




Article

Micro Artificial Immune System for Traffic Light Control

Raul Galvan-Correa ¹, Mauricio Olguin-Carbajal ^{1,*}, Juan Carlos Herrera-Lozada ¹,
Jacobó Sandoval-Gutierrez ² , José Félix Serrano-Talamantes ¹ , Rodrigo Cadena-Martinez ³ 
and Carlos Aquino-Ruiz ¹

¹ Instituto Politécnico Nacional, CIDETEC, Postgraduate Department, Av. Luis Enrique Erro S/N, Gustavo A. Madero, Unidad Profesional Adolfo López Mateos, Ciudad de México D.F. 07738, Mexico; raulgalvan92@outlook.com (R.G.-C.); jlozada@ipn.mx (J.C.H.-L.); jfserranotal@gmail.com (J.F.S.-T.); caquino@ipn.mx (C.A.-R.)

² Departamento de Procesos Productivos, Universidad Autónoma Metropolitana Unidad Lerma, Lerma de Villada, Edo. de México CP 52005, México; j.sandoval@correo.ler.uam.mx

³ Universidad Tecnológica de México, UNITEC, Calle Nte. 67 23460, San Salvador Xochimanca, Azcapotzalco, Ciudad de México 02870, México; rocadmar@mail.unitec.mx

* Correspondence: molguinc@ipn.mx

Received: 29 September 2020; Accepted: 4 November 2020; Published: 9 November 2020



Featured Application: This work can be used to reduce the emission of pollutants expelled by motor vehicles in specific areas of a city.

Abstract: A new bio-inspired meta-heuristic, called the micro artificial immune system (MAIS), has been developed in order to reduce the rates of pollution for a specific region of Mexico City through the optimization of vehicular flow. Simulation of urban mobility (SUMO) was used to simulate the effects of the programming of the traffic lights obtained by the MAIS. Currently, pollution and travel times from one place to another are increasing due to the number of inhabitants that live in big cities, which has generated a decrease in people's quality of life. Hence, we propose the optimization of the programming of the sequences of traffic lights through this bio-inspired meta-heuristic. The obtained results show that the MAIS outperforms most of the algorithms tested in this research.

Keywords: traffic light; control; micro-algorithms; artificial immune system; mobility

1. Introduction

Vehicular traffic has grown exponentially, in an excessive way, in all the big cities of the world. Developing strategies to control or at least reduce the problems associated with vehicular traffic in large cities mainly involves solutions in two areas: the modification of infrastructure and the topology of the city, but in many cases, this is not feasible. Hence, researchers have developed other strategies to deal with the problem of vehicular traffic. One of these is the control of traffic lights. It has been found that their proper programming is able to reduce congestion, improving vehicular traffic and reducing travel times as well as the levels of pollutants emitted. Properly configuring the synchronization of traffic light cycles to optimize vehicle flow is a problem with a non-polynomial computational complexity. Mexico City has a unique geographical location as well as very particular orographic conditions, such as its population of 20 million, its elevation of 2000 m above sea level, and the fact that it lies in a valley surrounded by volcanoes, so there is only a limited possibility of modifying the infrastructure of roads, streets, and avenues. The proper use of synchronization in traffic lights is a viable option to support the reduction of pollutant emissions throughout the city. Therefore, it

is necessary to use computational tools that can find good solutions to the problem in a reasonable time. Bio-inspired computation has shown that it is capable of solving difficult problems in real-world applications, particularly for programming traffic light cycles, see [1–6].

In recent years, efforts have been made to seek the correct management and optimization of traffic in an automatic way without the need for human interaction. For example, the adaptive control system proposed by McKenney [7] has better results than the fixed signal control that is traditionally used. In Sanchez et al. [8] a three-element architecture was used: a genetic algorithm (GA) as a non-deterministic optimization technique, cellular automata (CA) based on the traffic simulation performed within the GA routine, and a Beowulf cluster. The simulation consists of creating many vehicles that enter and leave the network. During the process, some of them arrive at their destination and leave the network. The number of vehicles arriving at their destination is an indicator of the effectiveness of the simulation and serves as a point of comparison for another cycle of particles. A microscopic simulator is used, where the traffic is considered as a discrete collection of particles following certain rules for their interaction. Subsequently, we sought to develop a model that had the ability to learn and adapt actively. Through a neural network model, it is possible to perform this task. Nagare et al. [9] used a back propagation neural network (BPNN), which analyzes the positive and negative signals of the propagation indefinitely until the error output is reduced. However, the prediction of the algorithm becomes ineffective when there are complex crossroads. In the cities of Bahia Blanca, Argentina, and Malaga, Spain, Olivera et al. [10], carried out tests with hundreds of traffic lights to optimize traffic light cycle programs through particle swarm optimization (PSO). They used a mono-objective function based on polluting emissions to control the timing of the traffic light signals although they did not take into account the vehicular flow for sections of heavy traffic in the cities. Garcia-Nieto et al. [4] used a simulation of urban mobility (SUMO) to model the behavior of vehicular flow and traffic light changes, for Malaga and Seville in Spain. As a proposal to address the vehicular flow and the emission of pollutants, Ji proposed a system establishing a complete emission model and also using a microscopic emission simulation platform to estimate the reduction of pollutant emissions for the city of Changzhou in China [11]. A similar task was done by Damay to reduce emissions and improve vehicular flow using a multi-objective genetic algorithm and a microscopic traffic simulator optimizing the duration of the green lights in the traffic lights for a real traffic network in the city of Rouen in France [12].

