2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)
October 15-18, 2019. Diez Hotel, Medellin, Colombia.

Control Strategies for Urban Traffic Tested in a Co-Simulation Framework

Andres Insuasty, Gabriel Coral, and Andrés Pantoja
Departamento de Electrónica
Universidad de Nariño
Pasto, Colombia
andresinsuastyd10@gmail.com, gabrielcoralc95@gmail.com, ad_pantoja@udenar.edu.co

***Abstract*—The management of urban traffic is a challenging task due to its complexity and importance in modern cities. This work propoeses two control strategies for a study case developed in a co-simulation platform between a specialized traffic simulator (VISSIM) and Matlab to implement the control methods. The proposed strategies (Particle Swarm Optimization and Replicator Dynamics) are compared with a fixed-time technique (Webster) to analize the influence of the adaptive methods on some performance indexes such as queue lenghts, vehicle fluxes, and fuel consumption in a real network.**

***Index Terms*—Co-simulation, Particle Swarm Optimization, Replicator Dynamics, Urban Traffic Control.**

I. INTRODUCTION

Urban transportation as a whole is a high-impact system to people's daily life and cities development. The negative consequences of saturated traffic levels include significant time losses and increasing pollution due to the high fuel consumption. As an additional consequence, the excess of traffic produces health issues related to unacceptable sound environments and low air quality, with direct and indirect implications on general welfare, sleep disorders, anxiety, depression, and even, heart diseases [1].

The development of advanced traffic-management strategies and novel technologies on smart signal lights [2] allow the transportation system to reduce the traffic congestion. These new techniques are based on a coordinated system of lights, sensors, and controllers in charge of a growing vehicle density over almost all cities. The different methodologies for traffic control can be divided in two groups. The open-loop strategies are implemented with fixed times in the lights, taking into account historical data and statistics on the vehicle fluxes. On the other hand, the closed-loop strategies are more sophisticated and sensitive to the traffic demand in each moment.

The closed-loop control is a complex algorithm based on real-time measurements of the vehicle inputs to each intersection. For instance, an ANFIS model (Adaptive Network Based Fuzzy Inference Systems) is used in [3] to optimize the flux in a system with 8 street intersections, by controlling synchronization and phasing times. As a result, the vehicular density and the average number of queued vehicles are reduced by 42.6% and 61.7%, respectively, in comparison to the results for open-loop strategies. In a similar way, [4] presents evolutionary algorithms to solve a multiobjective optimization problem in a road corridor, obtaining improvements for the speed of traffic in buses (15.3%) and private cars (24.8%). Moreover, researches on advanced control strategies show significant improvements over several performance indices. Control techniques such as optimal control (MPC) [5], [6], adaptive control [7], bioinspired strategies [8], [9], and intelligent control [10], present better fluxes, fuel consumption, and delays, among other indicators, in comparison to fixed-time techniques.

Due to the complexity of a real traffic system, the validation of the control strategies is not carried out in actual transportation networks or general-purpose simulators. The stochastic behavior of vehicles are emulated by specialized platforms that take into account many characteristics of individual cars and groups of vehicles, such as constraints in fluxes, allowed directions, and geometry of the ways, just to name a few. In this work, we use VISSIM as a platform with large available technical information, high precision in simulations, and extended use in research [11]. To program the interaction between the controller and the traffic system, we use a co-simulation environment, where the plant is simulated in VISSIM to provide the data about vehicle fluxes to the control scheme programmed in MATLAB, in charge of processing the information and feeding the control signals (light times in the semaphores) back to the traffic system.

In complement, as the main contribution of the work, we propose two control strategies (a modified Particle Swarm Optimization and Replicator Dynamics) over a real scenario of two street intersections. In these algorithms, the time of each signal is a resource that is allocated in fixed and variable cycles in response to the stochastic changes in the shared fluxes of the interconnected lanes. The results of the control methods are compared with an open-loop controller through several performance indices.

The rest of the work presents the description of the proposed strategies in the next section. Then, the co-simulation structure is presented in Section III to introduce the study case. Finally, the simulation results are analyzed and compared in Section IV.

978-1-5386-6962-4/19/$31.00 ©2019 IEEE

II. Traffic-Control Strategies

Traffic-management systems use coordinated semaphores in the street intersections to direct the vehicle flux, which is the amount of cars crossing by a point in a time unit. The flux depends on the signals, corresponding to a single light or light combinations at the same time, such as the ones required for straight-and-turn signaling.

