Received: 25 May 2020 | Revised: 28 September 2020 | Accepted: 19 October 2020

DOI: 10.1049/itr2.12021

ORIGINAL RESEARCH PAPER

Intelligent IoT systems for traffic management: A practical application

Antonio Guillen-Perez | Maria-Dolores Cano

Information and Communication Technologies Department, Universidad Politécnica de Cartagena, Cartagena, Murcia 30203, Spain
(Email: mdolores.cano@upct.es)

Correspondence
Information and Communication Technologies Department, Universidad Politecnica de Cartagena, 30203, Spain.
Email: antonio.guillen@edu.upct.es

Abstract
The incorporation of Artificial Intelligence algorithms in Intelligent Transportation Systems gives rise to new opportunities for a more sustainable urban mobility. However, one of the main challenges is to decide when and where these techniques should be applied; several options appear, such as cloud computing, fog computing, edge computing, or even edge devices. In this paper, an Internet of Things-based solution for smart traffic management is presented. Using the lightweight Random Early Detection for Vehicles Dynamic mechanism as a basis, we optimize using evolutionary algorithms. Random Early Detection for Vehicles Dynamic can be applied in signaled intersections to proactively detect incipient congestion and set the best cycle and phases of traffic lights. Then, the authors demonstrate that once Random Early Detection for Vehicles Dynamic has been appropriately optimised offline, it can be later used in unknown traffic scenarios without the burden of applying Artificial Intelligence in constrained Internet of Things devices. The performance of this mechanism is widely tested with the SUMO simulation tool. Results show that this improved version, called iREDVD, notably reduces the vehicles' waiting time, average trip time, fuel consumption, and emission of particles and gaseous pollutants compared with other proposals.

1 | INTRODUCTION

Intelligent Transportation Systems (ITS) are witnessing a paradigm shift in their modus operandi. Intelligence, understood as the capability of a system to assist, manage, and make decisions to improve key performance metrics, is entering a new reality. One of the reasons lies in prompt advances in traffic data acquisition [1]. This increasing amount of available data allows better and more responsible use of resources. And the origin of these data is vast. Still, in general, we can affirm that either from sensor devices or from Road Side Units (RSU), in-vehicle communication, Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Everything (V2X), etc. is being transformed thanks to IoT devices. These IoT devices can be embedded in practically any system or device, offering the ability to communicate, provide information, or automate tasks. Besides, these IoT devices have low energy consumption, which allows them to be integrated into battery-powered devices, thus extending their useful life.

Traffic management, as a part of traffic control, represents a relevant segment of the studies carried out in ITS. The Panel on Future Directions in Control, Dynamics, and Systems identified in [2] several challenges for control systems with a straightforward application in ITS: the notion of operating in a distributed system and the need for coordination and autonomy. Currently, ITS are distributed by nature; hence an effort is needed to visualize the system as such [3]. A distributed system is usually defined as a set of independent computing elements whose hardware and software components communicate and coordinate their actions only through passing messages, appearing to the final user as a whole system (transparency). Among others, design challenges in distributed systems include performance, robustness, and reliability, with emphasis on achieving these features in the context of communication within ITS. Regarding coordination and autonomy, Murray et al. explain in [2] that the study and development of robust control systems require more elaboration in order to realize higher level decision-making systems. The benefits of such decisions are enormous: to enhance the efficiency of the vehicle transportation system by improving safety, reducing pollution (particle and noise), decreasing fuel consumption, and increasing service quality (e.g. with shorter vehicles' travel and waiting times). For instance, according to the National Highway Traffic Safety Administration data [4], incorrect driver decisions account for 33% of accidents, being driver recognition the top one reason (41%). These figures are expected to be significantly reduced with proper processing of collected data, and an adequate system-intelligence applied.

For intelligent and autonomous traffic light control at intersections, researchers have explored a wide range of approaches. The general methods that have been used to create smart, autonomous control systems for signaled intersections have been diverse, such as fuzzy logic [5–7], reservation and market-based system [8–10], neural networks [11–13], reinforcement learning [14–18], and swarm intelligence and evolutionary computation [19–21].

The main problem with most of these algorithms is that they are complex to run in real-time on IoT devices, demanding the resources of cloud computing [22]. But with the rise of Smart Cities, autonomous cars, 5G, and ITS, using the advantages offered by IoT (integration, embedded, speed, simplicity, etc.) can be a definite advantage for the development of intelligent traffic light control systems. In this paper, we face the problem of combining IoT with intelligent algorithms. Random Early Detection for Vehicles Dynamic (REDVD) is a very simple and efficient mechanism that optimizes the phases and cycles of traffic lights in signaled intersections [23]. However, REDVD outperforms other mechanisms in a stand-alone intersection, it does not behave as expected in more complex scenarios. The reason lies in the number of configuration parameters that influence its performance. Therefore, we propose to use an evolutionary algorithm (EA) to obtain the best parameter configuration of REDVD. Then, the performance of this new enhanced version, called improved REDVD (iREDVD), is tested under complex and previously-unknown traffic scenarios. The Key Performance Indicators (KPI) used in this study are: waiting time, trip time, travel speed, emissions (specifically, CO, CO2, HC, PMx, and NOx), and fuel consumption. In the light of our outcomes, iREDVD is a light-by-design method that can be deployed in IoT devices and outperforms not only REDVD but also other well-known traffic management methods in all the KPI under study.

The rest of the paper is organised as follows. In Section 2, we briefly summarise the state of the art. Section 3 describes our proposal. In Section 4, the optimization process is detailed. The methodology followed for the performance evaluation is explained in Section 5. In Section 6, we show and discuss the obtained results. The paper ends summarising the most important outcomes.

2 | RELATED WORKS

Within evolutionary computation, there are two main branches. First, we have those that use genetic algorithms (GAs) to represent the phases and timing of traffic lights as a set of chromosomes and perform optimisation directly on these chromosomes (could also be included within the fuzzy logic). Second, we have those that use GA as an optimisation algorithm to optimise a traffic light control system with a large number of parameters, which is unfeasible to optimise otherwise. The main advantage of these algorithms is that they are light to run but require a large computational load for training or for adjusting their parameters. However, this disadvantage can be mitigated by working in the cloud during a training phase and updating the algorithm easily once finished.

Within the first group, it stands out the work carried out by Sánchez et al. [24], where GA is used to encode a fuzzy logic controller in the chromosomes and find the optimal parameters according to the number of cars waiting. Also, similar works on optimising traffic light timings using GA are presented in [21], [25]. An approach linking GAs with device communication (D2D) can be found in [26]. This D2D approach is used to collect information from sensors and actuators, as well as to try to reduce the response time.

Concerning the second group, it is within this one that our work is encompassed, since we use a GA to optimise the traffic light control algorithm. The advantage of this method is that it allows us to shift the complexity of the search for the parameters of the control algorithm to a training scenario. This allows us to adapt the control algorithm to numerous scenarios and changes in the environment and act in advance to planned changes to be able to study the scenario beforehand. Other works use optimisation algorithms similar to GAs, such as ant colony optimization (ACO) algorithms. An ACO algorithm is a probabilistic technique for solving computational problems that can be reduced to find good paths through graphs. Within this category, the work done by Abdul Rehman *et al.* [27] and Kponyo Jerry et al. [28] stand out, where both solve the traffic congestion problem applying ACO.

Differing from previous works in the existing literature, our work focuses on optimising an adaptive traffic light control algorithm known as REDVD by means of a GA. Our improved version of this algorithm is called iREDVD. The novelty lies in the use of GAs for the optimisation process, facilitating the task of setting the best values for the numerous configuration parameters that the algorithm has. To the best of our knowledge, whereas most works from the related literature include GAs in traffic management procedures themselves, we only use it as an offline step. Then, we demonstrate that once the parameters are set, iREDVD works successfully under unknown traffic scenarios keeping its lightness feature and without the burden of AI. Therefore, it is possible to use it in IoT devices.