Ferrer et al. [13] proposed a validation of scenarios by checking the robustness of the traffic programs. Tests were carried out in Malaga, Spain and a group of programs generated by four optimization algorithms were validated: PSO for traffic lights, differential evolution for traffic lights, random search and the SUMO cycle program generator. The tests were carried out taking into account the following factors. (1) Climate: the types of weather considered were rainy, thunderstorm, sunny, windy and cloudy. Weights were assigned to each of them taking into account the percentage of the days of the year that have these conditions. (2) Number of vehicles: the number of vehicles was established as a little or a lot. (3) Driver imperfection: this characteristic represents poor driving quality (low, medium or high). In many traffic situations, the driver makes a difference, which is why three levels of driving were defined. (4) Reaction time of the driver: time interval needed by the driver to perform an action. Three levels were considered (low, medium and fast). (5) Type of vehicle: two types of vehicles were taken into account (light and heavy). The weights were defined according to the information about the vehicles that operate in the city.

Perez used a multi objective evolutionary algorithm (MOEA) for the optimization of traffic signal control in the city of Montevideo in Uruguay for the programming of intelligent traffic lights with good results [14]. MOEA algorithms were combined with a microscopic traffic model to simulate the vehicular flow while simultaneously reducing pollutant emissions. Another approach is the one used by Leal et al. [15], applying a predictive model with time delay used in conjunction with differential evolution techniques and GAs in an urban simulator reproduction of a city in Belo

Horizonte, Brazil. They achieved a 47% reduction of the delay time of the vehicles with respect to the normal programming of the traffic systems.

A study by Vidal and Olvera [16] of two neighborhoods of cities in South America, Jacinto Vera in Montevideo and the micro-center Posadas in Argentina (areas with very different topologies), performed tests of the programming of traffic light cycles with the objective of minimizing the waiting time of the vehicles and maximizing their speed. As a result, traffic flow was improved, reducing fuel consumption and therefore the emission of pollutants. They tested five metaheuristics and concluded that the one that produced the best results was differential evolution.

The present paper proposes a method, referred to as the micro artificial immune system algorithm (MAIS), for the optimization of traffic cycle programming based on an artificial immune system micro-algorithm, and analyzes it on a real urban topology in a very important avenue of Mexico City (Avenida Insurgentes) using meta-heuristic techniques combined with simulation with the objective of improving the vehicular flow and reducing the emission of pollutants into the environment. The objective is to minimize the waiting time of the vehicles flowing in the network and maximize their speed. The results obtained improve the traffic flow by reducing waiting times and increasing the speed of the vehicles, favoring a decrease in fuel consumption and environmental pollution. The MAIS demonstrates that it improves not only the flow of the vehicles but also reduces pollution and fuel consumption.

Our objective is to test MAIS as an option for the efficient programming of traffic lights, and verify whether it is equivalent to other similar algorithms and if it has some advantages. The MAIS algorithm was designed and configured in a unique way not used before by an artificial immune system with reduced population.

The main contributions of this paper are as follows.

1. The proposed MAIS is the first to be used in a practical problem of this kind.
2. MAIS and PSO provide the best results, better than the other tested algorithms, namely, simulated annealing (SA), the genetic algorithm (GA), and differential evolution (DE).
3. The behavior of the MAIS is tested.
4. The reason for the improved performance of the MAIS is its reduced population and the periodical replacement of the antibodies to introduce diversity.

The rest of this paper is organized as follows: after the Introduction, Section 2 presents the microscopic simulator and introduces the artificial immune system. Section 3 presents the algorithms that are the basis of the MAIS. The experimental setup and the results are presented in Section 4. Lastly, the conclusions are given in Section 5.

2. Background

2.1. Simulation of Urban Mobility (SUMO)

SUMO is a traffic simulator that has been free since 2001. Its main feature is the modeling of traffic systems, specifically road vehicles, public transport, and pedestrians. SUMO has a large number of support tools that are responsible for finding transport routes, vehicle traffic visualization, and calculation of pollutant emissions. One of the main advantages of the software being open source is that it offers the opportunity to implement any algorithm to determine the programming of the traffic lights. SUMO is not software only for traffic simulation, it also offers tools that allow analyzing the performance of the programming implemented in the road network. As an essential part for the simulation of SUMO, it is necessary to represent the road network through the “netgen” application, where each of the network crossings is defined. On the other hand, it is possible to import a map of a road network that is digital from an original map view (see Figure 1a) and the SUMO network (see Figure 1b).