The term phase is defined as the time associated with the signaling of one or more independent movements that receive simultaneously the right-of-way. A phase begins with the loss of the right-of-way of the vehicles that are in conflict with the ones that win the right to cross. With several traffic lights in a set of intersections, a cycle is defined as the time required to complete a sequence of all the phases in the semaphores.

Some of the most used measures to evaluate the performance of traffic systems include the queue length in a street, which is the distance in meters of the number of vehicles pending for the right-of-way. The saturation percentage at an intersection is a measure of the current demand related to the total capacity that depends mainly on the geometry of the streets. Moreover, other indices can be calculated based on the previous measures such as the fuel consumption, waiting times, delays, and polluting emissions, among others.

A general classification of traffic-control strategies defines the fixed-time and variable-time controllers. The first one refers to a control where the cycle, duration, and sequence of intervals are invariable and defined previously. The calibration depends on historic data that must be evaluated every certain time. The controllers with variable time are more complex and allow the system to improve the performance of the network according to measurements of the vehicle flux at each intersection. They are usually implemented in three basic steps, starting with the collection of information (vehicle amounts, queue lengths, and saturation values, among other measures) to calculate some indicators that must be optimized. Then, an algorithm is executed for the adjustment of the times in the traffic lights (i.e., the control variables), and finally, adjust the parameters to improve the performance in the next cycle [7].

A. A Fixed-Time Control Strategy

The control methods with fixed times determine the phases of each traffic light based on data and statistical measurements prior to the implementation [12], [13]. Based on field observations and simulation of traffic conditions, F. V. Webster in [14] determines an optimal cycle time length given by

$$C = \frac{1.5L + 5}{1 - \sum_{i=1}^{\varphi} Y_i}, \tag{1}$$

where L is the loss time in seconds per cycle, φ is the number of phases, and Y_i is the maximum value of the ratio between the average and saturation flows for lane i. The loss time is given by the duration of the phases in yellow and red, which are respectively defined by

$$T_y = t + \tfrac{v}{2a} \quad \text{and} \quad T_r = \tfrac{W + L_v}{v}, \tag{2}$$

where t represents the perception or reaction time of the driver, v is the average velocity of the vehicles, a is the deceleration rate, W is the width of the intersection, and L_v is the queue length of the vehicles. Then, the loss time in each cycle is $L = T_r + T_y$.

To calculate the minimum delay at the intersection, the total effective green time of the cycle $(C - L)$ is proportionally distributed according to the saturation ratio values Y_i

$$G_i = \frac{Y_i(C - L)}{\sum_{j=1}^{\varphi} Y_j}, \tag{3}$$

expression in which G_i is the effective green time for the phase i. With the previous formulae, the optimal times are established for each phase, which are permanently programmed in the traffic-control system.

B. Particle Swarm Optimization (PSO)

This technique is based on models of social behaviors, such as the movements of flocks of birds or schools of fish. This algorithm optimizes a problem with a population of individuals that represents the candidate solutions. Individuals can move around the search space according to simple criteria and restrictions, looking for a better solution [9], [15].

The update criterion for the movement of the i^{th} individual takes into account the best personal position $(pBest_i)$ and the best global position $(gBest)$ of an agent in the group. These two parameters, including a random factor, influence the speed of movement of each individual in the search space at the next time interval. In each iteration of the algorithm, the velocity and position of the agents are updated according to [15]

$$V_i(k+1) = wV_i(k) + c_1 r_1 (pBest_i - X_i(k)) + ...$$
$$... + c_2 r_2 (gBest - X_i(k))$$

and

$$X_i(k+1) = X_i(k) + V_i(k+1). \tag{4}$$

In these equations, $X_i(k)$ is the position of the i^{th} individual at the instant k, w is the inertial weight, c_1 and c_2 are positive coefficients that determine the relative influence of $pBest_i$ and $gBest$, respectively, and r_1, r_2 are random values in the interval $[0, 1]$ to provide the stochastic component of the search. The position of each individual is evaluated by a fitness function. If the new position of the individual is better than the previous one, $pBest_i$ is replaced by X_i, and if a better global position is discovered, $gBest$ is also updated.

