Article in Press
IJIMAI journal

Traffic Optimization Through Waiting Prediction and Evolutive Algorithms

Francisco García[1], Helena Hernández[2], María N. Moreno-García[2], Juan F. De Paz[1]*, Vivian F. López[2], Javier Bajo[3]

[1] Expert Systems and Applications Lab. University of Salamanca. Plaza de los Caídos s/n. Salamanca (Spain)
[2] Data Mining Research Group. University of Salamanca Plaza de los Caídos s/n. Salamanca (Spain)
[3] Department of Artificial Intelligence, ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid (Spain)

Received 10 May 2022 | Accepted 22 September 2023 | Early Access 5 December 2023

Abstract

Traffic optimization systems require optimization procedures to optimize traffic light timing settings in order to improve pedestrian and vehicle mobility. Traffic simulators allow obtaining accurate estimates of traffic behavior by applying different timing configurations, but require considerable computational time to perform validation tests. For this reason, this project proposes the development of traffic optimizations based on the estimation of vehicle waiting times through the use of different prediction techniques and the use of this estimation to subsequently apply evolutionary algorithms that allow the optimizations to be carried out. The combination of these two techniques leads to a considerable reduction in calculation time, which makes it possible to apply this system at runtime. The tests have been carried out on a real traffic junction on which different traffic volumes have been applied to analyze the performance of the system.

Keywords

Evolutionary Algorithms, Prediction Systems. Traffic Optimization, Traffic Simulator.

DOI: 10.9781/ijimai.2023.12.001

I. Introduction

According to United Nations data, in 2018 55% of the population was living in urban spaces, the distribution of the urban population varies considerably by region: Northern America 82%, Latin America and the Caribbean 81%, Europe 74%, Oceania 68%, Asia 60% and Africa 43%. The urban population is continuously increasing; it is estimated that 66% of the population will live in urban areas by 2050, an increase of 16% compared to 2008 [1]. These data are very similar to those provided by the United Nations organization since in 2018 it estimated that 68% of the population will live in urban areas in 2050. This increase implies greater traffic congestion in cities due to both the increase in traffic and the unsuitable infrastructures [1]. For this reason, programs such as Horizonte Europa have analyzed global challenges such as climate, energy and mobility, and in particular, intelligent mobility through the optimization of infrastructures. Due to this increase in population and the need to improve infrastructure management, there is a demand to create systems capable of improving traffic efficiency, which will be applied in this project.

Traditional operational research incorporates the use of queuing theory to make predictions about different parameters such as waiting times [2]. The queuing theory approach in which there are usually M/G/s models [3] where M refers to the arrival of vehicles which is represented by a poisson, G the service rate which in certain cases can be modeled by an exponential and finally, s represents the number of servers. From these definitions, it is possible to determine parameters such as waiting times, which will be the object of study of this project. However, classical queuing theory would not take into account parameters that need to be considered, such as the time lost from the moment a traffic light turns green until the cars start moving. For these reasons, simulators such as SUMO [4] are currently being used for time estimation. The SUMO simulator uses an extension of the Gipps model [5] in which aspects such as user reaction time, braking time, or speed differences between the vehicles in the queue are taken into account. However, for this study the aim is not to apply the use of certain equations, but rather to create a system that is capable of estimating waiting times from traffic data obtained from SUMO simulations in order to subsequently perform optimizations.

In order to improve traffic efficiency, studies are mainly based on the analysis of intersections with or without traffic lights [6]. In the study [6] the convenience of introducing traffic lights at an intersection is analyzed by converting a nonlinear integer programming problem to linear integer programming in order to achieve an efficient resolution. The intersection problem is not restricted to decide only when it is more appropriate to introduce a traffic light, but it also involves the problem of dynamically controlling the timing of traffic lights to reduce waiting times [7] through the application of different techniques such as Bayesian networks [8], evolutionary techniques [9], reinforcement learning [10], fuzzy logic [11], [12]... Waiting times are usually associated with vehicles, but it is also relevant to consider pedestrian waiting times since they also have a relevant impact on vehicle waiting times at intersections.

* Corresponding author.
E-mail address: fcofds@usal.es

Please cite this article in press as:
F. García, H. Hernández, M. N. Moreno-García, J. F. De Paz, V. F. López, J. Bajo. Traffic Optimization Through Waiting Prediction and Evolutive Algorithms, International Journal of Interactive Multimedia and Artificial Intelligence, (2023), http://dx.doi.org/10.9781/ijimai.2023.12.001

In this project it is proposed a system that allows to cover two aspects, first, the system allows to make an estimation of waiting times through the use of different prediction techniques, which allows to calculate these waiting times without the need of testing with a simulator, which would require a high computational time. On the other hand, the system allows the optimization of traffic light configurations, thus reducing waiting times through evolutionary algorithms. The use of estimators allows a considerable time reduction, which makes this technique more dynamically applicable to traffic changes. The system has been tested on a real intersection on which different traffic flows have been applied in order to analyze the performance of the prediction systems and also of the optimization method applied.

The article is structured as follows: section 2 contains a description of the state of the art, section 3 the proposal for the data analysis and finally sections 4 and 5 the case study and the results obtained.

II. Related Works

Systems for the improvement of mobility in infrastructures are usually based on the management of intelligent traffic lights in which the timing of the different states can be changed dynamically [9]. In this review we will analyze different studies that determine the timing of the different traffic light states in order to reduce the waiting times of vehicles.

Among the studies that can be found are those based on fuzzy logic, which have been carried out for quite some time. For example, there is the work [13] from 1977, in which a study of time intervals and vehicle flow to manage an intersection was carried out. This work includes a model for traffic simulation, in which they consider different aspects in each traffic light cycle such as number of waiting vehicles, queues, saturation, and car delays in order to calculate the optimal time of the traffic lights. Subsequently, these studies were extended to more intersections [14], [15], [7], [16] and the simulator initially defined in [13] was also adapted by incorporating more intersections in one or more roads [17]. In more recent studies [11] a combination of Fuzzy Logic Controllers and genetic algorithms is performed to optimize the management of several intersections with traffic lights, this procedure allows using Fuzzy Logic to establish times, specifically the number of vehicles in the intersection is taken into account and applying fuzzy logic and the Mandami method the time interval of each traffic light is established. Genetic algorithms are used to maximize the number of vehicles crossing the intersections and fuzzy logic for the estimation of the green intervals of the traffic lights. In the paper [7] it is possible to find an extensive study on different works in which different defuzzification and memberships functions are applied. In some works such as [18] the combination of fuzzy logic and a neural network is analyzed to control the delay of the green state of traffic lights taking into account the size of the queue of cars. There are also works that attempt to improve traffic flow from route prediction through the use of regression methods for time estimation and fuzzy logic for the selection of the best route [19].