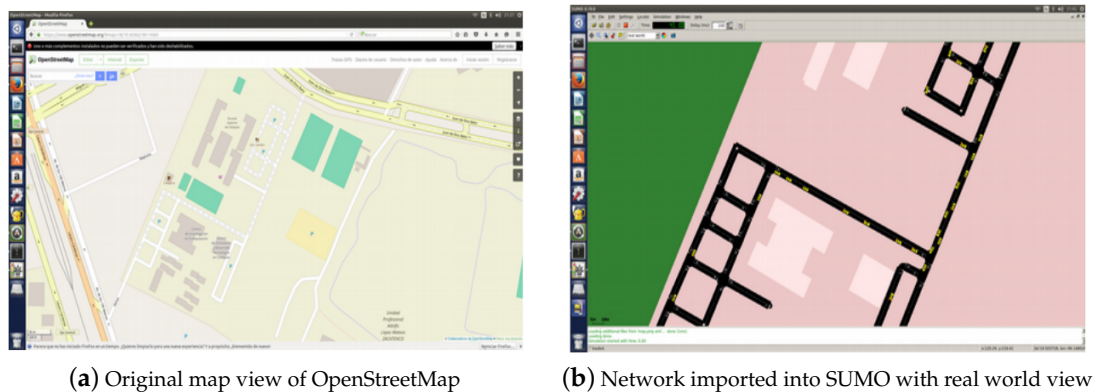


Figure 1. Example of converting a map in OpenStreetMap.

SUMO performs the simulation microscopically, which means that the movement of the vehicles can be observed graphically, as well as the starting times and the paths they can take within the network. All the characteristics of the vehicles, such as their speed and position, can be defined. As relevant information, SUMO offers information at the end of simulation of the vehicles within the road network. For example, if you want to obtain the global travel information, you must use the *tripinfo-output* command. Additionally, SUMO allows several outputs to be generated for each of the simulations, such as the amount of sound emitted, the pollutants generated, and fuel consumption, among others. SUMO allows the modification of the cycles of the traffic lights, the observation of intersections, roads, directions, and vehicles moving along their routes.

2.2. Artificial Immune System

Nunes de Castro and Von Zuben proposed and developed the clonal selection algorithm (CLONALG) on the basis of the clonal selection theory of the immune system [17,18]. Human B-cells and T-cells adapt in order to match and kill foreign cells. Clonal selection is based on the way in which both of these types of cells of the immune system adapt. The algorithm is described as follows:

1. Generate randomly a set P of antibodies (candidate solutions), composed of the memory cells M and the remaining population P_r , where $P = P_r + M$;
2. Select the n best antibodies P_n , based on affinity;
3. Clone these n best antibodies in proportion to their affinity using

$$N_c = \sum_{i=1}^n \text{round}\left(\frac{\beta N}{i}\right) \quad (1)$$

where N_c is the total number of clones generated for each of the antigens, like an objective function, β is a multiplying factor, where n is the total number of antibodies, and *round* is the operator that rounds its argument toward the closest integer. For $n = 100$ and $\beta = 1$, the antibody with highest affinity will produce 100 clones; the antibody with the second highest affinity produces 50 clones, and so on, giving rise to a temporary set of clones C ;

4. Apply a hyper mutation to C . The degree of mutation is inversely proportional to the affinity. The mutated antibodies generate the set C^* ;
5. Re-select the best elements from C^* to compose the memory set M . Some members of P can be replaced by other improved members of C^* ;
6. Replace d antibodies with new ones to introduce the concept of diversity. The probability of being replaced is inversely proportional to the affinity of the previous remaining population P_r .

3. Design

The fitness function is a minimization function, and we take the fitness function designed by Ferrer et al. (see [19]), but with some changes for our particular case. The parameters to minimize of our model within the optimization of vehicle flows are as follows:

- Emission of pollutants in the environment (Epe).
- Waiting times in the vehicular flow (W).
- The transit of one or more vehicles (V).

The decision variables used by the proposed model are continuous variables, since it yields decimal values during the entire simulation period.

$$F(s) = \frac{(V_{rem}(P) \times t_{rem}) + (\sum_{v=0}^{V(P)} t_v(P) + W_v(P) + (Epe_v(P))^2)}{V(P) + GR(P)}, \tag{2}$$

where

- $V_{rem}(P)$ refers to the vehicles with incomplete journeys and is proportional to the simulation time t_{rem} . This is used to penalize the incomplete journeys during the simulation process.
- $t_v(P)$ is the travel time of a vehicle and v is the number of simulation steps.
- $W_v(P)$ is the process that involves the number of vehicles that arrive at their destination (v).
- $Epe_v(P)$ refers to the emission of polluting waste during the entire simulation process. Since our main objective is the decrease in the emission of pollutants, we square the variable $Epe_v(P)$ so that this parameter is prioritized.
- $V(P)$ refers to the fact that we only consider cars that make complete tours and that arrive at their destination throughout the simulation process. Vehicles with incomplete routes and that take a long time to move are those that emit polluting waste.
- The C_r term (see [19]) is calculated with the following equation:

$$GR(P) = \sum_{i=0}^n \sum_{j=1}^{I_i} \varphi_{ij} \times \frac{G_{ij}}{R_{ij}}, \tag{3}$$

where G_{ij} is the number of green traffic lights and R_{ij} is the number of red traffic lights. φ_{ij} is the duration time of the sequence to change from one state to another, i is the number of traffic lights in general, and it must always be $i = 1$, to avoid dividing by 0.