To use the PSO algorithm in traffic control, it is assumed that there exist φ movement phases, including all the traffic lights. Each phase corresponds to one individual belonging to the PSO swarm, where the position of each agent X_i, represents the green time of its phase. According to the problem, the objective function or *fitness* that each individual should minimize is defined by

$$f_i = \frac{Qlen_i}{Cap_i}, \tag{5}$$

which represents the relationship between the current queue length $Qlen_i$ and the maximum capacity of the lane Cap_i of the semaphore i. Clearly, the minimization of these fitness functions represents an improvement in the network and is a direct consequence of the implementation of the green time of each phase X_i.

The algorithm, at the beginning of each cycle C, evaluates the queue length for each lane to obtain the value of the fitness (5). Then, the speed and position are calculated using (4), taking into account the more recent values of $pBest_i$ and $gBest$. These values are updated if the fitness decreases with respect to the previous iteration, replacing the old values with the new best X_i. Thus, the algorithm at each iteration evaluates the traffic conditions and updates the control trying to converge to a minimum queue-length value.

However, there are some drawbacks related to this classic PSO configuration, given that the traffic conditions (lengths and flows) change constantly over time. The problem lies in the updating criteria for $pBest$ and $gBest$, values that only change if the fitness at time k is less than that the one obtained in $k-1$. If the conditions of the route worsen in consecutive cycles, $pBest_i$ and $gBest_i$ do not change, causing the system to stagnate in an unfavorable allocation (i.e., a good road condition is assumed when in reality it is worse). To solve this condition, we propose to modify the algorithm by resetting the values of $pBest_i$ and $gBest$ more frequently. For this purpose, $pBest_i$ is updated if the value of the fitness has not improved during two consecutive cycles. If the new fitness does not provide a value better than the stored $pBest_i$ in these two intervals, this parameter is replaced by the current X_i. Regarding $gBest$, this parameter is updated by the minimum $pBest$ of all the population when $gBest$ achieve a value less than a 10% of the saturation level. In addition, it is worth noting that the value of c_2 is considerably less than c_1, since the problem prioritizes the value of the best personnel over the best global.

1) Fixed-Cycle PSO: In a fixed cycle time, the summation of all phases remains constant, i.e., $\sum_{i=1}^{\varphi} X_i = C$. However, in PSO each X_i is the result of an evolutionary process, so in general, the restriction is not fulfilled. To satisfy the constraint, at the end of each iteration, the positions of the individuals are adjusted by calculating the difference between the cycle C and $\sum_{i=1}^{\varphi} X_i \neq C$. Then, the values of X_i are proportionally to obtain the required fixed cycle.

2) Variable-Cycle PSO: To allow the cycle length to change according to the road conditions and provide more flexibility in the optimization of green times, we propose that the sum of the phases may vary within a range $C - \frac{a}{2} < \sum_{i=1}^{\varphi} x_i < C + \frac{a}{2}$, where a is a positive tuning constant. If the limits are exceeded, the surplus (or scarce) time is cut (or added) to each individual proportionally. Consequently, the algorithm determines an optimal cycle time for the current system situation.

C. Replicator Dynamics (RD)

The RD model describes how certain characteristics of a population can evolve through the mutual interaction of individuals and the evaluation of their welfare state [16]. In analogy, urban traffic can be seen as a population system, since each vehicle has an individual behavior and all of them should be organized pursuing certain objectives (e.g., taking the shortest path or getting the lower fuel consumption). In these scenarios, the population is characterized by presenting replicable behaviors among individuals in search of common goals.

To describe the RD strategy, we assume that there is a set of n habitats in a natural environment, where a large number of individuals of a species must be distributed. In this case, $p_i(t) \geq 0$ is the number of individuals that choose the habitat i, while $p = [p_1, ..., p_n]^\top$ is the population state, and P_{tot} is the total population in the environment. The changes in each population define the RD model given by

$$\frac{dp_i}{dt} = p_i(f_i(p) - \bar{f}(p)), \forall i = 1, \dots, n, \tag{6}$$

where $f_i(p)$ represents the fitness function perceived in the i^{th} habitat and $\bar{f}(p)$ is the weighted average of the fitness functions,

$$\bar{f}(p) = \frac{1}{P_{tot}} \sum_{i=1}^{n} p_i f_i(p). \tag{7}$$

An interesting consequence of the definition in (7), is that the total population is kept constant, i.e., $\sum_{i=1}^{n} p_i = P_{tot}$ for all the time. [17].