For the estimation of the time of traffic lights, procedures can be applied in order to determine the congestion levels, thus, in k [20] the congestion level is estimated through a time series from the use of decision trees, regression and neural networks to try to reduce pollution and energy consumption collecting data for five days. In the work [21] a prediction of congestion is also made through the use of neural networks such as LSTM (Long Short-Term Memory) and also regressors such as Support Vector Regression, Random Forest, Gradient Boosting Regression and other statistical techniques and it is verified how these systems are able to predict congestion from the creation of matrices that represent congestion, and to do so they use information from historical data of speeds, road maps, distances. In [22] an estimation of the daily traffic in England and Wales is made from the application of cluster and regression techniques such as Support Vector Regression (SVR) and Random Forest (RF). Likewise in the work [23] the SVR is applied to make a traffic flow prediction, but in this case other aspects such as meteorological factors are considered. In other study [24], traffic flow prediction is carried out through the use of an ARIMA model and an LSTM network that predicts the number of vehicles in 15-minute periods. First, the linear regression feature of the traffic data is captured by using ARIMA, then back-propagation is applied to train the LSTM network and capture the nonlinear features of the data, and finally, both results are combined based on the dynamic weighting of the sliding window. Using three sets of highway data, this method was compared with the other techniques separately (ARIMA, LSTM and EW) and it was determined that the proposed combined model has better prediction effects.

On the other hand, another parameter to be used to describe the traffic flow can be the average speed of cars within a given period of time [25], generally focused on the short term. In this work, recurrent neural networks are explored using historical time data, as well as a number of contextual factors, including additional information such as date, week, etc., to determine how accurate the speed prediction is. A multi-layered RNN (two versions, one with LSTM and one with GRU) is used to learn the sequential traffic data, and a sparse autoencoder is used for the contextual data. Both outputs are merged and delivered to the predictor (neural network) to learn traffic patterns and predict future speed. The model was tested with two real-world data sets and compared with ten frequently used models, k nearest neighbor (k-NN), support vector machine (SVM), decision tree (DT), gradient booting decision tree (GBDT), random forest (RF), stacked autoencoder (SAE), LSTM, GRU, Con-vLSTM, BiLSTM, showing that the proposed model (specifically the version with LSTM) performs better than the rest in terms of stability and accuracy.

Finally, due to the impossibility of considering, with existing algorithms, nonlinear historical data and other uncertain factors that influence peak-hour congestion, hybrid neural network algorithms such as CNN (Convolutional Neural Network) and LSTM are also proposed for short-term prediction of traffic flows based on multivariate analysis [26]. Traffic information is obtained from a Pavement Management System (PMS) that stores data from multiple detectors located throughout California, and weather information (such as temperature, humidity, etc.) from Mesowest. Experimental results show that the combination of CNN and LSTM obtains a high degree of accuracy compared to other models.

Another type of methods used for traffic optimization are reinforcement learning methods. These methods allow an agent to interact in a smart way with the environment in real time. At each instant of time, the agent perceives the environment, evaluates the policy, and performs the optimal action according to the policy. For each action performed by the agent, a reward is assigned according to whether this action brings the agent closer to or further away from the objectives. From previous observation-action pairs and their associated rewards, the agent is able to optimize its policy to maximize the rewards obtained.

Within these reinforcement learning methods, the most commonly used in traffic optimization are Q-learning based methods. The most common is traditional Q-learning with works such as [27], [28], [29], [30] and [31]. Other variants such as Deep Q-learning with works such as [31], [32], and [33]; and Double Deep Q-learning [34] also appear frequently in the state of the art. However, although they are the most common, Q-learning based techniques are not the best performers. The best results are obtained by other algorithms such as SARSA [27] or variants of Actor-Critic, for instance, Traditional Actor-Critic [27], Advantage-Actor-Critic [35] o Deep Deterministic Policy Gradient [36].

In addition to the techniques used for traffic optimization, it is also relevant to consider the alternatives in traffic simulators. There are several software packages for traffic simulation that offer different functions and, therefore, it is common to find articles that use different options according to the needs of the study or, sometimes, several simulators within the same study in order to make comparisons.

For instance, the VISSIM software package can be used to train an algorithm using reinforcement learning to optimize the safety of signalized intersections [37] This same traffic simulator is also used simultaneously with TransModeler [38] as a comparison to demonstrate that the proposed SSAT model offers better performance when applied to simulate mixed traffic on two-way, two-lane roads. Furthermore, in the work [39] the ability of the CORSIM software package to replicate the highway failure process is assessed and a sensitivity analysis is performed on different driver behavior parameters to determine the effect of these on such failures. However, the simulator finally chosen to carry out the tests on the study intersection was SUMO due to the widespread use of this simulator in the scientific field such as [30], where this software is used to obtain traffic information which will be used in a Q-learning algorithm to create a TSC system that maximizes the number of vehicles passing through an intersection; or [9], where it is used to evaluate in real time the performance of the proposed algorithms (swarm heuristic optimization algorithms, PSO) using real-world data (intersection in Turkey) to optimize traffic light control.

As illustrated above, there are studies that make use of regression techniques in traffic analysis, but these studies are focused on traffic flow estimation. In this work, the use of these techniques will be focused on the prediction of waiting times in order to reduce the time in the simulations to determine the behavior of different configurations without the need to perform a simulation. The use of this procedure would allow the system to adapt to different behaviors of the environment without being tied to any simulator, although SUMO will be used for testing. Subsequently, this data can be used with different optimization techniques, which will reduce computation time and thus improve its applicability to dynamic environments that require constant traffic adaptations.

