On Tuning the Particle Swarm Optimization for Solving the Traffic Light Problem*

Gonçalo O. Silva[0000−0002−9954−4803], Ana Maria A. C. Rocha[0000−0001−8679−2886], Gabriela R. Witeck[0000−0002−0131−7347], António Silva[0000−0001−7075−3364], Dalila Durães[0000−0002−8313−7023], and José Machado[0000−0003−4121−6169]

Centro ALGORITMI, University of Minho, Gualtar, 4710-057 Braga, Portugal
g.oliveirasilva96@gmail.com, arocha@dps.uminho.pt, gabiwiteck@gmail.com, asilva@algoritmi.uminho.pt, dalila.duraes@algoritmi.uminho.pt, jmac@di.uminho.pt

Abstract. In everyday routines, there are multiple situations of high traffic congestion, especially in large cities. Traffic light timed regulated intersections are one of the solutions used to improve traffic flow without the need for large-scale and costly infrastructure changes. A specific situation where traffic lights are used is on single-lane roads, often found on roads under maintenance, narrow roads or bridges where it is impossible to have two lanes. In this paper, a simulation-optimization strategy is tested for this scenario. A Particle Swarm Optimization algorithm is used to find the optimal solution to the traffic light timing problem in order to reduce the waiting times for crossing the lane in a simulated vehicle system. To assess vehicle waiting times, a network is implemented using the Simulation of Urban MObility software. The performance of the PSO is analyzed by testing different parameters of the algorithm in solving the optimization problem. The results of the traffic light time optimization show that the proposed methodology is able to obtain a decrease of almost 26% in the average waiting times.

Keywords: Traffic Lights Problem · Particle Swarm Optimization · Simulation of Urban MObility.

1 Introduction

The current increase of world population and the fast economic growth causes an increase in mobility needs, in particular within big cities where a large number of different vehicle types are present [1]. The existing infrastructure was designed with a distant perception of reality and does not support the amount of vehicles that need to transit within these cities. This situation generates traffic jams with

* This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and the project "Integrated and Innovative Solutions for the well-being of people in complex urban centers" within the Project Scope NORTE-01-0145-FEDER-000086.

a negative impact on society, mainly on the rise of fuel consumption, together with the increase in greenhouse gas emissions and major delays in transport systems [15]. One of the ways to regulate traffic and improve circulation without the need for infrastructures changes is the optimization of traffic light cycle times. A peculiar situation where traffic lights are deployed is on single-lane roads, often found on road under maintenance, narrow roads or small bridges.

The simulation of Urban MObility software (SUMO) is a microscopic traffic flow simulation platform that includes network and demand modeling components [11]. Since its release as an open source traffic simulation package in 2002, SUMO has been supporting the traffic simulation community with a set of traffic modeling utilities on a variety of optimization problems [1, 2, 7, 14].

In this paper the particle swarm optimization (PSO) algorithm was used to optimize the waiting time in the one-lane-two-ways traffic light problem.

The optimization process involves an interaction between the PSO algorithm and the SUMO simulator, where the traffic light cycle times combined with simulated traffic data are used to evaluate the waiting times. Four experiments are conducted to explore how different parameters affect the behaviour of the PSO algorithm. The results obtained in the experiment with the best performance will be compared to a real scenario in order to assess if there is an improvement in the vehicle waiting times in the one-lane-two-ways traffic light system.

The remaining of this paper is organized as follows. In Sect. 2, a literature review is carried out about optimization techniques and simulators used to solve the traffic light problem. Section 3 presents the traffic light optimization problem and the methodology based in the interaction of PSO and SUMO simulator is described in Sect. 4. In Sect. 5, the implementation details and results are presented and discussed. Finally, in Sect. 6, a final review and insight on future work are made.

2 Related Work

The global optimization of traffic light problems has been applied in case studies of smart cities when the real-time control of traffic lights is not possible or expensive. It requires the optimization and simulation of a traffic scenario that is estimated after collecting data from street level sensors. Among the approaches used, metaheuristics such as particle swarm optimization stand out. PSO is a stochastic global optimization technique that is simple to implement and has proven to be effective in several applications [9, 6, 3].

In the literature, simulation-optimization models have been used in several traffic light optimization problems. The PSO algorithm together with the VISSIM (Verkehr In Städten-SIMulationsmodell) micro simulation software [8] was applied to a real data set of traffic flow in a roundabout with 28 traffic signals. The study achieved a reduction of 55,9% in the average delay time per vehicle and an increase on the number of vehicles transiting the roundabout per unit of time of 9,3%. The SUMO simulator and PSO were used to program the traffic light cycle time in [14]. The results obtained with the simulation-based traffic light cycle optimization showed significant improvements in terms of the number of vehicles completing the simulation and the average travel time required for vehicles to reach their destination. In [1], the genetic algorithm and the particle swarm algorithm were applied in a case study of a road network containing 13 traffic lights in order to produce the minimum total travel time. The SUMO software was used to simulate the road network located in the city center.

Recently, in 2020, [2] introduced the social learning particle swarm optimization (SL-PSO) for the real-time traffic light problem in order to mitigate the falling success rate of the classical PSO in high dimensional optimization problems. Based on real traffic data, an intersection was modeled in SUMO and the SL-PSO showed higher computational efficiency and convergence speed than PSO.