In addition, observing (6), the steady state is achieved when $p_i^*(f_i(p^*) - \bar{f}(p^*)) = 0$, where $p^* = [p_1^* \ p_2^* \ ... \ p_n^*]^\top$ is an equilibrium point. If $p_i^* > 0 \ \forall \ i$, the condition $f_i(p^*) = \bar{f}^*$ is satisfied, where $\bar{f}^*$ is the average fitness in equilibrium. Then, at steady state, all habitats have the same welfare. Appropriate analogies with the population schemes have shown that certain choices of fitness functions lead to find a stable equilibrium point that solves various engineering optimization and control problems [18].

1) Fixed-Cycle RD: Assuming that the total number of habitats n represents the number of phases φ of the traffic network, the number of individuals in each habitat p_i represents the time in green for the phase i. Thus, the constant population condition is $\sum_{i=1}^{\varphi} p_i = P_{tot} = C$, where C is the cycle time, and the fixed-cycle restriction is respected by the RD for all the time. Now, an appropriate fitness function for the traffic system is given by

$$f_i(p) = \frac{w_1 q_i + w_2 Q_i}{w_1 S_{q_i} + w_2 S_{Q_i}}, \tag{8}$$

where, for the time cycle C, q_i is the vehicle flow, Q_i is the queue length, and S_{q_i} and S_{Q_i} are the saturation values of flow and queue for the phase i, respectively. Moreover, w_1 and w_2 are adjustable weights to determine the influence of flows or queues, taking into account that $w_1 + w_2 = 1$.

The fitness function is determined by the queue length and the vehicular flow in a proportional relationship given that larger fitnesses require longer green times. The combination of these two measures in the fitness definition allows the RD to take advantage of each one. For instance, estimating the vehicle flow requires to count the number of vehicles entering the network in the next cycle. On the other hand, the queue length estimation only takes into account the current status of the network and has slow changes compared to the flow when the system is saturated. In these cases, the rate of vehicles entering the network is very low and the calculation of the fitness only with the flow would estimate that the network is not saturated. This is a wrong indication and it is solved by adding the queue-length control, which is able to identify the clogging and can give priority to the controlled street. In this way, these two parameters together can be adapted to different states of the transit network. Finally, the weight parameters w_1 and w_2 are adjusted to determine the priority of the variables. In this case, $w_1 > w_2$, offering better performance under normal traffic conditions.

2) Variable-Cycle RD: To adjust the replicator dynamics to a variable cycle, the restriction $\sum_{i=1}^{\varphi} p_i = C$ must be relaxed to allow $\sum_{i=1}^{\varphi} p_i < C$. In this case, the time is a resource with an upper bound to be distributed in each phase. If one habitat is added to the environment in (6), we have $\varphi + 1$ equations and a new phase in the traffic system, obtaining $\sum_{i=1}^{\varphi+1} p_i = C$. The additional phase is a "fictitious agent" to [16], which is not assigned to any semaphore but acts as a surplus variable storing the time that is not used in the real phases. The fitness function for the fictitious phase is defined by

$$f_{\varphi+1}(p) = \frac{w_1 \sum_{i=1}^{\varphi} q_i + w_2 \sum_{i=1}^{\varphi} Q_i}{w_1 \sum_{i=1}^{\varphi} S_{q_i} + w_2 \sum_{i=1}^{\varphi} S_{Q_i}}, \tag{9}$$

which takes into account the fitness functions of all the real phases by monitoring the complete state of the system. The average fitness function must consider the new phase in the form

$$\bar{f}(p) = \frac{1}{P_{tot}} \sum_{i=1}^{\varphi+1} p_i f_i(p). \tag{10}$$

With the addition of the new phase, all fitnesses at equilibrium are still the same (including the fictitious one), maintaining the stability and optimal properties of the equilibrium point [16]. Furthermore, the control with an upper limit for the cycle time of the intersections can be better adapted to the dynamic conditions of a real traffic flow, without changing the structure of the controller.

III. Simulation Environment

In this work, we propose a co-simulation environment similar to the one in [11], with the plant in VISSIM as the specialized traffic simulator, and Matlab in charge of processing the information and sending the control signals back to the system.