III. Proposal

The proposed system consists of three components as shown in Fig. 1, the first of which would be the waiting time prediction part that allows estimating waiting times based on the time intervals of the traffic lights. In addition, it must be considered that it is necessary for the traffic lights to comply with some temporal relationships that are defined as contracts in such a way that the time intervals of a traffic light affect the time intervals of the rest. The second component is optimization. Optimization uses the first estimation component to generate from evolutionary algorithms an optimal configuration of traffic light times to reduce a certain parameter, in this case the waiting time of vehicles. The last component is the simulator, which is initially used to generate waiting time data from different configurations and use this information for the first estimation component; subsequently, the simulator component is replaced by the predictor component when performing the optimizations in the optimization component.

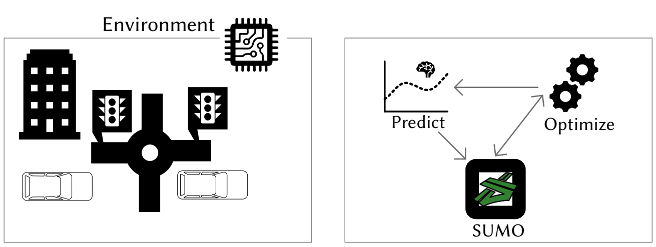


Fig. 1. System Components.

A. Prediction Component

The prediction component acts as a substitute for the traffic simulator during the optimization process. Therefore, its function is to estimate the waiting time at the traffic lights from the time intervals provided to the traffic lights. In this prediction component, techniques based on regressors and neural networks have been incorporated to estimate the waiting times. Specifically, Random-Forest [40], AdaBoost based on a decision tree [41], Bagging also based on a decision tree [42], ExtraTrees using the Gini index for the gain [43], and deep learning techniques and neural networks [44] have also been included to make the predictions within the prediction model.

Four different architectures are used within neural networks: a neural network with a single hidden layer, a neural network with multiple layers (specifically, 14 layers in the best performing one), a neural network with multiple layers and jump connections (specifically, 16 layers in the best performing one), and, finally, an LSTM.

All these models are trained using data generated by the simulator before the optimization process, but future work could use data obtained in real environments and remove the simulator completely from the system. In addition, adding inputs related to road conditions and structure to the predictors could move towards real-time traffic optimization by adapting to actual flow conditions.

B. Optimization

Traditionally, in traffic estimation studies, the optimizer launches multiple simulations with different parameters in the traffic simulator in order to evaluate its efficiency. These simulations are complex and have a high time cost, resulting in an inefficient optimization process. In this work, the optimizer does not communicate at any time with the traffic simulator; instead, the optimizer communicates only with the prediction component, greatly accelerating the optimization process in exchange for a small penalty in the time cost.

This optimization is performed using a particle optimization algorithm, but considering that, in this case, the particles correspond to configurations of the traffic lights and , therefore, there are some relationships and restrictions between them that must be fulfilled as their position is updated. For this reason, in Fig. 2 the information of the restrictions and relations between the traffic lights is included in order to limit the value in each of the iterations and thus obtain valid solutions. Each particle at time instant t is represented by $x_i(t)$, and will contain as many values as variables are being optimized. X is the set of particles, $v_i(t)$ is the velocity with which particle i moves, c_1 is the cognitive acceleration factor, c_2 is the social acceleration factor, p_i is the most optimal solution calculated for particle i, p stores the set of most optimal values for all particles, p_{beast} is the best calculated global solution. C_i contains the constraint for traffic light i, c_i^l is the lower bound for constraint i, and c_i^u is the upper bound for constraint i, C contains the set of constraints for all traffic lights, T is a set of values of the estimation of the time lost for each particle i.

C. Traffic Simulator

In this work, after evaluating several alternatives, SUMO was used as a traffic simulator. SUMO is a spatially continuous microscopic traffic simulator. This means that SUMO simulates vehicle-to-vehicle traffic flow in a non-discretized space.

SUMO uses the traffic model proposed by Stefan Krauß [45], [46]. This model is intended to be a simpler alternative to previous proposals, but, at the same time, to accurately capture traffic dynamics. The model is based on car tracking, i.e., the behavior of one car is conditioned by

// Update particles
repeat
foreach $(x_i) \in X$ **do**
$v_i(t+1) = v_i(t) + c_1 \cdot rand \cdot (p_i - x_i(t)) + c_1 \cdot rand \cdot (p_{best} - x_i(t))$
$x_i(t+1) = x_i(t) + v_i(t+1)$
end
// Update constraints according to the relation among semaphores
foreach $c_i \in C$ **do**
c_i = updateConstraint (X, C)
end
// Update particles according to the constraints
repeat
foreach $(x_i) \notin c_i$ **do**
if $x_i <$ then
$x_i(t+1) = c_i^l$
else
$x_i(t+1) = c_i^u$
end
end
until $\forall i x_i \in c_i$;
// Predict time loss with regressor
T = timeLossParticle (X, regressorCars, C, listSemaphores)
// Update local and global best
(p, p_{beast}) = updateLocalGlobal (T)
until;
// X matrix with particples
// regressorCars regressor to predict loos time
// C matrix with constraints semaphores
// listSemaphores list with each semaphore

Fig. 2. Optimization Process.

the positions and speeds of neighboring cars. Specifically, the S. Krauß model is based on the calculation of a maximum speed at which a car can go in such a way that it is impossible for it to collide with the cars it is following, considering a specific deceleration capacity and reaction time. Cars try to go at the maximum safe speed at all times. In addition, it introduces a stochastic term in the calculation of the current speed as a function of acceleration, which introduces a random element into the simulation.

Ultimately, continuous models offer greater accuracy at the cost of a performance penalty. Similarly, microscopic simulations offer greater accuracy than macroscopic simulations, which simulate the behavior of cars at a higher accuracy, again at the cost of a performance penalty. The model proposed by Krauß, despite being one of the simplest among the spatially continuous microscopic models, also carries a high time complexity that scales linearly with respect to the number of cars and the number of time instants simulated.

IV. Case Study

In order to simulate the intersection, the SUMO tool was used, which has a series of console commands that allow generating the flow of cars and pedestrians to carry out the simulation. In order to facilitate the continuous use of these commands and to automatize the process, all of them were grouped in a Python script that makes the necessary calls and creates the files required by SUMO to run the simulation. In the following section, we will describe how both the generation of cars and pedestrians from this file and the creation of the traffic light logic work.