3 | RANDOM EARLY DETECTION FOR VEHICLES

As seen in the previous section, there are several approaches to traffic light controlled intersection control algorithms. In this section, we will show the basis of the ITS proposed in [23], namely REDVD, which serves as the basis for this work.

The operating principle of REDVD is the Random Early Detection (RED) [29] algorithm, used for congestion control in communications networks. RED is a congestion control technique extensively used in packet-switching communication networks. It is used to detect and avoid congestion and delay at an early stage and to improve the throughput rate. RED avoids congestion by randomly discarding packets, depending on the number of packets waiting to be processed in the router. These packets can be dropped if the packet queue to be processed exceeds a certain threshold; the dropping probability increases as the queue size increases. By doing so, congestion can be avoided even before it occurs. For more information about RED, please refer to [29].

Note that the concept of a queue at a router and a queue at a traffic light in a signalled intersection are analogous. As proposed in [23], the analogy with intersections occurs when, instead of discarding packets, we increase the green-time when many vehicles are queuing to cross an intersection. With this, we can get more vehicles crossing the intersection. The first adaptation of RED for traffic control in urban scenarios was introduced in [23] and was called RED for Vehicles (REDV). REDV adapts the RED algorithm to the particularities of vehicles and signalled intersections. The idea was as follows. In REDV, the number of vehicles waiting to cross at each intersection branch is similar to the number of packets queued for processing in a router. Instead of dropping a packet with a certain probability that depends on the number of packets to be processed, what REDV does is to increase or decrease the green time of each intersection branch with a probability that depends on the number of cars waiting to cross. Thus, when there are many cars on a branch, the green time for that branch increases frequently. Otherwise, when there are too few cars waiting to cross, the green time is likely reduced. In any case, the total cycle of the intersection was a fixed value. REDV demonstrated significant performance as an adaptive control system, managing incipient congestion at isolated intersections proactively.

To deal with changes in the traffic flow rate i.e. in the number of vehicles crossing the intersection per unit of time, it was necessary to adapt the total cycle. The idea was to provide long cycles to accommodate a large volume of traffic and short cycles when the volume of traffic was small. Consequently, the same authors introduced in [23] a new version called REDV Dynamic (REDVD). Although the performance in terms of traffic metrics was improved, this adaptation added new adjustable parameters to the configuration, parameters that need to be appropriately tuned for the algorithm's optimal performance. Table 1 includes all the configuration parameters of REDVD.

These new parameters contribute to higher flexibility in traffic control, being adaptive to continuously changing traffic conditions. However, the process of optimising this large set of parameters is not trivial. Indeed, the values used in [23] for configuring REDVD were obtained empirically. For this reason, one of the goals of this work is to find out the optimal configuration able to offer the best performance in terms of the average waiting time per intersection for each vehicle. This optimisation will be carried out using an EA, more specifically, employing a GA, which is the most popular type of EA. This version optimised by means of a GA is called iREDVD. The entire optimisation process is described in the following section.

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 because of a probabilistic increase when it is 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	Factor that determines the weight of the historical value versus the current value in the queue length.
maxp	Factor used in RED algorithm.

4 | OPTIMISATION PROCESS

A GA consists of a series of mechanisms inspired by the theory of evolution to optimise a set of parameters within a problem. It allows us to optimise that problem, a priori complicated, using a set of light procedures with a low computation load, which derives in a fast parameter optimisation [30], [31].

The general procedure consists of four phases: initialize population, fitness calculation, selection, and crossover, which form a generation. The overall optimisation process will take as many generations as needed. A population is a set of individuals. The number of genes that each individual contains is determined by the number of parameters to be optimized. In our case, the parameters are included in Table 1. And each gene contains the coded information of each parameter; e.g. in our case, the *maxp* parameter will be a float number between 0 and 1. As an example, in a problem with six binary parameters to optimise (such as the knapsack problem [32]), an example of population would be the one shown in Figure 1. In this figure, it can be seen a population of four individuals (I_1 to I_4), each one with six parameters to optimise (P_1 to P_6), and each of these parameters is coded in binary (0 or 1) in each gene.

For each generation, the best-adapted individuals to the environment are selected so that the next generation individuals contain their genes. The meaning of "better adapted to the

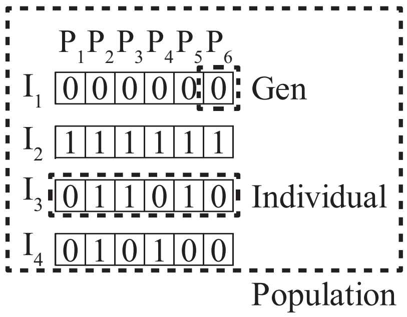


FIGURE 1 Example of a population of a GA, which consists of four individuals and six binary parameters to be optimised

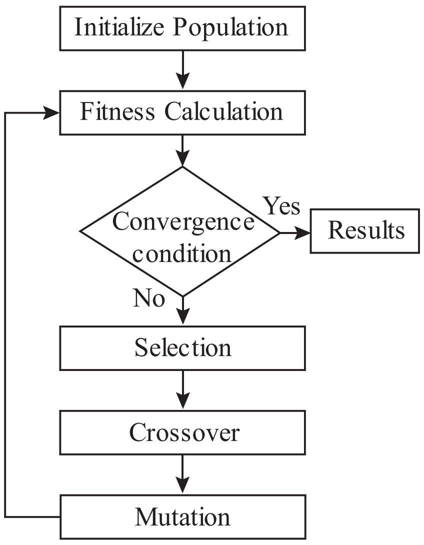


FIGURE 2 A general GA procedure. The convergence condition can be different, either several entire generations or a fitness value improvement after a few generations

environment" is given by the fitness function, which numerically measures how well an individual can adjust to the problem being optimised. For example, in tasks of humanoid robots learning to walk where genes encode the movement of joints and muscles, the fitness function will be the distance travelled, and those individuals who have gone the furthest will be most likely to be selected. Next, we describe the four phases of the general optimisation procedure in detail (see Figure 2).

4.1 | Initialize population

The procedure starts by initialising the population of individuals with random genes (although other approaches can be followed in order to reduce the convergence time [33], [34]). The population's size is a crucial factor for the correct convergence of the GA because it causes more diversity in the population and ensures that a large amount of the parameter space is being explored.

Nevertheless, there is an important trade-off between the population's size, genetic diversity, and convergence speed. If the population size is considerable, then we have great genetic diversity by exploring a wide range of the parameter space at random. Still, it will take a long time to get the fitness of all the individuals in the population. The size of the population depends strongly on the problem studied as to cover a considerable range of the parameters, it is necessary to increase the population's size exponentially. A basic (minimum) recommendation is to have twice more individuals than the parameters studied. Typical population sizes can be between 20 and 100 individuals [30], [35], [36].

4.2 | Fitness calculation

Each individual in the population is evaluated, and its level of fitness is obtained with its parameter settings. For instance, if we are designing a robot that is able to walk, the fitness of each individual would be how far it can walk [37], or in the knapsack problem [32] the adjustment would be the benefit obtained by the objects inside the knapsack that minimize the weight. In our case, it will be the average waiting time per intersection for each vehicle.