The equation refers to the state of the traffic lights at each instant of time in which each vehicle must wait at a red light or continue its journey on a green light. A prolonged red light state causes road chaos at the intersection crossing.

To reduce the computational cost in the solution of population algorithms, the use of micro-algorithms is proposed, whose main characteristics are a drastically reduced population and a nominal convergence. The nature of micro-algorithms is different from the standard, so the intention is to make their performance acceptable in terms of their results and computational time. For the same reason, in this work we present the micro artificial immune system algorithm (MAIS) as an alternative to solve the problem of traffic light control and it is not the intention, at least for the moment, to compare it with other mathematical methods.

Our MAIS algorithm is based on the method proposed by Goldberg in [20], who presented experiments with different sizes of populations for GAs and showed the relation between the size and the error levels. Later, Krishnakumar designed a binary representation micro-genetic algorithm with a population of only 5 individuals [21]. By comparing the performance of this micro-AG against a simple GA of 50 individuals, Krishnakumar obtained better results on single objective functions.

Additionally he verified that the micro-GA converged faster than the simple GA. Krishnakumar reported better and faster results for the micro-GA when tested with two static functions and two control problems of the real world. Rahnamayan and Tizhoosh [22] developed a micro-DE and a micro-ODE for image thresholding, where the micro-ODE outperformed the micro-DE as well as a traditional Kittler thresholding method. Micro-algorithms have been used and have demonstrated advantages in other hard-to-optimize problems, as in [23–25].

The MAIS proposes applying variation operators to a small population (randomly generated) to achieve first-phase convergence. Subsequently, a new population should be generated by transferring the best individuals of the population obtained after the convergence to the new one. The remaining individuals are randomly generated.

This MAIS was specifically developed for the optimization of the traffic light program cycles (traffic lights).

In Algorithm 1, *ExtGen* is the current generation and *Max Generations* is the stop criterion for the MAIS external loop. *Generation* is the current generation and *NominalConv* is the stop criterion for the MAIS inner loop. *F* is the initial population, *S* is the selected population, and *C* is the cloned population. In addition, *M* denotes the cloned mutated population, *R* is the ranked population, and *E* is for the self regulated population.

Algorithm 1 MAIS—Micro Artificial Immune System

```

1: function MAIS( $F(x)$ )
2:    $F \leftarrow$  Start a population of five antibodies()
3:   for ExtGen  $\leftarrow$  1 to ExtGen < MaxGenerations do
4:     for Generation  $\leftarrow$  1 to Generation < NominalConv do
5:        $S \leftarrow$  Select the  $n$  best antibodies based on ranking ( $F$ )
6:        $C \leftarrow$  Clone all the antibodies ( $S$ )
7:        $M \leftarrow$  Cloning mutation( $C$ )
8:        $R \leftarrow$  Selection based on ranking ( $M$ )
9:        $E \leftarrow$  Replace  $d$  antibodies by novel ones ( $R$ )
10:       $F \leftarrow$  SelfRegulation by eliminating remaining clones( $E$ )
11:    end for
12:     $E \leftarrow$  Algorithm reset( $F$ )
13:    if Generation < NominalConv then
14:       $f \leftarrow$  Create 3 new random antibodies to maintain diversity( $E$ )
15:    end if
16:  end for
17: end function

```

The MAIS algorithm works as follows:

1. A population of five antibodies (individuals) is generated randomly (Algorithm 1, line 2). The first generation of the population will contain these five antibodies. It is established that ten generations will be sufficient to achieve nominal convergence. The stop criterion (**MaxGenerations** for the MAIS external loop) is given as an input parameter when the algorithm is used.
2. A selection based on ranking with respect to affinity is used (Algorithm 1, line 5), where each of the antibodies is evaluated using the objective function. In the MAIS terminology, it is called an antigen (*Ag*).
3. Subsequently, the cloning of all antibodies is performed (Algorithm 1, line 6), using the following equation:

$$N_c = \sum_{i=1}^n (n - (i - 1)), \quad (4)$$

where N_c indicates the number of clones that will be generated for each of the antibodies, n is the total number of antibodies in the population, and i is the current antibody, starting with the one with the highest affinity (*BestAb*). If a population of five antibodies is considered, a population of fifteen

clones will be generated: the best antibody will generate five antibodies, the second best antibody will be cloned four times, and so on, until only one clone is generated for the worst antibody.