For the generation of cars, we decided to create several flows for each of the routes that make up the intersection shown in Fig. 3, each of which can have a different number of cars, which is indicated as a variable within the Python script. This was done to have more control over the vehicles and to be able to make a model which was closer to reality, since otherwise the random generation could put all the cars on the same route.

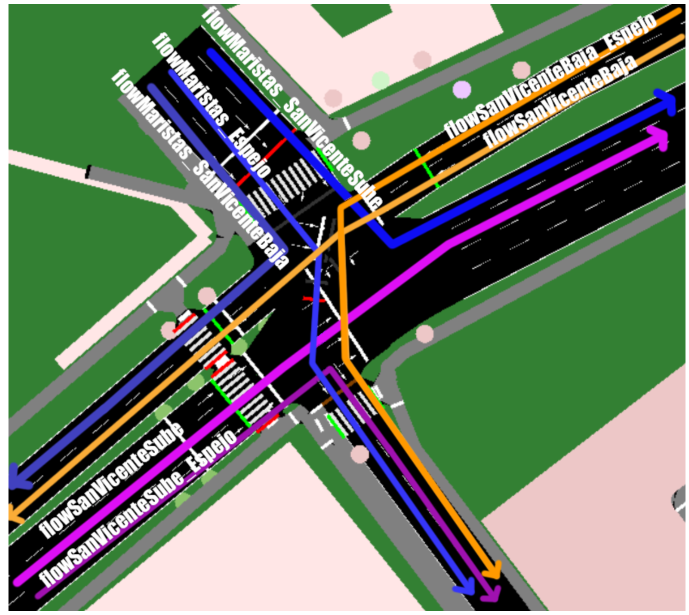


Fig. 3. Car flows of the different routes.

The value of these variables will be written inside an xml file (*flows.rou.xml*, Fig. 4) specifying the above-mentioned routes so that SUMO can understand them, that is, indicating the ids of the origin (*from*) and destination (*to*). In addition, the time that these flows will last (*end*), is also included, which must coincide with the time that the simulation is expected to last, and randomness is added to the frequency with which the cars of a flow are generated, since by default a uniform frequency is used. The following variable values were used for this case study:

- **simTime:** 1000 (Simulation duration in milliseconds)
- **flowSanVicenteBaja:** 20 cars
- **flowSanVicenteBaja_Espejo:** 20 cars
- **flowMaristas_Espejo:** 40 cars
- **flowMaristas_SanVicenteBaja:** 20 cars
- **flowMaristas_SanVicenteSube:** 20 cars
- **flowSanVicenteSube:** 20 cars
- **flowSanVicenteSube_Espejo:** 20 cars

Therefore, a maximum of 160 cars are generated in 1000 ms in total.

```
    f = open("flows.rou"+nombreFichero+".xml", "w")
    f.write("""<routes>
    <flow id="flowSanVicenteBaja" begin="0" end="{}" number="{}" from="44309629#3"
    to="45496883#2"/>
    <flow id="flowSanVicenteBaja_Espejo" begin="0" end="{}" number="{}"
    from="44309629#3" to="24616035#1"/>
    <flow id="flowMaristas_Espejo" begin="0" end="{}" number="{}"
    from="124430876#0" to="24616035#1"/>
    <flow id="flowMaristas_SanVicenteBaja" begin="0" end="{}" number="{}"
    from="124430876#0" to="45496883#2"/>
    <flow id="flowMaristas_SanVicenteSube" begin="0" end="{}" number="{}"
    from="124430876#0" to="207595128"/>
    <flow id="flowSanVicenteSube" begin="0" end="{}" number="{}"
    from="342268062#0" to="207595128"/>
    <flow id="flowSanVicenteSube_Espejo" begin="0" end="{}" number="{}"
    from="342268062#0" to="24616035#1"/>
</routes>""".format(simTime, flowSanVicenteBaja, simTime, flowSanVicenteBaja_Espejo
, simTime, flowMaristas_Espejo, simTime, flowMaristas_SanVicenteBaja, simTime,
flowMaristas_SanVicenteSube, simTime, flowSanVicenteSube, simTime,
flowSanVicenteSube_Espejo))
    f.close()

    # Randomizacin de los flujos para que la distancia entre coches sea aleatoria
    os.system("duarouter -n map.net.xml -r flows.rou"+nombreFichero+".xml
    --randomize-flows -o map.passengers.rou.xml")
```

Fig. 4. Flow generation in the xml file.

The generation of pedestrians is less complex since it is not divided into flows (since it is not possible to specify routes for them), so it is only necessary to indicate the maximum total number of pedestrians in the entire crossing (100 in this case study). In order to carry out this task, it was necessary to use the *randomTrips* tool included with SUMO, a Python script that allows to create a random file with pedestrian trips (*map.pedestrians.trips.xml*) by means of the following command:

python "%SUMO_HOME%\\tools\\randomTrips.py" -n map.net.xml -o map.pedestrians.trips.xml -r map.pedestrians.rou.xml -e **simTime** -p **pPed** -l --pedestrians --max-distance 500

Where simTime is the total simulation time (1000 ms), used to indicate for how many milliseconds pedestrians have to be generated, and pPed is the repetition rate, obtained by dividing the simulation time by the number of pedestrians. This is because the script generates pedestrians with a constant frequency of 1/pPed per second, so if 100 pedestrians must be generated in 1000 ms, the frequency should be 1000/100. Furthermore, the *--max-distance* option was used to set the maximum length of the trips, so that pedestrians would not be circulating for too long.

Finally, for the traffic light logic, an additional file (*traffic_lights.add.xml*) containing the durations of the green, yellow, red and amber phases for each of them. The goal was to reproduce the real operation of the traffic lights at the intersection, but at the same time allow to modify their durations to a certain extent. For this purpose, four variables are used, as shown in Fig. 5, from which the value of the other phases of the traffic lights are calculated so that the real configuration is respected.

- **green1:** Green time of the traffic light of San Vicente Uphill (Must be less than or equal to green2)
- **green2:** Green time of the traffic light of San Vicente Downhill
- **green3_d:** Green time of the traffic light of Maristas (right lanes, must be greater than or equal to amber3_i)
- **yellow3_i:** Amber time of the traffic light of Maristas (left lanes) The green time will be calculated by subtracting this value from green3_d, so that the left lanes are at most the same time on green as the right lanes and, if they last less, the rest will be on amber.