4.3 | Selection

Once all the individuals in the population have been evaluated, the next step is to select the subset of individuals that are best-adapted to the environment. There are various mechanisms for selecting the best individuals [38–40] e.g. roulette wheel selection, Boltzmann selection, tournament selection, rank selection, steady-state selection, or Elitism selection, among others. One of the simplest and most effective methods is to select the percentage of individuals who have the highest level of fitness. With this mechanism, the subset of best-adapted individuals to the environment will be obtained in the current generation. These selected individuals are known as parents and will be the ones from whom the children are generated.

The percentage of individuals selected is crucial for the correct functioning of the GA. A very high value will prevent rapid convergence, but a very low value will nevertheless lead to low genetic diversity and a high possibility of converging towards a local minimum. Typical values are between 5% and 10%, depending on the population size [30], [35], [36]. The remaining individuals (children) will be generated from these parents to obtain the total population employing crossover and mutation mechanisms.

Although there are very sophisticated methods for the selection mechanism, it is recommended [41] to use rank selection in the first generations to quickly obtain preliminary results. More sophisticated techniques can be found in [40].

4.4 | Crossover

Crossover is a mechanism by which new individuals (children) are generated from two or more parents' direct mating. This mechanism is done to obtain children with the genes of those parents with a good fitness value. The basic operation of this mechanism consists of randomly selecting genes from different parents to generate new children. There are various crossover mechanisms, such as the selection of a single crossover point, the selection of multiple crossover points, or the gene-by-gene combination [42–45]. The mechanism used in this work is the single crossover point due to its simplicity and good results [44]. In this mechanism, a crossing point is selected. Then, from the beginning of the genes to the crossing point is transferred from one parent and the rest is transferred from the second parent.

The crossover rate indicates how often parents are crossed over to get new children, so it controls what percentage of the children are exact copies of the parents and which ones are crossed over. That is, if the crossover rate is 100%, all the children are obtained by crossing the parents. However, if the crossover rate is 0%, all the children are exact copies of the parents. This 0% does not mean that the new generation will be identical to the parents since, after the crossover, the genes of these created children can be mutated. The crossover rate should generally be high, between 80% and 95% [30], [35], [36].

4.5 | Mutation

In order to obtain a high genetic diversity, mutation mechanisms can be applied to a small percentage of genes in the population. This mutation means that the value of each gene can vary randomly with a small probability. The mutation prevents the GA from converging to a local optimum, as it allows other combinations of genes that can achieve better fitness levels to be explored at random [46], [47]. However, this mutation should not occur very often, since then it would become a random search.

The mutation rate indicates how often an individual's gene will be mutated. If the mutation rate is 0%, the population created after the crossover does not mutate. However, if the mutation rate is 100%, all the offspring's genes are changed randomly, within a range of change, which depends on the scenario. This margin of change should be less than 20% of the allowed range for each gene [47]. In turn, the mutation rate should remain around 0.1% and 1%, to enable the exploration of a wider space of parameters, obtaining genetic diversity, but without falling into exploring the entire parameter space at random [48], [49]. In addition, it is advisable to decrease the mutation rate, as well as the margin of change of the parameters, as the generations go by, in order to obtain a rapid convergence [30], [35], [36]. Figure 2 shows the general procedure followed by the GA until the convergence condition is satisfied.

For iREDVD, we choose a population size of 256 individuals, 16 selected parents, a crossover rate of 80%, a mutation rate of 0.1%, and up to 20 generations.

5 | MATERIALS, SCENARIOS, AND METHODS

The algorithms under study have been evaluated via computer simulations. All methods were programmed in Python v3.7. To

TABLE 2 Characteristics of the training and the testing scenarios

Scenario	Number of intersections	Intersection layout	Distance between intersections	Simulation duration
Train	16	4×4	300m	10h
Test1	5	1×5	200m	12h
Test2	100	10×10	250m	12h

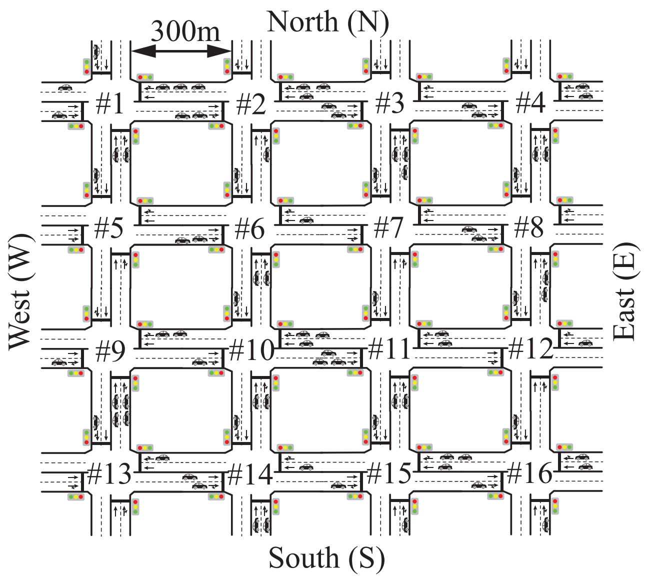


FIGURE 3 Simulated topology for the training scenario: Manhattan 4×4 network with 300 m between each intersection

compare the different proposals, these scripts were coupled to the microscopic traffic simulator SUMO [50] v1.0.1 with TraCI (Traffic Control Interface). SUMO is a valuable tool as it allows flexible simulations in customizable scenarios and editable traffic patterns. The CPU used was an Intel Xeon 16 cores @ 2.6GHz. Three scenarios were used in this work, one for training and two for testing. The details of each scenario can be seen in Table 2. These three scenarios share some characteristics and differ in others, which are described in the next subsections.

On the one hand, the training scenario consists of a grid of 4×4 intersections (in total 16 intersections) with 2 lanes for each direction, and a distance between intersections of 300m (see Figure 3). The vehicle flow rate fluctuated over time, remaining constant for one hour and following the pattern shown in Figure 4 afterward. The low flow rate was 600 vehicles/hour/branch, the medium flow rate 1200 vehicles/hour/branch, and the high flow rate 1600 vehicles/hour/branch. The north (N) and south (S) branches had identical flow rates, and the same applies to the east (E) and west (W) branches. As shown in Figure 4, there are times when the flow rate is symmetric (e.g. hours 0, 1, and 2) and others when is asymmetric (e.g. hours 3, 4, and 5).

On the other hand, the testing scenarios are used to evaluate the performance in new conditions i.e. situations the algorithm has not been trained for. Simulations run in the testing scenarios