4. *Cloning mutation* (Algorithm 1, line 7). The probability of mutation is set at the beginning of the nominal convergence for each group of clones obtained from the same antibody. This probability is proportional to the affinity of the antibody that generated the clones and will decrease in each of the generations. In this way, the clone group of the best antibody will be less likely to mutate than the other clone groups that were generated from the rest of the antibodies. The clone obtained from the *worst* antibody will have the greatest chance of *mutating*.

$$\text{ProbMutation}(i) = \frac{\text{Aff}(i)}{\sum_{i=1}^n \text{Aff}(i)} \tag{5}$$

Here, i is the antibody that will establish the probability of mutation for the group of clones that were generated from it and n is the total number of antibodies of the population. The following equation is used to uniformly decrease the probability of mutation in each of the generations.

$$\text{if prob} \leq \frac{\text{ProbMutation}(i)}{\text{Generation}}, \tag{6}$$

where, prob is randomly chosen $\in [0, 1]$ and Generation is the current generation within the nominal convergence. This value is divided by Generation because ProbMutation should not be divided by zero. There are two operators (for the algorithm section, that will affect the clones) that allow exploring the search space by performing modification steps of different sizes during the mutation process. The choice of which operator to apply to the solution vector is made with a probability of 0.5.

$$x' = x \pm \frac{\alpha \times \text{range} \times \text{Generation}}{N_c}, \tag{7}$$

or

$$x' = x \pm \frac{\alpha \times \text{range}}{\text{Generation} \times N_c}, \tag{8}$$

where x' is the mutated variable, x is the variable to mutate, α is a random number $\in [0, 1]$, g Generation is the current generation of the nominal convergence (inner loop), and N_c is the total number of clones. To ensure that the values are within the delimited ranges, the following instruction is defined.

$$\begin{aligned} &\text{if } x \leq 5 \text{ then } x \leftarrow 5 \\ &\text{if } x \geq 60 \text{ then } x \leftarrow 60 \end{aligned}$$

For the five clones obtained from **BestAb**, $\text{range} \in [LB, UB]$ is a random number between the lower limit and the upper limit of the values that the decision variable can take. For the rest of the clones, range is any decision variable of the **BestAb** antibody and the position within the vector dimension is obtained randomly.

5. *Selection based on ranking*. The 15 clones are sorted using ranking with respect to their affinity (Algorithm 1, line 8), with the best clone having the best affinity. If the nominal convergence has not been reached, the two best clones are selected and the new population is completed with another three clones randomly selected from the clone population (Algorithm 1, line 9).

The remaining clones will be eliminated, maintaining a population of five antibodies (Algorithm 1, line 10).

6. *Algorithm reset* (Algorithm 1, lines 13–14). If the nominal convergence has been reached (Generation = **NominalConv**), the best two clones will be kept and three other antibodies will be generated randomly to be able to complete the initial population and restart the nominal convergence (Algorithm 1, lines 4 and 11) until the external cycle meets the algorithm's stop condition.

4. Experiments and Results

The experimental setting was as follows. For each experiment, 700 runs were executed for each algorithm (micro artificial immune system (MAIS), simulated annealing (SA), genetic algorithm (GA), particle swarm optimization (PSO), and differential evolution (DE)) with a stop criterion of one million function evaluations (FEs). Our MAIS used a population of five individuals. For the other algorithms, the size of the population was forty. The settings used for the two functions were the same (see Table 1). The MAIS and all other algorithms were programmed in C and compiled by using g++ -O3.

Table 1. Algorithm setting.

Algorithm	Parameters	Value
Micro artificial immune system (MAIS)	Population	5
	Clone limit	15
	Selection	Ranking
	Mutation	Based on affinity
Simulated annealing (SA)	Population	1
	Freezing factor	0.995
	Disturb	Round [−5, 5]
	Disturb rate	10%
Genetic algorithm (GA)	Population	40
	Selection	Binary tournament
	Crossing type	SPX
	Crossing probability	100%
Particle swarm optimization (PSO)	Population	10
	Local coefficient	2.06
	Global coefficient	2.06
	Inertia	0.5
Differential Evolution (DE)	Population	40
	F	0.4
	CR	0.85

The hardware used was a heterogeneous cluster with 16 machines using Intel Core2 Quad processors Q9400 running at 2.66 GHz and 4 GB of memory. The cluster was administrated by a HTCondor 8.2.1. The total computing time was 200 hours of cluster time. The initial solutions for optimization were defined by the initial state of all the algorithms. The cluster it is located at México City, México, and is the 3rd cluster version.

Two areas of study were taken as reference to offer a proposal as close to reality as possible. The research was done on the *Avenida Revolucion* area and *Avenida de los Insurgentes* in Mexico City, see Figure 2a,b. The view of the area in OSM and SUMO is shown in Figure 3. For the simulation, 15 traffic lights were considered, with 66 phases in the changes of its lights, covering 0.66 km².