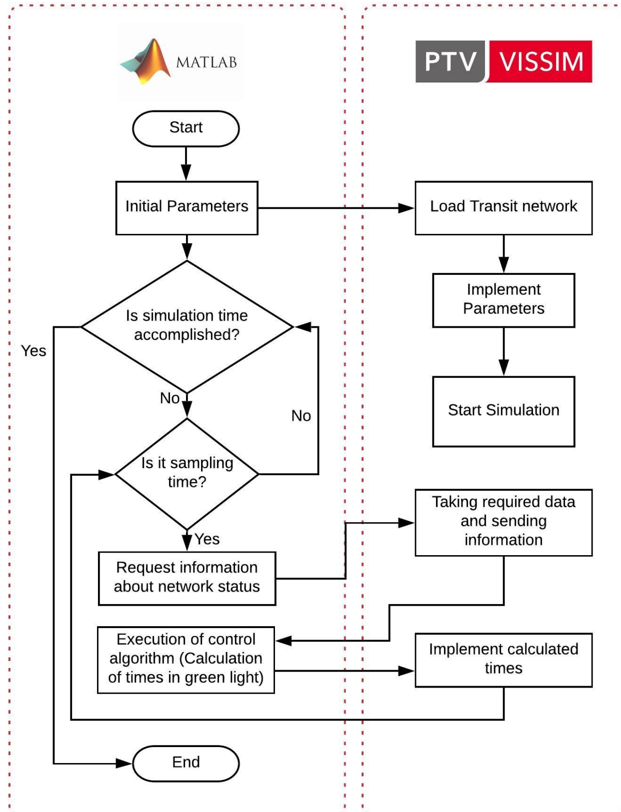

Fig. 1. Flow diagram describing the operation of the co-simulation strategy between Matlab and VISSIM for traffic control.

To communicate the two platforms, we use the VISSIM-COM interface, where the functions and parameters of the simulator can be manipulated by programming commands in different languages (handling COM objects), such as C++, Visual Basic, Java, Python, and Matlab, among others. With this interface, the user can manipulate directly the attributes of the internal objects of the traffic simulation. The program in Matlab uses the COM port to import and export information, while the implementation of control algorithms is eased by the well-known control and optimization toolboxes.

Figure 1 shows a general scheme of the proposed co-simulation operation. First, the initial parameters are established for the traffic system (simulation time, traffic network to be used, vehicular inflow, and seed of randomness, among others) that define the work scenario. These parameters are set in Matlab and then, sent to VISSIM to initiate the simulation. Within the Matlab environment, the algorithm waits until the time for data sampling is completed, requesting the relevant information from the traffic simulator (e.g., queue length, flows, delays, and fuel consumption). With this information, the control strategy calculates the green time for each traffic light, and the control variables are fed back and implemented in VISSIM, repeating the sequence continuously until the simulation time is reached.

IV. Simulation Results and Analysis

Next results are obtained in a network of two connected intersections (inspired by a real street network in the city of Pasto, Colombia), analyzing the behavior of the algorithms in cases where the output flow of an intersection becomes the inflow to the other. The analysis of the connection effects of several lane intersections is important since it represents most of the real traffic systems. In these cases, the influence of the road capacity on the system's performance is not relevant if an inefficient control strategy is implemented.

The study case is composed by two connected intersections with 8 traffic lights, as it is shown in Figure 2. It is worth noting that the intersections share vehicular flow, so the performance of the system depends on an implicit coordination between the controllers of each semaphore. The simulation time is 18000 seconds (5 hours), with a sampling time of $T = 115$ s, which corresponds to the optimal cycle calculated by the Webster's technique. The roads shown in Figure 2 have similar geometric characteristics.

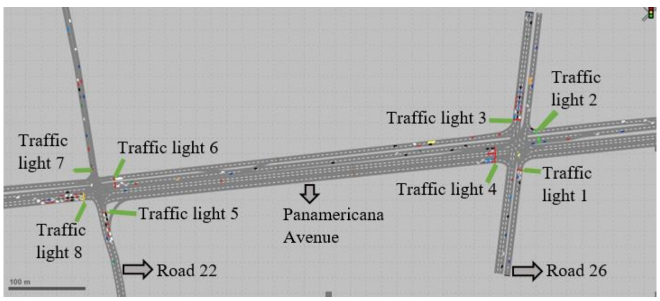

Fig. 2. Street network of two intersections with 8 traffic lights inspired by the system made up of Panamericana Avenue between streets 22 and 26, Pasto, Colombia.