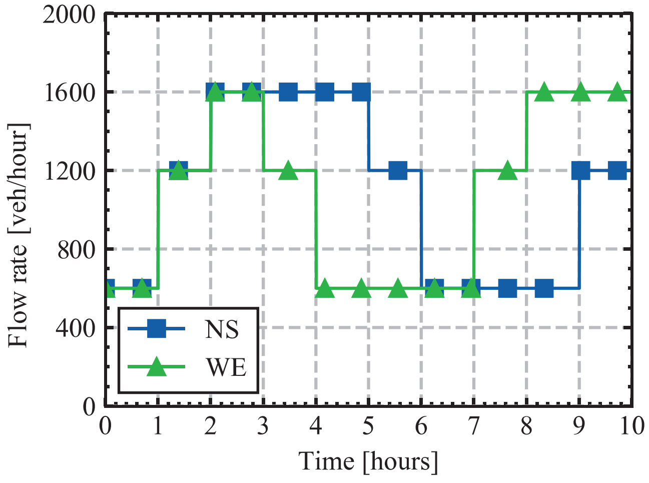


FIGURE 4 Vehicle flow rate per branch used in the training scenario

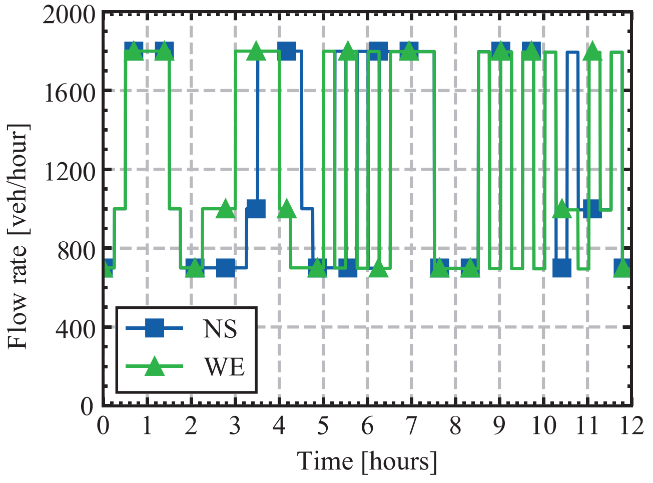


FIGURE 6 Vehicle flow rate per branch used in the testing scenarios

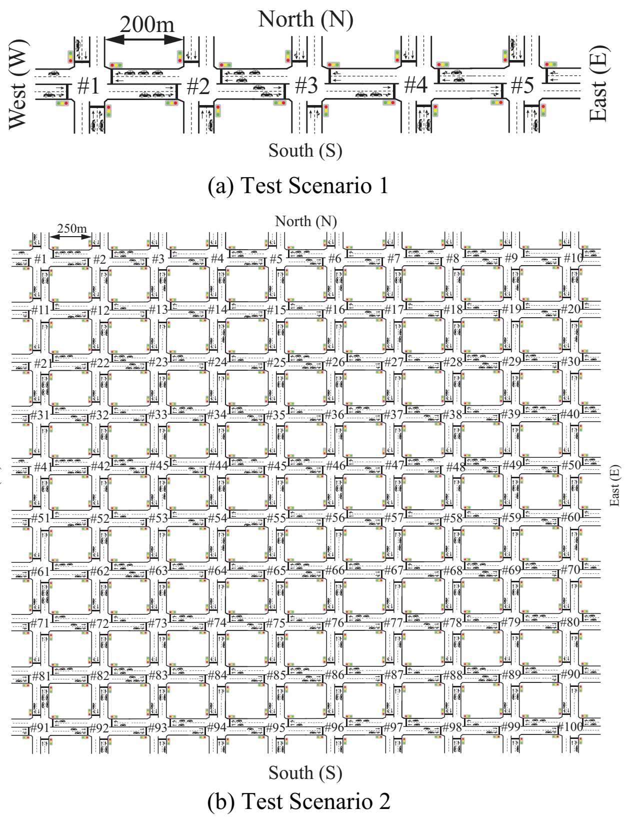


FIGURE 5 Simulated topology for the testing scenarios: (a) 1×5 network with 200m between each intersection; (b) Manhattan 10×10 network with 250m between each intersection

use the optimal parameters obtained previously in the training. The test scenario 1 consists of a large avenue of 5 intersections (1×5) separated by 200 m between each intersection, as shown in Figure 5(a). The test scenario 2 consists on a 10×10 Manhattan grid, with a total of 100 intersections, as can be seen in Figure 5(b).

Both test scenarios have the same vehicle flow rate, depicted in Figure 6. The vehicle flow rate is more chaotic than in the training scenario, hence, stressing the different algorithms under study. As depicted in Figure 6, the constant vehicle flow rate stretches are maintained for 15 min, and the low, medium, and high flow rates are 700, 1000, and 1800 vehicles/hour/branch, respectively. With these test scenarios, we check that iREDVD algorithm is not overfitted to the training scenario.

TABLE 3 Type of vehicles

Vehicle	Percentage	Fuel
Car	30%	Gasoline
Car	40%	Diesel
Motorcycle	10%	Gasoline
Moped	10%	Gasoline
Van	5%	Diesel
Bus	5%	Average of all fuel types

In all scenarios, the selected vehicle distribution was based on the fleet of vehicles of the city of Madrid (Spain) [51]. This distribution can be seen in Table 3. Also, in order to obtain both fuel consumption and pollutant emissions from vehicles, the pollution model used was the HBEFA [52]. HBEFA is an emission factor database for road vehicles and provides emission factors (hot start, cold start, evaporation) for all regulated and important unregulated air pollutants as well as for fuel consumption.

Similarly, each intersection had four traffic lights that control each branch of the intersection (North, South, East, West) in all scenarios. Each intersection had two lanes on each branch. An example of an intersection is depicted in Figure 7. For simplicity, only the straight forward and the right-turn movements were allowed (see Figure 7). This constrain can be found in numerous articles [53–57]. Mainly, it allows a quickly/simple implementation of a new concept to study the advantages and disadvantages it can offer. When adding the left turn, only one new phase needs to be incorporated in the whole cycle, thus allowing the vehicles to turn left. Therefore, we consider that the selected configuration is appropriate to show the benefits of the

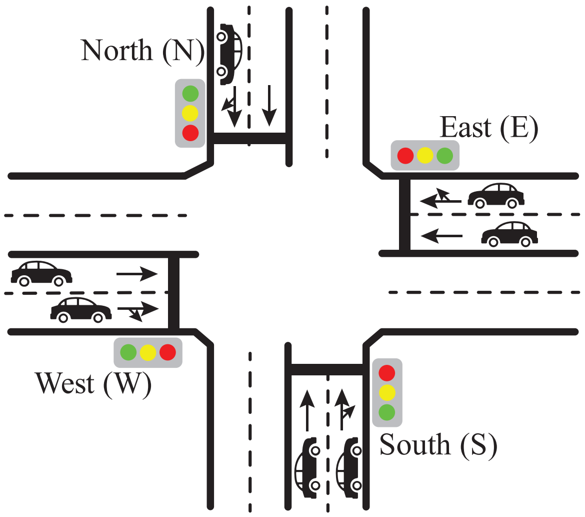


FIGURE 7 An example of a four-way signaled intersection, as used in this study

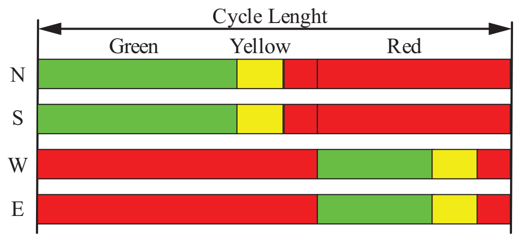


FIGURE 8 Example of a cycle in a traffic light. N≡North; S≡South; W≡West; E≡East

optimization process using the GA and the robust performance of iREDVD. The phases of the traffic lights were coupled (North with South and East with West) and were opposite (North and South vs. East with West) between each couple. In other words, if the traffic lights in the North and South branches were green, it means that East and West branches were red. Figure 8 illustrates an example of a cycle. The yellow-time interval of the traffic lights was set to 3 seconds and the all-red time to 2 seconds, for safety and realism. The safety distance used between cars was 1.5 m. A Poisson distribution was used to generate the traffic flows.