Figure 2. Study area, "Av. Insurgentes" Street in Mexico City.

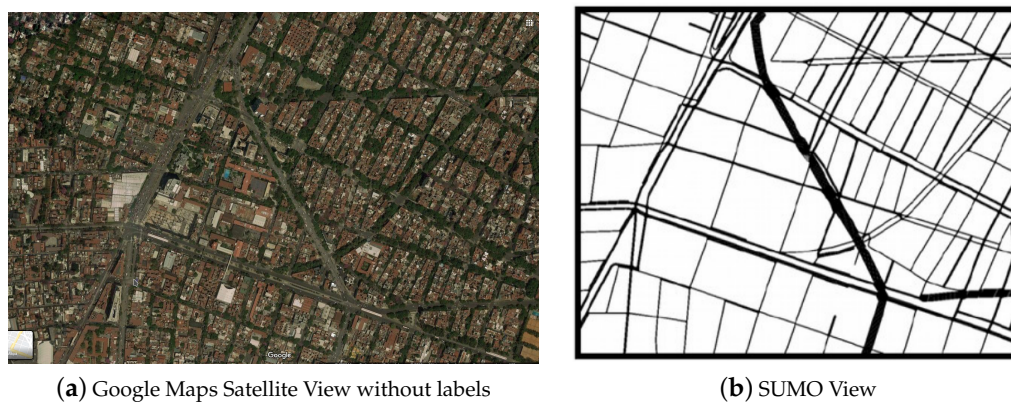


Figure 3. Detailed study area.

From the data obtained so far, it can be seen that the PSO (2.682) and MAIS (2.721) algorithms obtained better results, in terms of average fitness, than the rest, but PSO has a slightly bigger standard deviation, 0.060, compared to SA and MAIS (with 0.040).

However, it is important to note that the MAIS algorithm has the best average value after PSO, see Table 2, consuming less resources in carrying out operations, but with almost the same time when executing SUMO, and MAIS has a better standard deviation, meaning that it has less variation in the results, see Table 3.

Table 2. Average results for travel metrics for both experiments.

Algorithm	SA	GA	MAIS	PSO	DE
Average waiting (minutes)	1.6	2.2	1.3	1.4	1.7
Journey time (minutes)	10.6	11.2	10.3	10.4	10.7
Average distance traveled (kilometers)	3.58	3.85	3.52	3.50	3.61
Average speed (kilometers per minute)	0.337	0.343	0.341	0.336	0.337

Table 3. Average fitness value obtained by the algorithms for the first experiment set.

Number of Traffic Logics (NTL)	Number of Vehicles 100				
	SA	GA	MAIS	PSO	DE
66 phases	2.899	2.943	2.721	2.682	3.024
Standard dev.	0.040	0.0606	0.040	0.063	0.113

A second experiment was carried out, where the number of tests performed on each of the algorithms was increased. Likewise, the MAIS algorithm was modified to improve the mutation of the clones, because there were cases where very high values were assigned to the modification in the times of the traffic lights. There are two operators (for the algorithm section that will affect the clones) that allow exploring the search space by performing modification steps of different sizes during the mutation process. The choice of which of the operators to apply to the solution vector is made with a probability of 0.5.

4.1. The Kolmogorov–Smirnov test

Once the executions were carried out, the Kolmogorov–Smirnov test was applied to define whether there was a significant difference in the results obtained by the best two algorithms (MAIS and PSO), see Table 4. For this test, a positive hypothesis and a negative hypothesis are defined, with $\alpha = 0.05$.

- H0: The results of the fitness function in both algorithms are the same.
- H1: The results of the fitness function in both algorithms are unequal.

Table 4. Two sample Kolmogorov–Smirnov test.

Frequency	Algorithm	N
Fitness	1	30
	2	30
Total		60

Because the asymptotic significance is less than 0.05, H1 is proven, meaning that there are significant differences between the results of both algorithms. Since the H1 hypothesis is accepted, the Z-value is 2.711 concluding that the differences are large, see Table 5, so it is concluded that both algorithms have significantly different behaviors.

Table 5. Two independent sample Kolmogorov–Smirnov test. Statistics Grouping variable: Algorithm.

	Fitness	
Extreme Max. Differences	Absolute	0.700
	Positive	0.000
	Negative	−0.700
Kolmogorov–Smirnov Z	2.711	
Next Bilateral Asymptotic	0.000	

4.2. ANOVA Single Factor Test

An analysis of variance (ANOVA) will help to verify the hypothesis that the population means are equal. For this, the homogeneity test of variances is performed. Because the significance is greater than 0.05 (see Table 6) the assumption of variance homogeneity is met (see Table 6). The above means that the samples used have no significant differences, so no algorithm has an advantage in the test. Once the homogeneity assumption was validated, the single-factor ANOVA test was performed, see Table 7.