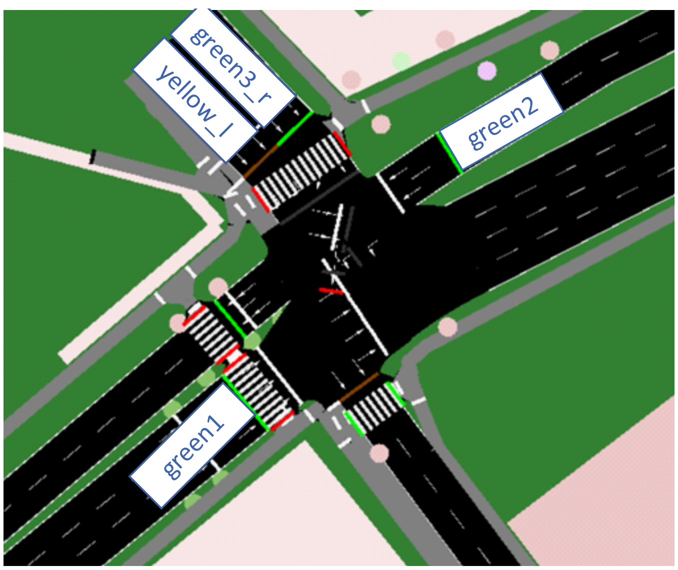


Fig. 5. Variables for traffic lights durations.

V. Results

In this section, the results obtained with the proposed method are discussed, both those of the models for estimating the time lost by vehicles at traffic lights (subsection V.B) and those of the traffic optimization algorithm (subsection V.C). In addition, there is a section in which we discuss why we consider only the time lost by vehicles instead of both vehicles and pedestrians (subsection V.A).

A. Time Lost by Pedestrians and Time Lost by Vehicles

Since the methods consulted in the state of the art only consider the time lost by vehicles to perform traffic optimization, at the beginning of this work, one of the novelties intended to be included was to consider the time lost by pedestrians at traffic lights when performing this optimization. However, when evaluating the results of the optimization using the proposed method, the total lost time (sum of the time lost by vehicles and the time lost by pedestrians) predicted for specific traffic signal times was far from the time calculated by the traffic simulator.

In order to analyze this discrepancy, 1000 runs of the traffic simulator were performed with the same traffic light times, specifically, those predicted as optimal by the optimizer. In these runs, both the time lost by pedestrians and the time lost by vehicles were collected, analyzed and plotted as the density plot shown in Fig. 6. In addition to the optimal times, two other tests were also performed with different values for traffic light times but with similar results. The conclusion of these tests is that introducing randomness in the pedestrian paths introduced a standard deviation of less than 1 second in the time lost by vehicles, but of about 3.5 seconds in the time lost by pedestrians with a difference of more than 20 seconds between the minimum and maximum. In contrast, by repeating these tests with constant pedestrian recoveries and introducing randomization in the vehicle paths, the standard deviation of both lost times is very close to zero.

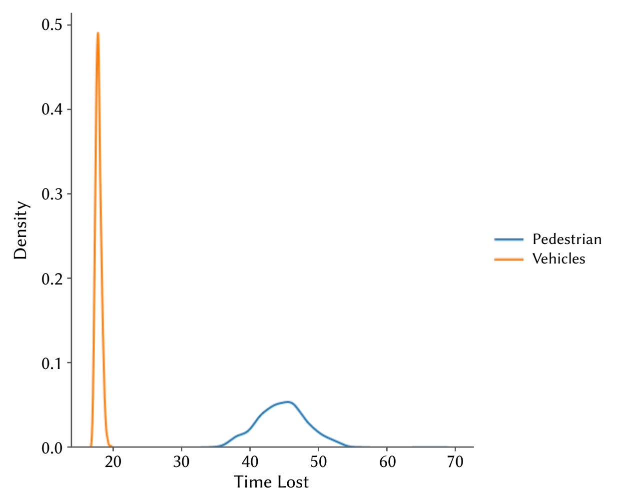


Fig. 6. Density plot of time lost by vehicles and pedestrians for different pedestrian random paths.

The reason behind this disparity in the values of time lost by pedestrians when randomness is introduced for pedestrians, but not when randomness is introduced for vehicles, is due to the implementation of the traffic simulator used, SUMO. Specifically, SUMO allows us to specify the number of vehicles that will make a route between a specific origin and a specific destination, but it generates the pedestrian routes in a completely random way.

Thus, in order to guarantee reproducibility and considering that this discrepancy in the results was due to the implementation of the traffic simulator itself, there are two possible solutions to the problem: eliminating the randomness introduced in the pedestrians or considering only the time lost by vehicles, which have a smaller deviation. Since limiting the randomness to only vehicles could introduce a bias towards a certain number of pedestrian paths, it was decided to consider only the time lost by vehicles to perform the optimization.

B. Comparison Between Models for Estimating Lost Time at Traffic Lights

As mentioned in Section 3, this project has analyzed the use of several artificial intelligence models to estimate the time lost by vehicles at traffic lights. The methodology used to evaluate these models is described below and a comparison of results is provided.

In order to train these estimation models, we used a dataset generated from the lost time at traffic lights retrieved from multiple simulations for different traffic light times using SUMO. Specifically, these simulations were performed on the scenario described in the case study. A total of 625,000 simulations were performed corresponding to all possible combinations giving values between 1 and 50 seconds to each of the green times of the different traffic lights. In addition, in the case of neural network architectures [43], the hyperparameters were selected using the Bayesian hyperparameter as the tuning method and the mean MAE over a cross-validation of 10 folds as the evaluation criteria. The optimizer used was Adam and the batch size (131,072 samples) was selected to maximize GPU utilization.

To evaluate the performance of each of the models, cross-validation of 10 iterations and the mean absolute error (MAE) metric on each of them was used. The mean of the results over these 10 iterations is given in Table I with a 95% confidence interval.

