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer*, *27*(6), 17–26. https://doi.org/10.1109/2.294849

[9] Roberge, V., Tarbouchi, M., & Labonté, G. (2012). Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE transactions on industrial informatics*, *9*(1), 132–141. https://doi.org/10.1109/TII.2012.2198665

[10] Mamdani, E. H., & Pappis, C. P. (1977). A fuzzy logic controller for a traffic junction. *IEEE transactions on systems, man, and cybernetics*, *7*(10), 707–717. https://doi.org/10.1109/TSMC.1977.4309605

[11] Srinivas, M., & Patnaik, L. M. (2002). Genetic algorithms: A survey. *Computer*, *27*(6), 17–26. https://doi.org/10.1109/2.294849

[12] Chu, P. C., & Beasley, J. E. (1998). A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics*, *4*, 63–86. https://doi.org/10.1023/A:1009642405419

[13] Viveros-Jiménez, F., León-Borges, J. A., & Cruz-Cortés, N. (2014). An adaptive single-point algorithm for global numerical optimization. *Expert systems with applications*, *41*(3), 877–885. https://doi.org/10.1016/j.eswa.2013.08.018

[14] Toğan, V., & Daloğlu, A. T. (2008). An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Computers & structures*, *86*(11–12), 1204–1218. https://doi.org/10.1016/j.compstruc.2007.11.006

[15] Johnson, J. M., & Rahmat-Samii, V. (1997). Genetic algorithms in engineering electromagnetics. *IEEE antennas and propagation magazine*, *39*(4), 7–21. https://doi.org/10.1109/74.632992

[16] Venkatesh, P., Gnanadass, R., & Padhy, N. P. (2003). Comparison and application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints. *IEEE transactions on power systems*, *18*(2), 688–697. https://doi.org/10.1109/TPWRS.2003.811008

[17] Dip, G., Prahlad, V., & Kien, P. D. (2009). Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed. *Robotica*, *27*(3), 355–365. https://doi.org/10.1017/S026357470800475X

[18] Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms* (Vol. 1, pp. 69–93). Elsevier. https://doi.org/10.1016/B978-0-08-050684-5.50008-2

[19] Blickle, T., & Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary computation*, *4*(4), 361–394. https://doi.org/10.1162/evco.1996.4.4.361

[20] Jebari, K., Madiafi, M., & others. (2013). Selection methods for genetic algorithms. *International journal of emerging sciences*, *3*(4), 333–344. https://www.researchgate.net/profile/Khalid-Jebari/publication/259461147_Selection_Methods_for_Genetic_Algorithms/links/5480d9100cf22525dcb60519/Selection-Methods-for-Genetic-Algorithms.pdf

[21] Kora, P., & Yadlapalli, P. (2017). Crossover operators in genetic algorithms: A review. *International journal of computer applications*, *162*(10), 34-36. https://www.researchgate.net

[22] Magalhaes-Mendes, J. (2013). A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. *WSEAS transactions on computers*, *12*(4), 164–173. https://www.wseas.com/journals/computers/2013/5705-156.pdf

[23] Abdoun, O., & Abouchabaka, J. (2012). A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. *International journal of computer applications, 3*(11), 49-57. https://doi.org/10.48550/arXiv.1203.3097

[24] Spears, W. M., & Anand, V. (1991). A study of crossover operators in genetic programming. *International symposium on methodologies for intelligent systems* (pp. 409-418). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-54563-8_104