In order to compare the performance of iREDVD with other algorithms for traffic control, we also implement and evaluate the performance of a system with fixed cycle (FX) time and with pre-set offsets between consecutive intersections, known as Green Wave (GW). This offset is obtained as the time that a vehicle needs to move between two intersections once the green phase starts (i.e. in a coordinated manner like a green wave). Three different versions were studied for the FX and GW algorithms, which also modify the total cycle time. The cycle times of 30, 45, and 60 seconds were studied. The nomenclature used for each algorithm is expressed as the algorithm used and the cycle time; so, for example, FX45 corresponds to the fixed algorithm with a cycle time of 45 seconds, equally divided between the

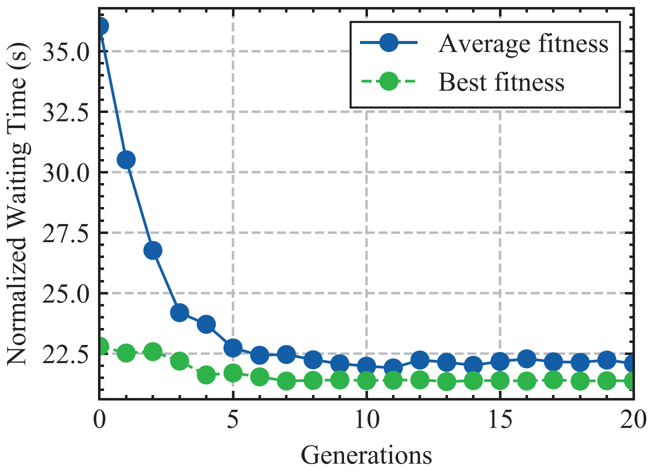


FIGURE 9 Optimization process. "Best fitness" is the individual with the lowest normalised waiting time and "Average fitness" is the average normalised waiting time of all population. X-Axis represents the generations

branches. In all cases, 10 experiments were run for each algorithm under study in each scenario, obtaining the mean value and its standard deviation.

Finally, the optimised metric was the normalised waiting time (the waiting time per intersection), although trip time, average speed, CO, CO2, HC, PMx and NOx emissions, and fuel consumption were also studied. The normalised waiting time eliminates the influence of the number of intersections crossed by vehicles in each flow. Therefore, we obtain for each vehicle the ratio between the total waiting time and the number of intersections crossed during the travel.

6 | PERFORMANCE EVALUATION RESULTS

6.1 | Training scenario

Figure 9 illustrates the value of the normalised waiting time of the vehicles per intersection during the optimisation process carried out by the GA in the training phase. As it is shown, after about five generations, the GA achieves promising values for most of the simulated individuals. Although the simulation could have been stopped at that time, we left it more to find a robust set of parameters, which over time are able to demonstrate their superiority over the rest. The time required for each of the simulations in the optimisation process for the test scenario was approximately 1 min. This means that simulating the 256 individuals for 21 generations (the initial one plus the next 20 generations) on a server with 16 processing cores (i.e. 16 simulations were simulated in parallel) required approximately 48 hours. Each simulation was repeated 10 times, and the mean value and standard deviation of each variable under study were obtained. Once the best parameter configuration has been obtained, the execution of the traffic light control algorithm is almost instantaneous since it no longer requires any artificial intelligence algorithm since it is based on a light, fast, and quick

TABLE 4 Optimised parameters and values

Parameter	Optimal value
minth	10
maxth	25
min_greentime	15
max_greentime	60
min_cycle	45
max_cycle	130
delta	10
delta_cycle	15
maxp	0.85
wq	0.7
limin	5
limdec	1

method that can be rapidly executed by low power IoT devices. Further information can be found in [23].

The values of the optimised parameters after training are shown in Table 4. The optimisation process shows very interesting values. The parameter *limdec* indicates how many times the green time of the traffic lights is reduced so that the total cycle of the intersection is reduced. The fact that *limdec* is 1, means that if a reduction in traffic is detected, even a slight one, it is worthwhile to shorten the total cycle time (and therefore the green time) quickly. However, the opposite is true for the increase in total cycle time. The parameter *liminc* controls the increase in the total cycle time of the traffic lights when traffic increases. With a high value of 5, it means that iREDVD should be very sure before increasing the total cycle time.

The results obtained in the training scenario (see Table 5) corroborate that iREDVD has an improvement of more than 50% (reduction in almost 25 seconds) in the average waiting time of the vehicles per intersection when compared to the traditional algorithms (Fixed 30, Fixed 45, Fixed 60, GreenWave 30, GreenWave 45, and GreenWave 60).

Compared with the original REDV algorithm proposed in [23], our improvement is still visible. This reinforces the fact that adjusting the parameters is necessary for the proper functioning of the algorithm. Finally, compared to the original REDVD variant also presented in [23], there is an improvement of 27%, which means a reduction of more than 8 seconds in the average normalised waiting time. Regarding other key performance metrics, such as average travel time, average speed, average CO, CO2, HC, PMx and NOx emissions, and average fuel consumption, we can see that the iREDVD improves in all of these KPI in the range of 7%–45%; most notably an increase in speed of 16%–26%, a decrease in average CO emissions of 13%–46%, and a decrease in average fuel consumption of 7%–18%, compared to traditional algorithms.

6.2 | Testing scenarios

After the training, we stressed the algorithm in the testing scenarios using the optimised values for the configuration parameters. It can be seen from the results shown in Tables 6 and 7 that improvements are very similar to those obtained in the training scenario. This corroborates that iREDVD is able to adjust to unknown conditions (scenarios that it was not trained for), showing its robustness to be applied in real deployments.

For example, iREDVD achieves a reduction in average normalised waiting time of 34%–49% in Test 1 and 78%–82% in Test 2, compared to other control techniques, which is equivalent to reducing the average waiting time of vehicles at each intersection crossed by more than 6–100 seconds. This improvement is also reflected in other metrics, such as the reduction of average CO emissions by around 25% or a reduction in average fuel between 6%–17%.

6.3 | Final remarks

The excellent performance of the optimised iREDVD is due to the ability to adapt to road conditions. In Figure 10(a) and Figure 10(b), it can be observed how iREDVD proactively adapts the cycle time according to the simulated traffic flow rate, not only in training Figure 10(a), but also under unknown conditions in the testing Figure 10(b). The cycle increases when the flow rate of vehicles passing through the intersection increases, and vice versa. In addition, we can see in Figure 10(a) and Figure 10(b) that the time assigned to each branch is independent, since when the traffic is asymmetric [see Figure 10(b) hour 4] iREDVD obtains a time for each branch that is different, self-adjusting to the traffic conditions. This independence allows iREDVD a better time distribution and a reduction of waiting time.

7 | CONCLUSION

With the growing number of vehicles, the massification of large cities, and the rise of the IoT, there is a growing need for orchestration for smart cities. The use of ITS allows for efficient traffic control when applied to traffic-light managed intersections, improving traffic flow, and diminishing pollution and fuel use. In this work, we have presented the optimisation and performance of the iREDVD ITS method using an EA as the optimisation tool. iREDVD is a lightweight, optimised, and fast algorithm capable of being executed in IoT devices with limited requirements, whereas controls signaled intersections in a very efficient way. We have shown that iREDVD outperforms traditional traffic light control techniques, as well as its different previous versions. Compared to traditional control techniques, it showed reductions between 24 to 100 seconds in the waiting time of vehicles per intersection travelled through, equivalent to between 50% and 80% reduction in the waiting time of vehicles at traffic lights. Compared to the original REDVD, iREDVD reduces the waiting time at each intersection by more than 27%. Besides, iREDVD reduces both the pollutant emissions and fuel consumption between 7%–38% and between 7%–14%, respectively.