TABLE I. Mean MAE and NMAE of the 10-Fold Cross-Validation of Each Method With 95% Confidence Interval

Method	MAE	NMAE
Random Forest	0.094 ± 0.000	0.00083 ± 0.00000
Ada Boost	5.488 ± 0.234	0.04840 ± 0.00206
Bagging	0.094 ± 0.000	0.00083 ± 0.00000
Extra Trees	0.006 ± 0.000	0.00005 ± 0.00000
LSTM	1.629 ± 0.004	0.01437 ± 0.00004
Shallow Network	1.761 ± 0.009	0.01553 ± 0.00008
Deep Network	1.143 ± 0.041	0.01008 ± 0.00036
Residual Network	0.873 ± 0.007	0.00770 ± 0.00006

Even though it would be possible to determine which methods perform better from the values available in this table, it was decided to use the Mann Whitney hypothesis validation test to ensure that this assessment has a certain statistical reliability. Specifically, two separate tests were performed for all possible pairs of methods. The first one had as null hypothesis the equality of the results between pairs of methods and as alternative hypothesis the inequality of the results between them. The second test had as the null hypothesis the inferiority of the results of the first method and the alternative hypothesis the superiority of the results of the first method. The results of these tests are shown in Fig. 7.

According to these graphs, it can be shown that the results of the Bagging Regressor and the Random Forest Regressor are equivalent, while the rest of the methods are quite different from each other. On the other hand, it can be observed that the results of the Extra Trees Regressor outperform the results of the other methods while the results of the Ada Boost Regressor are inferior to the rest. Furthermore, it can be observed that the results of neural network based methods are worse than those of traditional Machine Learning algorithms, with the exception of Ada Boost. Based on these observations, it was decided to use the Extra Trees Regressor algorithm as a method for estimating the waiting time at traffic lights in order to optimize traffic flow.

C. Traffic Optimization

The traffic optimization experiments were developed using Python scripts which use a particle optimization algorithm implemented in the pyswarm library. Two implementations were performed. The first one used the SUMO simulator to calculate the waiting times of vehicles at traffic lights and the second one used the Extra Trees Regressor algorithm to calculate an approximation. In the first case, the run lasted 1 day, 7 hours, 33 minutes, and 48 seconds. In the second case, the run lasted 2 minutes and 26 seconds. Although the difference in execution time is dramatic, the results obtained are very similar. Specifically, in the first case, the average waiting time for vehicles was 16.66 seconds and in the second case, 17.26 seconds.

VI. Conclusions and Future Enhancements

We have developed a system capable of optimizing traffic based on particle flooding which improves its performance by replacing the traffic simulator with an estimation system based on machine learning algorithms. Several estimation methods have been analyzed and the one with the best results, the Extra Trees Regressor, has been selected. Finally, the loss of precision in the results when using our method has been evaluated and it has been observed that the resulting waiting time when using the approximator is 0.6 seconds longer than when using the simulator. However, the computation time when using the simulator (113627.74892 seconds) is up to 777 times longer than when using the approximator (146.138249 seconds).

In future studies, we will analyze the use of other traffic optimization algorithms and their compatibility with our Extra Trees Regressor-based approach. Furthermore, mechanisms will be studied to allow the approximation algorithm to be able to generalize to other intersections without the need for a complete retraining. Finally, we will consider the development of a system capable of collecting data to train the estimator automatically by analyzing images obtained by cameras implanted in the traffic lights.

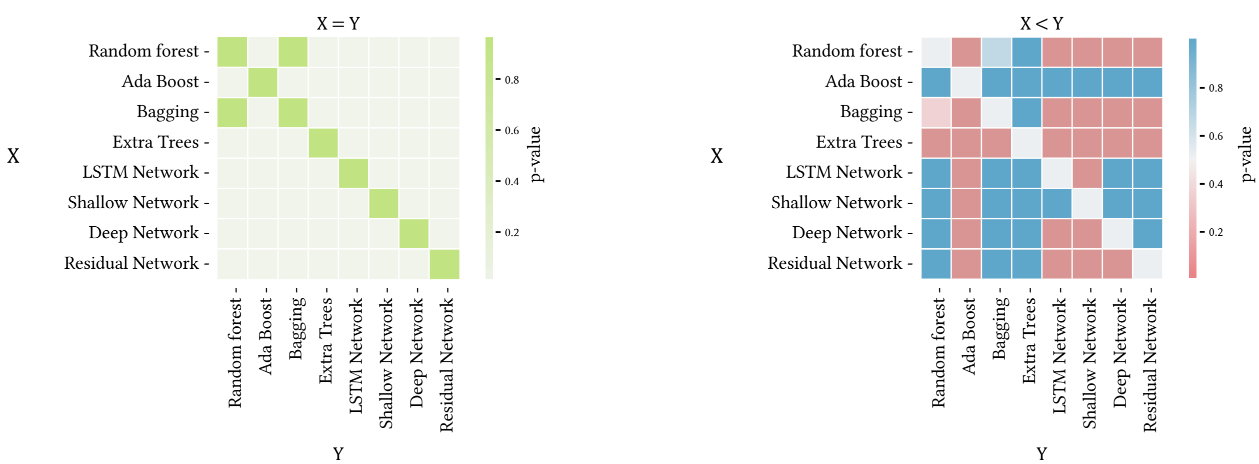


Fig. 7. P-value matrices of the Mann Whitney test for equality and inferiority of the results of the different methods, respectively.

Acknowledgment

This work was supported by the Spanish Agencia Estatal de Investigación. Project Monitoring and tracking systems for the improvement of intelligent mobility and behavior analysis (SiMoMIAC). Sistema de procesamiento distribuido para la optimización en movilidad y transporte - Distributed processing system for mobili-ty and transport optimization PID2019-108883RB-C21 / AEI / 10.13039/501100011033. PID2019-108883RB-C22 - Sistema de monitorizacion y control para el apoyo a la toma de decisiones en el mantenimiento de infraestructuras. Francisco García Encinas' research was partly supported by the Spanish Ministry of Universi-ties (FPU Fellowship under Grant FPU19/02455).

References

[1] S.M. Khan, M. Rahman, A. Apon, M. Chowdhury, "Chapter 1 - Characteristics of Intelligent Transportation Systems and Its Relationship With Data Analytics", Data Analytics for Intelligent Transportation Systems, Elsevier, 2017, pp. 1-29.