Table 6. Homogeneity test of variances.

Based on the		Levene			Sig.
		Statistics	gl1	gl2	
Fitness	Average	1.579	1	58.00	0.214
	Median	1.433	1	58.00	0.236
	Median and fit gl	1.433	1	57.49	0.236
	Cropped average	1.577	1	58.00	0.214

Table 7. ANOVA single factor test.

	Sum of Squares	gl	Quadratic Mean	F	Sig.
Between groups	0.201	1	0.210	61.672	0.000
Within groups	0.197	58	0.003		
Total	0.407	59			

5. Discussion

An ANOVA will help to verify the hypothesis that the population means are equal. For this, the homogeneity test of variances was performed. Because the significance was greater than 0.05, the assumption of variance homogeneity was met. Once the homogeneity assumption was validated, the single-factor ANOVA test was performed. The unifactorial ANOVA indicates that there are differences in the results of the fitness function of the algorithms, because the value of significance is less than 0.05.

This means that although both algorithms have seemingly similar performances, they are in fact algorithms with statistically different behaviors and therefore can be used with different results, see Tables 8 and 9. In this case, the proposed algorithm (MAIS) demonstrates effective behavior, is efficient and robust, and can be used in the optimization of vehicular flow through the control of lights, and with a slightly smaller standard deviation than the PSO algorithm.

Table 8. Average fitness value obtained by the algorithms.

Number of Traffic Logics (NTL)	Number of Vehicles 100				
	SA	GA	MAIS	PSO	DE
66	2.899	2.943	2.564	2.579	3.024

Table 9. Average fitness obtained in minutes.

Number of Traffic Logics (NTL)	Number of Vehicles 100				
	SA	GA	MAIS	PSO	DE
66	56.87	123.56	73.86	110.53	112.14

6. Conclusions

This work experimentally studied the performance of a reduced population artificial immune system algorithm.

The simulation of a network of roads in Mexico City has been presented, where routes were added randomly, defining the behavior of 100 vehicles, obtaining results on the average waiting and journey times, the average distance traveled, and the average speed of the vehicles within the road network.

The optimal control of traffic lights is beneficial for minimizing travel times, thus reducing fuel consumption and pollution emissions. Many studies have been carried out using different techniques

for the control of traffic lights, such as meta-heuristic algorithms in conjunction with fuzzy logic, obtaining good results for different cities (see [1,12,14,16,26]). As future research, we seek to implement a bio-inspired meta-heuristic for a medium-size city, in order to reduce pollution rates for a specific region through vehicle flow optimization.

The main conclusions are as follows.

1. Our proposal, MAIS, has a better global performance than the other tested algorithms, namely simulated annealing (SA), the genetic algorithm (GA), particle swarm optimization (PSO), and differential evolution (DE), for this particular problem.
2. By using the Kolmogorov–Smirnov test, we can say that MAIS is significantly different from the other tested algorithms.
3. The results show that MAIS gets good results in the traffic light synchronization problem, so it can be used, just as the other tested algorithms, and can achieve competitive results.

Our future research will test MAIS on a larger urban area. We will also test MAIS on real-life problem applications, where its low computational cost could be a key advantage over other techniques.

Author Contributions: R.G.-C. and M.O.-C. designed this research and collected the data set for the experiment. In addition, R.G.-C. developed the proposed methodology. M.O.-C. and J.C.H.-L. wrote this manuscript and the original draft. J.S.-G. and J.F.S.-T. analyzed the data to show the validity of this paper. R.C.-M. performed all the research steps. C.A.-R. performed the writing—review and editing process. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding from Instituto Politécnico Nacional and the Mexican National Council of Science and Technology.

Acknowledgments: The authors would like to thank the Instituto Politécnico Nacional and the Mexican National Council of Science and Technology for funding for this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chou, L.-D.; Shen, T.-Y.; Tseng, C.-W.; Chang, Y.-J.; Kuo, Y.-W. Green wave-based virtual traffic light management scheme with VANETS. *Int. J. Ad Hoc Ubiquitous Comput.* **2017**, *24*, 22–32. [[CrossRef](#)]
2. Nguyen, P.T.M.; Passow, B.N.; Yang, Y. Improving anytime behavior for traffic signal control optimization based on NSGA-II and local search. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; Volume 10, pp. 4611–4618.
3. Sartikha, A.I.; Sulisty, S. A Study on Metaheuristics for Urban Traffic Light Scheduling Problems. In Proceedings of the 10th International Conference on Information Technology and Electrical Engineering, Kuta, Indonesiapp, 24–26 July 2018; pp. 545–550.
4. Garcia-Nieto, J.; Alba, E.; Olivera, A.C. Swarm intelligence for traffic light scheduling: Application to real urban areas. *Eng. Appl. Artif. Intell.* **2012**, *274*–283. [[CrossRef](#)]
5. Olivera, A.C.; García-Nieto, J.M.; Alba, E. Reducing vehicle emissions and fuel consumption in the city by using particle swarm optimization. *Appl. Intell.* **2018**, *42*, 389–405. [[CrossRef](#)]
6. Jovanović, A.; Nikolić, M.; Teodorović, D. Area-wide urban traffic control: A bee colony optimization approach. *Transp. Res. Part C* **2017**, *77*, 329–350 [[CrossRef](#)]
7. McKenney, D.; White, T. Distributed and adaptive traffic signal control within a realistic traffic simulation. *Eng. Appl. Artif. Intell.* **2012**, *26*, 574–583. [[CrossRef](#)]
8. Sánchez, J.; Galan, J.; Rubio, E. Evolutionary computation applied to urban traffic optimization. In *Advances in Evolutionary Algorithms*; Kosiński, W., Ed.; I-Tech Education and Publishing: Vienna, Austria, 2008; pp. 421–441.
9. Nagare, A.; Bhatia, S. Traffic flow control using neural network. *Int. J. Appl. Inf. Syst.* **2012**, *1*, 50–52. [[CrossRef](#)]
10. Olivera, A.C.; ; Garcia-Nieto, J.; Alba, E. Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2013**, *823*–839. [[CrossRef](#)]