In conclusion, the use of these ITS will allow future smart cities to orchestrate the different actors better, allowing for

TABLE 5 Train results

Algorithm		Avg. norm. waiting time (s)	Avg. trip time (s)	Avg. speed (m/s)	Avg. CO emissions (mg)	Avg. CO2 emissions (g)	Avg. HC emissions (mg)	Avg. PMx emissions (mg)	Avg. NOx emissions (mg)	Avg. Fuel consumption (ml)
Traditional	*FX30*	51,5 ± 0,11	213,96 ± 0,23	7,25 ± 0,01	6642,68 ± 21,51	783,90 ± 4,49	131,83 ± 1,08	62,75 ± 0,56	3988,72 ± 31,07	315,59 ± 1,79
	FX45	48,3 ± 0,04	201,81 ± 0,09	7,61 ± 0.02	6184,13 ± 24,62	739,80 ± 3,07	122,93 ± 0,22	58,39 ± 0,14	3749,9 ± 15,14	297,69 ± 1,11
	FX60	51,64 ± 0,06	204,76 ± 0,10	7,53 ± 0.01	6463,22 ± 50,19	739,23 ± 2,78	124,46 ± 1,42	58,87 ± 0,69	3754,17 ± 26,77	297,47 ± 1,19
	GW30	46,13 ± 0,08	203,29 ± 0,11	7,65 ± 0.04	6016,83 ± 36,41	751,81 ± 2,05	122,43 ± 0,38	58,54 ± 0,16	3788,61 ± 14,89	302,59 ± 0,71
	GW45	50,02 ± 0,04	206,29 ± 0,07	7,64 ± 0.01	6427,96 ± 20.95	748,24 ± 2,40	125,86 ± 0,89	59,63 ± 0,43	3796,72 ± 19,92	301,2 ± 0,99
	GW60	47,4 ± 0,07	199,45 ± 0,07	7,82 ± 0.02	6178,43 ± 18,74	725,66 ± 5,34	120,27 ± 1,06	56,98 ± 0,52	3665,26 ± 34,72	291,95 ± 2,11
REDV original		62,07 ± 4,01	234,22 ± 7,61	7,22 ± 0,09	7560,28 ± 363,68	819,66 ± 15,15	143,19 ± 5,02	67,75 ± 2,21	4221,82 ± 96,23	4221,82 ± 96,23
REDVD original		29,58 ± 0,33	184,06 ± 0,67	8,27 ± 0,03	4708,8 ± 27,36	723,41 ± 1,67	107,39 ± 1,00	52,51 ± 0,51	3573,78 ± 16,83	3573,78 ± 16,83
iREDVD		**21,37 ± 1,86**	**167,2 ± 2,91**	**9,14 ± 0,16**	**4074,01 ± 135,67**	**672,11 ± 10,13**	**96,01 ± 2,35**	**46,96 ± 1,15**	**3275,5 ± 59,96**	**3275,5 ± 59,96**
*Improvement of **iREDVD** vs traditional.**										
Abs		-24,76	-32,25	1,32	-1942,82	-53,54	-24,27	-10,02	-389,76	-21,83
%		53,67	16,16	16,87	32,28	7,37	20,18	17,58	10,63	7,47
*Improvement of **iREDVD** vs REDV orig.*										
Abs		-40,7	-67,02	1,92	-3486,27	-147,54	-47,19	-20,79	-946,32	-59,92
%		65,57	28,61	26,59	46,11	18,01	32,96	30,68	22,42	18,15
*Improvement of **iREDVD** vs REDVD orig.*										
Abs		-8,21	-16,86	0,87	-634,79	-51,29	-11,39	-5,55	-298,28	-20,85
Improvement %		27,75	9,16	10,51	13,48	7,09	10,60	10,56	8,34	7,16

*Improvement when compared to the best traditional algorithm.
Note: mean ± std of 10 test

TABLE 6 Test 1 results

Algorithm		**Avg. norm. waiting time (s)**	**Avg. trip time (s)**	**Avg. speed (m/s)**	**Avg. CO emissions (mg)**	**Avg. CO2 emissions (g)**	**Avg. HC emissions (mg)**	**Avg. PMx emissions (mg)**	**Avg. NOx emissions (mg)**	**Avg. Fuel consumption (ml)**
Traditional	*FX30*	21,22 ± 0,25	169,9 ± 1,21	7,34 ± 0,02	5395,04 ± 65,59	597,56 ± 4,11	102,91 ± 0,73	48,99 ± 0,32	3073,18 ± 22,29	240,61 ± 1,6
	FX45	19,2 ± 0,03	150,47 ± 0,16	7,72 ± 0,01	4756,42 ± 23,49	551,99 ± 1,54	92,4 ± 0,27	44,06 ± 0,15	2814,95 ± 5,68	222,12 ± 0,58
	FX60	21,74 ± 0,01	153,05 ± 0,08	7,58 ± 0,02	5047,54 ± 20,13	555,22 ± 3,02	95,51 ± 0,8	45,26 ± 0,42	2844,12 ± 23,67	223,56 ± 1,17
	GW30	17,68 ± 0,39	154,05 ± 1,79	7,81 ± 0,04	4586,92 ± 69,72	558,36 ± 3,49	91,28 ± 0,91	43,83 ± 0,41	2829,06 ± 20,56	224,73 ± 1,4
	GW45	19,25 ± 0,03	151,67 ± 0,15	7,83 ± 0,05	4803,28 ± 19,67	554,43 ± 3,27	92,62 ± 0,74	44,21 ± 0,38	2825,08 ± 21,72	223,09 ± 1,28
	GW60	21,36 ± 0,02	151,66 ± 0,12	7,75 ± 0,03	4919,4 ± 18,56	555,01 ± 3,09	93,9 ± 1,19	44,67 ± 0,55	2839,66 ± 22,56	223,32 ± 1,19
REDV original		25,38 ± 0,71	152,55 ± 2,94	7,73 ± 0,05	4663,73 ± 120,77	547,49 ± 5,15	91,42 ± 1,91	43,61 ± 0,86	2789,18 ± 34,95	220,32 ± 2,14
REDVD original		17,66 ± 0,47	172,52 ± 2,99	7,49 ± 0,08	4771,9 ± 84,21	580,91 ± 7,70	95,48 ± 1,86	45,93 ± 0,91	2953,28 ± 51,28	233,81 ± 3,08
iREDVD		**11,6 ± 0,27**	**132,6 ± 1,65**	**8,38 ± 0,06**	**3406,27 ± 66,39**	**511,34 ± 3,07**	**77,09 ± 1,56**	**37,62 ± 0,7**	**2536,01 ± 23,01**	**205,74 ± 1,36**
Improvement of ***iREDVD*** *vs traditional.**										
Abs		-6,08	-17,87	0,55	-1180,65	-40,64	-14,19	-6,21	-278,94	-16,38
%		34,38	11,87	7,02	25,73	7,36	15,54	14,16	9,91	7,37
Improvement of ***iREDVD*** *vs REDV orig.*										
Abs		-13,78	-19,95	0,65	-1257,46	-36,14	-14,33	-5,99	-253,17	-14,58
%		54,29	13,07	8,40	26,96	6,60	15,67	13,73	9,07	6,61
Improvement of ***iREDVD*** *vs REDVD orig.*										
Abs		-6,06	-39,92	0,89	-1365,63	-69,57	-18,39	-8,31	-417,27	-28,07
Improvement %		34,31	23,13	11,88	28,61	11,97	19,26	18,09	14,13	12,0

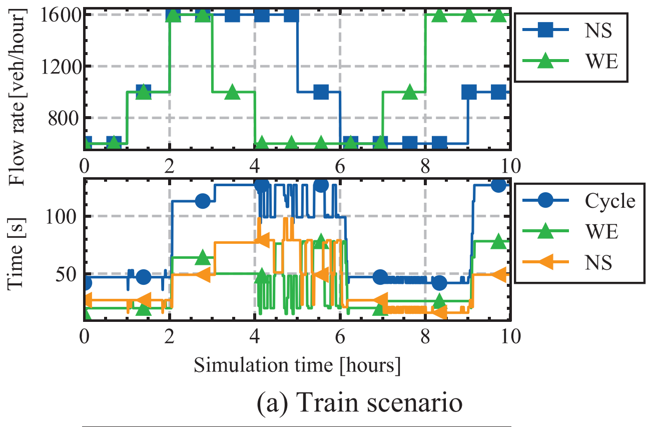
*Improvement compared to the best traditional algorithm.
Note: mean ± std of 10 test