[2] C. Zato, A. de Luis, J. Bajo, J.F. de Paz, J.M. Corchado, "Dynamic model of distribution and organization of activities in multi-agent systems", Logic Journal of the IGPL, vol. 20 no. 3, pp. 570-578.

[3] D.A. Menasce, "Trade-offs in designing web clusters", Internet Computing, I vol. 6, 2002, pp. 76–80.

[4] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.P. Flötteröd, R. Hilbrich, Leonhard Lücken, J. Rummel, P. Wagner, Evamarie Wießner, "Microscopic Traffic Simulation using SUMO", IEEE Intelligent Transportation Systems Conference (ITSC), 2018.

[5] D. Krajzewicz, G. Hertkorn and P. Wagner, Christian Rössel, "SUMO (Simulation of Urban MObility) An open-source traffic simulation", In: MESM2002 Proceedings. Comingout Okt. 2002

[6] Y. Zhang, R. Su, "An optimization model and traffic light control scheme for heterogeneous traffic systems", Transportation Research Part C: Emerging Technologies, vol. 124, 102911, 2021.

[7] C. Karakuzu, O. Demirci, "Fuzzy logic based smart traffic light simulator design and hardware implementation", Applied Soft Computing, vol. 10, no.1, 2010, pp. 66-73.

[8] X. Zhengxing, J. Qing, N. Zhe, W. Rujing, Z. Zhengyong, H. He, S. Bingyu, W. Liusan, W. Yuanyuan, "Research on intelligent traffic light control system based on dynamic Bayesian reasoning", Computers & Electrical Engineering, vol. 84, 2020.

[9] S.A.Celtek, A. Durdu, M.E.M. Alı, "Real-time traffic signal control with swarm optimization methods", Measurement, vol. 166, 108206, 2020.

[10] M. Greguri´c, M. Vuji´c, "Charalampos Alexopoulos and Mladen Mileti. Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data", Applied Sciences, vol. 10, 4011, 2020.

[11] S.M. Odeh, A.M. Mora, M.N. Moreno, J.J.Merelo, "A Hybrid Fuzzy Genetic Algorithm for an Adaptive Traffic Signal System", Advances in Fuzzy Systems. 2015.

[12] A. Ikidid, A. El Fazziki, M. Sadgal, "Multi-Agent and Fuzzy Inference-Based Framework for Traffic Light Optimization", International Journal of Interactive Multimedia and Artificial Intelligence, 2021,

[13] C. Pappis, E. Mamdani, "A fuzzy logic controller for a traffic junction IEEE transactions on Systems", Man and Cybernetics SMC-7/10, 1977, pp. 707-717,.

[14] M. Nakatsuyama, H. Nagahashi, N. Nishizuka, "Fuzzy logic phase controller for traffic functions in the one-way arterial road", IFAC 9th Triennial World Congress, Pergamon Press, 1984 pp. 2865-2870.

[15] J. Favilla, A. Machion, F. Gomide "Fuzzy traffic control: adaptive strategies" Second IEEE International Conference on Fuzzy Systems II, 1993, pp. 506-511.

[16] S.Komsiyah, E.Desvania, "Traffic Lights Analysis and Simulation Using Fuzzy Inference System of Mamdani on Three-Signaled Intersections", Procedia Computer Science, vol. 179, 2021, pp. 268-280.

[17] C.H. Chou, J.C. Teng, "A fuzzy logic controller for traffic junction signals", Information Sciences, vol. 143, no. 1–4, 2002, pp. 73-97.

[18] X. Fan, Y. Liu, "Alterable-Phase Fuzzy Control Based on Neutral Network", Journal of Transportation Systems Engineering and Information Technology, vol. 8, no. 1, 2008, pp. 80-85.

[19] A.S.Tomar, M. Singh, G. Sharma, K.V. Arya, "Traffic Management using Logistic Regression with Fuzzy Logic", Procedia Computer Science, vol. 132, 2018, pp. 451-460.

[20] T.S. Tamir, G. Xiong, Z. Li, H. Tao, Z. Shen, B. Hu, H.M. Menkir, "Traffic Congestion Prediction using Decision Tree", Logistic Regression and Neural Networks, IFAC-PapersOnLine, vol. 53, no. 5, 2020, pp. 512-517.

[21] M. Bai, Y. Lin, M. Ma, P. Wang, L- Duan, PrePCT: "Traffic congestion prediction in smart cities with relative position congestion tensor", Neurocomputing, vol. 444, 2021, pp. 147-157.

[22] A. Sfyridis, P. Agnolucci, "Annual average daily traffic estimation in England and Wales: An application of clustering and regression modelling", Journal of Transport Geography, vol. 83, 102658, 2020.

[23] X. Bao, D. Jiang, X. Yang, H. Wang, "An improved deep belief network for traffic prediction considering weather factors", Alexandria Engineering Journal, vol. 60, no. 1, 2021, pp. 413-420.

[24] S. Lu, Q, Zhang, G. Chen, Dewen Seng, "A combined method for short-term traffic flow prediction based on recurrent neural network", Alexandria Engineering Journal, vol. 60, no. 1, 2021, pp. 87-94,

[25] L. Qu, J. Lyu, W. Li, D. Ma, H. Fan, "Features injected recurrent neural networks for short-term traffic speed prediction", Neurocomputing, vol. 451, 2021, pp. 290-304.

[26] S. Narmadha, V. Vijayakumar, "Spatio-Temporal vehicle traffic flow prediction using multivariate CNN and LSTM model", Materials Today: Proceedings, 2021,

[27] M. Aslani, S. Seipel, M.S. Mesgari, M. Wiering, "Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown tehran", Advanced Engineering Informatics, vol. 38, 2018, pp. 639–655.

[28] M. Abdoos, A.L.C. Bazzan, "Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long short term memory", Expert Systems with Applications, vol. 171, 2021, pp. 114580.

[29] M. Essa, T. Sayed, "Self-learning adaptive traffic signal control for real-time safety optimization", Accident Analysis & Prevention, vol. 146, 2020, pp. 105713.

[30] H. Joo, S.H. Ahmed, Y. Lim, "Traffic signal control for smart cities using reinforcement learning", Computer Communications, vol. 154, 2020, pp. 324-330.