The results for the accumulated average queue length obtained with the proposed algorithms are shown in Figure 3. The data show that the adaptive algorithms have a better performance against the technique of fixed time. In particular, the reductions in queue lengths for PSO with fixed and variable cycle are about 31% and 39%, respectively. The results for RD are even better, obtaining an improvement of 36% and 43% at the end of the simulation (for fixed and variable cycles). This results also imply a decrease in fuel consumption and waiting time for the vehicles, results that are shown in Table I. The indices presented in the table are the averages for mean maximum queue lengths in meters (Qlen and QlenMax, respectively), the number of vehicles passing through the intersections (VehsAll), the delay for each vehicle measured in seconds (VehsDelayAll), the stop delays measured in seconds (StopDelayAll), the number of stops (StopAll), the emissions of carbon monoxide, nitrogen oxide, and volatile organic compounds measured in grams (EmissionCO, EmissionNOX and EmissionVOC), and the fuel consuption measured in gallons (FuelConsumption). It is remarkable that all the indices have a similar behavior, with the best results for the RD algorithm followed by the PSO, both with variable cycle.

TABLE I
Accumulated average performance indices for the fixed-time simulations (Webster), PSO, and RD with fixed and variable cycles (subscripts FC and VC, respectively).

	Qlen	**QlenMax**	**VehsAll**	**VehsDelayAll**	**StopDelayAll**
PSO_{FC}	8081.5	35424.8	36055.5	14495.1	11873.7
PSO_{VC}	7160.1	33028.5	36051.5	12943.3	10400.7
RD_{FC}	7410	30370.8	36061.5	13554.8	10986.4
RD_{VC}	6147	26513.9	36092.5	11378.3	8846.3
$Webster$	11718	53501.1	36590	13852.8	11398.3
	StopsAll	**EmissionCO**	**EmissionNOX**	**EmissionVOC**	**FuelConsumption**
PSO_{FC}	283.7	48626	9460.9	11269.5	695.6
PSO_{VC}	273.9	45525.3	8857.5	10550.9	651.2
RD_{FC}	276.6	46667.3	9079.8	10815.6	667.6
RD_{VC}	272.9	42933.4	8353.3	9950.2	614.2
$Webster$	269.6	47267.2	9196.5	10954.6	676.2

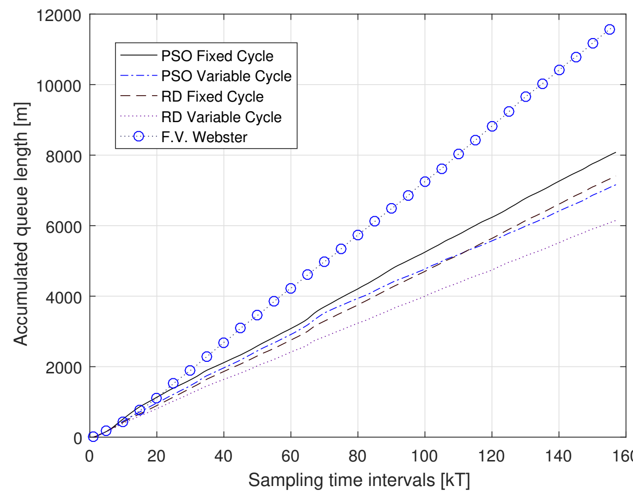

Fig. 3. Accumulated queue length in time for the network of two intersections.

Based on the results, we can observe that the control with fixed cycle (in PSO and RD) is restrictive, since the system does not react appropriately to high saturation conditions of the streets. In other words, regardless of the system's conditions, the entire time resource must be assigned, even though some of the intersections are not occupied and some others are very saturated. On the other hand, in the strategies with variable cycles the total cycle can be reduced depending on the saturation, meaning that all phases can last just a few to allow a faster evolution in the traffic lights, and giving right-of-way to small flows, more frequently, at each saturated lane.

Furthermore, for a variable cycle, a delay in the green-time values of the consecutive traffic lights is observed in some simulation instants, generating a phase coordination so-called green wave [19]. This coordination is related to the distance between intersections and the average speed of the vehicles. Thus, when a vehicle leaves the first intersection and arrives at the next one, it is highly probable that it will also pass. If a vehicle goes into a green wave, it will have a route with a minimum delay and possibly without stops, decreasing fuel consumption and gas emissions, among other performance indices. It is noteworthy that although the proposed algorithms carry out an independent control at each intersection (without sharing information), it is possible to notice a coordination of phases in the simulations. This is observed by an improvement of 3% in the queue length with replicator dynamics for the network of two intersections compared to tests performed with a single intersection, corresponding to the one on the right side of the Figure 2. realree

V. Conclusions

The proposed adaptation for the PSO and RD techniques is appropriate for the control of traffic systems, taking into account the characteristics of the streets and sensor information for the feedback. In addition, the co-simulation scheme allows the system to emulate real-implementation conditions with flows adapted to actual road characteristics provided by a specialized traffic platform. Moreover, using the Matlab toolboxes facilitates the implementation of advanced controllers. As future work, we propose to carry out tests to check synchronization, study algorithms with shared information between intersections, and analyze networks with a large number of traffic lights.