11. Ji, Y.; Hu, B.; Hill, G.; Guo, W.; Blythe, P.; Gao, L. Signal coordination scheme based on traffic emission. *IET Intell. Transp. Syst.* **2016**, *10*, 89–96. [[CrossRef](#)]
12. Damay, N. Multiple-Objective Optimization of Traffic Lights Using a Genetic Algorithm and a Microscopic Traffic Simulator. Master's Thesis, KTH, School of Computer Science and Communication, Toulouse, France, 2015.
13. Ferrer, J.; García-Nieto, J.; Alba, E.; Chicano, F. Intelligent testing of traffic light programs: Validation in smart mobility scenarios. *Math. Probl. Eng.* **2016**, 1–20. [[CrossRef](#)]
14. Peérez, M.; Ruíz, G.; Nasmachnow, S.; Olivera, A.C. Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay. *Appl. Soft Comput.* **2018**, *70*, 472–485. [[CrossRef](#)]
15. Leal, S.S.; De Almeida, P.E.M.; Chung, E. Active control for traffic lights in regions and corridors: An approach based on evolutionary computation. *Transp. Res. Proc.* **2017**, *25*, 1774–1785. [[CrossRef](#)]
16. Vidal, P.; Olivera, A. Management of Urban Traffic Flow Based on Traffic Lights Scheduling Optimization. *IEEE Lat. Am. Trans.* **2019**, 102–110. [[CrossRef](#)]
17. Nunes de Castro, L.; Von Zuben, F.J. The clonal selection algorithm with engineering applications. In Proceedings of the Genetic and Evolutionary Computation Conference, Workshop on AISAA, Las Vegas, NV, USA, 8–12 July 2000; pp. 36–37.
18. Nunes de Castro, L.; Von Zuben, F.J. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.* **2002**, *6*, 239–251. [[CrossRef](#)]
19. Ferrer, J.; Lopez-Ibañes, M.; Alba, E. Reliable simulation-optimization of traffic lights in a real world city. *Appl. Soft Comput. J.* **2019**, *78*, 697–711. [[CrossRef](#)]
20. Goldberg, D.E. Genetic Algorithms in Search. In *Optimization and Machine Learning*; Addison Wesley Publishing Co. Inc.: Boston, MA, USA, 1989.
21. Krishnakumar, K. Micro-genetic algorithms for stationary and non-stationary function optimization. In Proceedings of the Intelligent Control and Adaptive Systems, Philadelphia, Pennsylvania, 7–8 November 1989; pp. 289–296.
22. Rahnamayan, S.; Tizhoosh, H.R. Image Thresholding Using Micro Opposition-Based Differential Evolution (Micro-ODE). In Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China, 1–6 June 2008; pp. 1409–1416.
23. Olguin-Carbajal, M.; Herrera-Lozada, J.C.; Arellano-Verdejo, J.; Barron-Fern, R.; Taud, H. Micro Differential Evolution Performance Empirical Study for High Dimensional Optimization Problems. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2014; Volume 8353, pp. 281–288.
24. Olguin-Carbajal, M.; Alba, E.; Arellano-Verdejo, J. Micro-Differential Evolution with Local Search for High Dimensional Problems. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation Coello, Cancun, Mexico, 20–23 June 2013; pp. 48–54.
25. Herrera-Lozada, J.C.; Calvo, H.; Taud, H. A Micro artificial immune system. *Polibits* **2011**, *43*, 107–111. [[CrossRef](#)]
26. Gao, K.Z.; Zhang, Y.; Zhang, Y.C.; Su, R.; Suganthan, P.N. Meta-Heuristics for Bi-Objective Urban Traffic Light Scheduling Problems. *IEEE Trans. Intell. Transp. Syst.* **2019**. [[CrossRef](#)]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).