TABLE 7 Test 2 results

Algorithm		Avg. norm. waiting time (s)	Avg. trip time (s)	Avg. speed (m/s)	Avg. CO emissions (mg)	Avg. CO2 emissions (g)	Avg. HC emissions (mg)	Avg. PMx emissions (mg)	Avg. NOx emissions (mg)	Avg. fuel consumption (ml)
Traditional	*FX30*	139,05 ± 0,04	428,63 ± 0,43	6,49 ± 0,01	14273,45 ± 55,48	1508,16 ± 10,14	266,46 ± 1,25	125,60 ± 0,33	7738,98 ± 7,48	608,09 ± 0,87
	FX45	131,08 ± 0,07	400,10 ± 0,85	6,85 ± 0,02	13064,96 ± 44,84	1415,19 ± 4,25	246,19 ± 0,84	116,05 ± 1,15	7274,22 ± 10,44	569,66 ± 2,15
	FX60	134,60 ± 0,21	399.23 ± 0,37	6,90 ± 0,05	13388,89 ± 14,35	1396,32 ± 7,19	246,41 ± 3,44	115,58 ± 0,88	7181,57 ± 8,21	561,98 ± 1,49
	GW30	122,31 ± 0,18	403,24 ± 0,59	6,69 ± 0,05	12634,05 ± 43,01	1443,93 ± 13,25	245,45 ± 2,25	116,67 ± 0,91	7738,98 ± 4,97	581,48 ± 0,45
	GW45	136,20 ± 1,01	407,89 ± 0,89	6,97 ± 0,02	13394,84 ± 24,12	1435,91 ± 6,84	252,15 ± 0,98	118,97 ± 1,01	7274,22 ± 11,02	577,83 ± 1,04
	GW60	122,79 ± 0,50	382,22 ± 1,21	7,28 ± 0,01	12415,55 ± 33,21	1348,19 ± 9,50	231,86 ± 1,21	109,01 ± 2,45	7185,77 ± 6,81	542,51 ± 0,76
REDV original		140,69 ± 2,15	421,36 ± 8,16	5,98 ± 0,11	15584,88 ± 33,86	1550,40 ± 3,46	308,69 ± 3,48	130,18 ± 2,38	7911,04 ± 15,01	624,14 ± 1,24
REDVD original		99,22 ± 8,33	374,45 ± 5,74	8,12 ± 0,18	8847,37 ± 76,84	1321,84 ± 2,62	212,11 ± 1,12	99,01 ± 1,22	6812,88 ± 21,66	539,69 ± 4,18
iREDVD		**21,45 ± 0,24**	**286,85 ± 0,15**	**9,39 ± 0,11**	**6764,93 ± 27,11**	**1286,49 ± 0,96**	**178,71 ± 1,52**	**89,27 ± 0,69**	**6249,30 ± 25,21**	**517,07 ± 0,59**
*Improvement of **iREDVD** vs traditional.**										
Abs		-101,34	-95,37	2,11	-5650,62	-61,70	-53,15	-19,74	-932,27	-25,44
%		82,46	24,95	28,98	45,51	4,58	22,92	18,11	12,98	4,69
*Improvement of **iREDVD** vs REDV orig.*										
Abs		-119,24	-134,53	3,41	-8829,95	-263,91	-128,98	-40,91	-1661,74	-107,07
%		84,75	31,93	57,02	56,62	17,02	42,11	31,43	21,01	17,15
*Improvement of **iREDVD** vs REDVD orig.*										
Abs		-77,77	-87,60	1,27	-2082,44	-35,35	-33,39	-9,74	-563,58	-22,62
Improvement %		78,38	23,39	16,64	23,54	2,67	15,74	9,83	8,27	4,19

*Improvement compared to the best traditional algorithm.

Note: mean ± std of 10 test



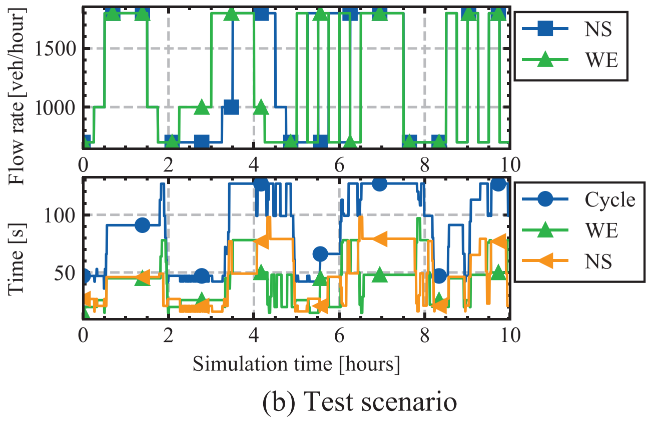


FIGURE 10 Vehicular flow rate simulated, time for each branch, and total cycle for intersection 1 vs. simulation time: (a) train scenario; (b) test scenario

increased safety, reduced pollution, and optimised transportation systems. As future work, research is being carried out on the integration of ITS with 5G technology, to obtain a fully connected traffic management ecosystem.

ACKNOWLEDGEMENTS

This work was supported by the AEI/FEDER, UE project grant TEC2016-76465-C2-1-R (AIM), by DGT-Ministerio del Interior, project grant SPIP2017-02230 (STREET), and by a pre-doctoral contract for the formation of the research personnel financed by Consejería de Empleo, Universidades, Empresa y Medio Ambiente of the CARM, through the Fundación Séneca - Agencia de Ciencia y Tecnología of the Region of Murcia, Spain.

ORCID

Antonio Guillen-Perez https://orcid.org/0000-0003-3067-6469
Maria-Dolores Cano https://orcid.org/0000-0003-4952-0325