Acknowledgments

We would like to thank Pablo Ñañez, Iván Gutierrez for their important inputs and discussions around the VISSIM API. We also thank to Gilberto Rueda from PTV Mexico for providing us a free student version of VISSIM for the tests in this work.

References

[1] J. Morales Pérez and J. A. Fernandez Gomez, "Análisis discriminante de algunas variables que influyen en la contaminación acústica debida al tráfico urbano en una gran ciudad," *Revista Ingenierías Universidad de Medellín*, vol. 11, no. 21, pp. 13–22, 2012.

[2] M. B. Santamaría and M. R. Moscol, "Semáforos inteligentes para la regulación del tráfico vehicular," *Ingeniería: Ciencia, Tecnología e Innovación*, vol. 1, no. 1, pp. 37–37, 2014.

[3] L. F. Pedraza, C. A. Hernández, and D. A. López, "Control de tráfico vehicular usando anfis," *Ingeniare. Revista chilena de ingeniería*, vol. 20, no. 1, pp. 79–88, 2012.

[4] Á. Acuna, E. Arreche, and S. Nesmachnow, "Planificación de tráfico y transporte colectivo en el corredor garzón utilizando un algoritmo evolutivo," *Universidad de la República, Montevideo*, 2015.

[5] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter, "Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 813–827, 2018.

[6] B. HomChaudhuri, A. Vahidi, and P. Pisu, "Fast model predictive control-based fuel efficient control strategy for a group of connected vehicles in urban road conditions," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 760–767, 2017.

[7] J. C. Peñaranda, C. E. Amado, S. S. Mora, S. C. Casadiego, B. M. Delgado, and J. R. Mateus, "Control adaptativo para optimizar una intersección semafórica basado en un sistema embebido," *Ingeniería y ciencia*, vol. 12, no. 24, pp. 169–193, 2016.

[8] A. Jovanović, M. Nikolić, and D. Teodorović, "Area-wide urban traffic control: A bee colony optimization approach," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 329–350, 2017.

[9] Y.-S. Qian and C.-L. Wang, "Urban traffic signal control based on three swarms cooperative-particle swarm optimization algorithm." *Computer Engineering and Applications*, vol. 43, no. 14, pp. 187–189, 2007.

[10] M. H. Khooban, N. Vafamand, A. Liaghat, and T. Dragicevic, "An optimal general type-2 fuzzy controller for urban traffic network," *ISA transactions*, vol. 66, pp. 335–343, 2017.

[11] T. Tettamanti, A. Csikós, I. Varga, and A. Eleőd, "Iterative calibration of vissim simulator based on genetic algorithm," *Acta Technica Jaurinensis*, vol. 8, no. 2, pp. 145–152, 2015.

[12] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[13] D. Robles, P. Ñañez, and N. Quijano, "Control y simulación de tráfico urbano en colombia: Estado del arte," *Revista de ingeniería*, no. 29, pp. 59–69, 2009.

[14] F. Webster, "Traffic signal settings, road research technical paper No. 39," *Road Research Laboratory*, 1958.

[15] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence*. Elsevier, 2001.

[16] A. Pantoja, G. Obando, and N. Quijano, "Distributed optimization with information-constrained population dynamics," *Journal of the Franklin Institute*, vol. 356, no. 1, pp. 209–236, 2019.

[17] J. Hofbauer and K. Sigmund, *Evolutionary games and population dynamics*. Cambridge university press, 1998.

[18] N. Quijano, C. Ocampo-Martinez, J. Barreiro-Gomez, G. Obando, A. Pantoja, and E. Mojica-Nava, "The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory," *IEEE Control Systems*, vol. 37, no. 1, pp. 70–97, 2017.

[19] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Evaluation of green wave policy in real-time railway traffic management," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 607–616, 2009.