[31] E. Walraven, M.T.J. Spaan, B. Bakker, "Traffic flow optimization: A reinforcement learning approach", Engineering Applications of Artificial Intelligence, vol. 52, 2016, pp. 203-212.

[32] T.M. Aljohani, A. Ebrahim, O. Mohammed, "Real-Time metadata-driven routing optimization for electric vehicle energy consumption minimization using deep reinforcement learning and Markov chain model", Electric Power Systems Research, vol. 192, 2021, pp. 106962.

[33] S. Koh, B. Zhou, H. Fang, P. Yang, Z. Yang, Q. Yang, L. Guan, Z. Ji, "Real-time deep reinforcement learning based vehicle navigation", Applied Soft Computing, vol. 96, 2020, pp. 106694.

[34] Y. Gong, M. Abdel-Aty, J. Yuan, Q. Cai, "Multi-Objective reinforcement learning approach for improving safety at intersections with adaptive traffic signal control", Accident Analysis & Prevention, vol. 144, 2020, pp. 105655.

[35] H. Maske, T. Chu, U. Kalabić, "Control of traffic light timing using decentralized deep reinforcement learning", IFAC-PapersOnLine, vol. 53, no. 2, 2020, pp. 14936-14941.

[36] Z. Li, H. Yu, G. Zhang, S. Dong, C.Z. Xu, "Network-wide traffic signal control optimzation using a multi-agent deep reinforcement learning", Transportation Research Part C: Emerging Technologies, vol. 125, 2021, pp. 103059.

[37] M. Essa, T. Sayed, "Self-learning adaptive traffic signal control for real-time safety optimization", Accident Analysis & Prevention, vol. 146, 105713, 2020.

[38] J. Sun, H. Liu, Z. Ma, "Modelling and simulation of highly mixed traffic flow on two-lane two-way urban streets", Simulation Modelling Practice and Theory, vol. 95, 2019, pp. 16-35, .

[39] A. Kondyli, I. Soria, A. Duret, L. Elefteriadou, "Sensitivity analysis of CORSIM with respect to the process of freeway flow breakdown at bottleneck locations", Simulation Modelling Practice and Theory, vol. 22,

2012, pp. 197-206.

[40] L. Wiene, A. Liaw, M. Wiener "Classification and regression by randomForest", R News, 2/3, 2002, pp. 18-22.

[41] Y, Freund Y., R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of Computer and System Sciences, vol,. 55, no.1, 1977, pp. 119-139.

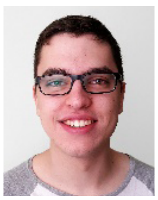
[42] D. Hernández-Lobato, G. Martínez-Muñoz, A. Suárez, "Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles", Neurocomputing, vol. 74, no.12–13, 2011, pp. 2250-2264.

[43] P. Geurts, D. Ernst, I. Wehenkel, "Extremely randomized trees", Machine Learning, vol. 63, no. 1, 2006, pp. 3-42.

[44] F. García Encinas, H. Hernández Payo, J.F. de Paz Santana, M.N. Moreno García, J. Bajo Pérez, "Estimating Time Lost on Semaphores with Deep Learning". New Trends in Disruptive Technologies, Tech Ethics and Artificial Intelligence, vol. 1410, 2022. Springer, Cham. https://doi.org/10.1007/978-3-030-87687-6_4

[45] S. Krauss, P. Wagner, C. Gawron "Metastable States in a Microscopic Model of Traffic Flow", Physical Review E, , vol. 55, no. 5, 1997, pp. 5597–602, https://doi.org/10.1103/PhysRevE.55.5597.

[46] S. Krauß "Microscopic modeling of traffic flow: Investigation of collision ree vehicle dynamics", 1998.



Francisco García Encinas

Francisco García Encinas studied a bachelor's degree in Computer Engineering at the University of Salamanca, Spain, from 2015 to 2019. Later, from late 2018 to mid-2019 he worked as a scientific software programmer at ALF-USAL research group at the University of Salamanca. In 2020, he got a master's degree in Intelligent Systems at the same university, and, nowadays, he is a Ph.D. student at ESALab research group.

Helena Hernández Payo

Helena Hernández Payo graduated from the University of Salamanca in 2021 where she obtained a bachelor's degree in Computer Engineering. From 2020 to 2021 she did an internship as software programmer for the research group GRIAL and at the same time contributed to the creation of the simulation described in this article as part of a collaboration scholarship at the University of Salamanca. Currently she is working as a Big Data developer at Viewnext.

María N. Moreno García

María N. Moreno García is currently Full Professor at the University of Salamanca, Spain, and head of the Data Mining Research Group of the same University (mida.usal.es). She has been a research scholar at the Intelligent System Lab of the University of Bristol, UK, and at the College of Computing and Digital Media of the DePaul University in Chicago, USA. Her research interests are in the areas of Data Science, Machine Learning and their application in different domains.

Juan Francisco De Paz Santana

Juan Francisco De Paz Santana received the degree in technical engineering in systems computer sciences, in 2003, and the Engineering degree in computer sciences, the degree in statistics, and the Ph.D. degree in computer science from the University of Salamanca, Spain, in 2005, 2007, and 2010, respectively. He is currently a Full Professor with the University of Salamanca, where he is also a Researcher with the Expert Systems and Applications Laboratory (ESALab). He has been a coauthor of published articles in several journals, workshops, and symposiums.

Vivian Félix López Batista

Vivian Félix López Batista received a PhD. in Computer Science from the University of Valladolid in 1996. At present she is a Full Professor of Computer Science at the University of Salamanca (Spain) where she has been since 1998. Member of the Data Mining Group (http://mida.usal.es/). She has done research on Natural Language Processing, Machine Learning and Neural Networks. She has also papers published in recognized journals, workshops and symposiums, books, and book chapters in these topics. She performed a research stay in the Center for Computational Science, University of Miami.

Javier Bajo

Javier Bajo received a PhD in computer sciences, in 2003, from the University of Salamanca, Spain. He is currently a Full Professor at the Department of Artificial Intelligence at the Universidad Politécnica de Madrid. He is also Director at the UPM Research Center in Artificial Intelligence. He has been a coauthor of more than 300 articles published in recognized journals, international conferences, workshops, and symposiums.