REFERENCES

1. Zhang, J., et al.: Data-driven intelligent transportation systems : A survey. IEEE Trans. Intell. Transp. Syst. 12 (4), 1624–1639 (2011)
2. Murray, B.R.M., et al.: The Panel on Future Directions in Control, Dynamics, and Systems. IEEE Control Syst. Mag. 23 (2), 20–33 (2003)
3. Cano, M.-.D., et al.: Coordination and agreement among traffic signal controllers in urban areas. In: International Conference on Transparent Optical Networks, 2016.
4. Singh, S.: Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey. Natl. Highw. Traffic Saf. Adm.1–2 (2015). Washington, DC. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115
5. Pappis, C.P., Mamdani, E.H.: A fuzzy logic controller for a traffic junction. IEEE Trans. Syst. Man Cybern. 143(1–4), 73–97 (1977)
6. Chiu, S., Chand, S.: Adaptive traffic signal control using fuzzy logic. In: 1st IEEE Regional Conference on Aerospace Control Systems, AEROCS 1993 - Proceedings, 1993.
7. Niittymäki, J., Pursula, M.: Signal control using fuzzy logic. Fuzzy Sets Syst., 16(1), 11–22 (2000)
8. Dresner, K., Stone, P.: Multiagent traffic management: A reservation-based intersection control mechanism. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004.
9. Vasirani, M., Ossowski, S.: A market-inspired approach to reservation-based urban road traffic management. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, 2009.
10. Li, Z., et al.: Modeling reservation-based autonomous intersection control in VISSIM. Transp. Res. Rec. 2381 (1), 81–90 (2013)
11. Wei, C.H.: Analysis of artificial neural network models for freeway ramp metering control. Artif. Intell. Eng. 15(3), 241–252 (2001)
12. Srinivasan, D., Choy, M.C., Cheu, R.L.: Neural networks for real-time traffic signal control. IEEE Trans. Intell. Transp. Syst. 7(3), 261–272 (2006)
13. Tubaishat, M., Shang, Y., Shi, H.: Adaptive traffic light control with wireless sensor networks. In: 4th Annual IEEE Consumer Communications and Networking Conference, CCNC 2007.
14. Mousavi, S.S., Schukat, M., Howley, E.: Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intell. Transp. Syst. 11(7), 417–423 (2017)
15. Genders, W., Razavi, S.: Evaluating reinforcement learning state representations for adaptive traffic signal control. Procedia Comput. Sci. 130, 26–33 (2018)
16. Liang, X., et al.: A deep reinforcement learning network for traffic light cycle control. IEEE Trans. Veh. Technol., 68 (2), 1243–1253 (2019)
17. Balaji, P.G., German, X., Srinivasan, D.: Urban traffic signal control using reinforcement learning agents. IET Intell. Transp. Syst. 4(3), 177–188 (2010)
18. Arel, I., et al.: Reinforcement learning-based multi-agent system for network traffic signal control. IET Intell. Transp. Syst. 4(2), 128–135 (2010)
19. Sánchez, J., Galán, M., Rubio, E.: Applying a traffic lights evolutionary optimization technique to a real case: 'Las Ramblas' area in Santa Cruz de Tenerife. IEEE Trans. Evol. Comput. 12(1), 25–40 (2008)
20. Garcia-Nieto, J., Olivera, A.C., Alba, E.: Optimal cycle program of traffic lights with particle swarm optimization. IEEE Trans. Evol. Comput. 17(6), 823–839 (2013)
21. Segredo, E., et al.: Optimising Real-World Traffic Cycle Programs by Using Evolutionary Computation. IEEE Access, 7, 43915–43932 (2019)
22. McKenney, D., White, T.: Distributed and adaptive traffic signal control within a realistic traffic simulation. Eng. Appl. Artif. Intell. 26 (1), 574–583 (2013)
23. Sanchez-Iborra, R., Cano, M.-.D.: On the similarities between urban traffic management and communication networks: Application of the random early detection algorithm for self-regulating intersections. IEEE Intell. Transp. Syst. Mag. 9 (4), 48-61 (2017)
24. Sánchez-Medina, J.J., Galán-Moreno, M.J., Rubio-Royo, E.: Traffic signal optimization in La Almozara District in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. IEEE Trans. Intell. Transp. Syst. 11(1), 132–141 (2010)
25. Teo, K.T.K., Kow, W.Y., Chin, Y.K., Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. In: Proceedings - 2nd International Conference on Computational Intelligence, Modelling and Simulation, CIMSim 2010.

26. Tang, C., et al.: Traffic signal phase scheduling based on device-to-device communication. IEEE Access 6, 47636–47645 (2018)
27. Rehman, A., et al.: Vehicular traffic optimisation and even distribution using ant colony in smart city environment. IET Intell. Transp. Syst. 12 (7), 594–601 (2018).
28. Jerry, K., et al.: NetLogo implementation of an ant colony optimisation solution to the traffic problem. IET Intell. Transp. Syst. 9(9), 862–869 (2015)
29. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. Netw. 1 (4), 397–413 (1993)
30. Srinivas, M., Patnaik, L.M.: Genetic Algorithms: A Survey. Computer 27(6), 17–26 (1994)
31. Roberge, V., Tarbouchi, M., Labonte, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. Ind. Informatics 9 (1), 132–141 (2013)
32. Chu, P.C., Beasley, J.E.: A genetic algorithm for the multidimensional knapsack problem. J. Heuristics 4(1), (1998)
33. Viveros-Jiménez, F., León-Borges, J.A., Cruz-Cortés, N.: An adaptive single-point algorithm for global numerical optimization. Expert Syst. Appl. 41 (3), 877–885 (2014)
34. Toğan, V., Daloğlu, A.T.: An improved genetic algorithm with initial population strategy and self-adaptive member grouping. Comput. Struct. 86 (11–12), 1204-1218 (2008)
35. Johnson, J.M., Rahmat-Samii, Y.: Genetic algorithms in engineering electromagnetics. IEEE Antennas Propag. Mag. 39 (4), 7–21 (1997)
36. Venkatesh, P., Gnanadass, R., Padhy, N.P.: Comparison and application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints. IEEE Trans. Power Syst. 18 (2), 688–697 (2003)
37. Dip, G., Prahlad, V., Kien, P.D.: Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed. Robotica, 27(3), 355–365 (2009)
38. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. 1, 69–93 (1991)
39. Blickle, T., Thiele, L.: A comparison of selection schemes used in evolutionary algorithms. Evol. Comput. 4 (4), 361–394 (1996)
40. Jebari, K., Madiafi, M.: Selection methods for genetic algorithms. Int. J. Emerg. Sci. 3, 333–344 (2013)
41. Reeves, C.R., Rowe, J.E., Genetic Algorithms—Principles and Perspectives. Kluwer, Boston (2002)
42. Kora, P., Yadlapalli, P.: Crossover operators in genetic algorithms: A review. Int. J. Comput. Appl. 6 (1), (2017)
43. Magalhães-Mendes, J.: A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. WSEAS Trans. Comput. 12 (4), 164–173 (2013)
44. Spears, W.M., Anand, V.: A study of crossover operators in genetic programming. in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin (1991)
45. Abdoun, O., Abouchabaka, J.: A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. 31 (11), 975–8887 (2012)
46. Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybern. 24 (4), 656–667 (1994)
47. Back, T.: Optimal mutation rates in genetic search. In: Proceedings of the fifth international conference on genetic algorithms, 1993, 2–8.
48. Hesser, J., Mäinner, R.: Towards an optimal mutation probability genetic algorithms. in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin (1991)
49. Greenwell, R.N., Angus, J.E., Finck, M.: Optimal mutation probability for genetic algorithms. Math. Comput. Model. 21(8), 1–11 (1995)
50. Lopez, P.A., et al.: Microscopic Traffic Simulation using SUMO. In: 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, 2575–2582.
51. "Madrid's Vehicle Fleet. 2018. [Online]. Available: bit.ly/2svNegb. Accessed: 28 Apr. 2019.
52. Borge, R., et al.: Comparison of road traffic emission models in Madrid (Spain). Atmos. Environ. 62, 461–471, (2012)
53. Yu, C., et al.: Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. Transp. Res. Part B Methodol. 112, 89–112 (2018)
54. Dresner, K., Stone, P.: Multiagent traffic management: An improved intersection control mechanism. In: Proceedings of the International Conference on Autonomous Agents, (2005).
55. Salman, M.A., Ozdemir, S., Celebi, F.V.: Fuzzy traffic control with vehicle-to-everything communication. Sensors (Switzerland), 18 (2), Art. no. 368 (2018)
56. Almeida, A.M.R., Macedo, J.A.F., Machado, J.C., Optimization of urban semaphore times turning into JSSP. In: CEUR Workshop Proceedings, 2018.
57. Mousavi, S.S., Schukat, M., Howley, E.: Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intel. Transport Syst. 11(7), 417–423 (2017)

How to cite this article: Guillen-Perez A, Cano M-D. Intelligent IoT systems for traffic management: A practical application. *IET Intell Transp Syst*. 2021;15:273–285. https://doi.org/10.1049/itr2.12021