3 The Traffic Light Problem

Along the traffic networks on cities, there are different situations where traffic lights are used. The one-lane-two-ways problem under study in this paper can occur mainly in two situations. On one case, the traffic lights are used at a fixed location and can be found on narrow roads or bridges. In this case, the average number of vehicles for each time of the day can be measured and a time fixed offline optimization can be made to reduce waiting times. On the other, temporary traffic lights are used to regulate traffic on spontaneous events such as road maintenance. In these events, the location of traffic lights changes frequently and this unpredictability requires a more dynamic reading of the demand and dimensions of the network.

This work aims to study the PSO parameters that achieve better and faster results in the first situation described above. Based on a small bridge with only one lane in the surroundings of the city of Braga, some network characteristics were measured, mainly the distance between the two traffic lights and their cycle times. Vehicle demand values where not measured but will be staged and simulated by SUMO.

Thus, in this work, the one-lane-two-ways problem will be optimized by the PSO, to determine the green light times of each of the two traffic lights in order to minimize the average waiting time of the vehicles. The average waiting time (W_{time}), in seconds, of all vehicles that go through the system is given in (1). The waiting time is the amount of time that a vehicle is stopped due to involuntary factors like traffic lights and other vehicles. The decision variable vector y refers to the duration time of the green light phase, in seconds, for each traffic light in the system.

The mathematical formulation of the traffic light optimization problem is given by

$$\begin{aligned} &\text{minimize} \quad W_{\text{time}} = \frac{\sum_{i=1}^{N} Wt_i(y)}{N} \\ &\text{subject to } y \in [20, 120]^2 \end{aligned} \tag{1}$$

where W_{time} is the average waiting time value for all the vehicles in the system, $Wt_i(y)$ is the waiting time for each vehicle i considering the green light phase time vector y and N is the total number of vehicles.

4 Methodology

In this section, the methodology is described, presenting the PSO algorithm, the SUMO software as well as the optimization strategy used in this paper.

4.1 Particle Swarm Optimization

Kennedy and Eberhart in 1995 [10], developed the PSO algorithm, a population-based metaheuristic inspired by the movement of birds within flocks, where the population, known as swarm, is composed by particles. The movements of the particles take into consideration their best-known position as well as the global best-known position among the entire swarm. When improved positions are found then they guide the movements of the swarm. The process is repeated until a stopping condition is met.

The candidate solutions of the problem are the positions of each particle, x, that are updated every iteration by its velocity v, as shown in (2).

$$x_{i+1} = x_i + v_{i+1} \tag{2}$$

The velocity of the particle has three components: the inertia, based on the velocity v_i, the cognitive component, based on the best position that the particle itself found (p), and the social component, based on the best position found by the swarm (g). The cognitive and social components are also multiplied by a random number that is uniformly distributed in the interval [0,1] ($r_1, r_2 \in U[0,1]$), respectively. In 1998 Shi and Eberhart [16] introduced a way to influence the inertia, by multiplying its value with the inertia weight, w, balancing the behaviour of the particle between global and local search. Thus, the update of the velocity is computed by

$$v_{i+1} = w * v_i + c_1 * r_1 * (p - x_i) + c_2 * r_2 * (g - x_i) \tag{3}$$

In order to further explore the effect of the inertia weight, Shi and Eberhart [17] used a decreasing linear inertia weight over the iterations to improve PSO performance given by

$$w_{iter} = \frac{iter_{max} - iter}{iter_{max}}(w_{max} - w_{min}) + w_{min} \tag{4}$$

where $w_{max} = 0{,}9$ and $w_{min} = 0{,}4$.

Later, a constriction factor was introduced into PSO to reduce the overall velocity values in order to ensure convergence of the algorithm [4, 5]. Thus, the new update of the velocity of the particle is given by

$$v_{i+1} = K * [w * v_i + c_1 * r_1 * (p - x_i) + c_2 * r_2 * (g - x_i)] \tag{5}$$

where the constriction factor is calculated as

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4 * \varphi}|}, \text{ where } \varphi = c_1 + c_2, \varphi > 4 \tag{6}$$

The pseudocode of PSO is presented in Algorithm 1.

Algorithm 1 Pseudocode of PSO

```
Inicialize swarm
while stopping condition not satisfied do
    for each particle do
        Evaluate objective function
        Update best position
    end for
    Update global best position
    for each particle do
        Update velocity (using (3) or (5))
        Update position (using (2))
    end for
end while
```

4.2 Simulation of Urban MObility

Eclipse SUMO [12] is an open source, purely microscopic and multi-modal traffic simulator that allows to simulate a variety of traffic management topics. The simulations are deterministic where each vehicle is modelled with its own characteristics and route throughout the network. Like any other simulation software, it allows the assessment of infrastructure and policy changes before implementation.

SUMO makes use of input files in order to assembly the simulation model, in particular, a configuration file (`.sumocfg.xml`), a network file (`net.xml`) and a route file (`.rou.xml`). The configuration file is used to load together the network and route descriptions from the input files, as well as, to detail processing decisions and to select the necessary outputs. The network file describes the traffic infrastructure like road and intersections where the vehicles will run during the simulation. A SUMO network is a graph where the edges are the streets, with the position, shape and speed limit of every lane, and the nodes are the intersections. In case of the intersections having traffic lights, their logic is also described in the file. The route file describes all vehicle characteristics and all the different possible routes. A route is a set of edges and nodes. In order to create the network and route files, SUMO provides a graphical network editor named *netedit*, which allows to create different networks from scratch.

4.3 Simulation-Optimization Strategy

Similarly to [7], the optimization strategy was defined as a two-step routine: the optimization algorithm and the simulation process. In the first step, the PSO algorithm is applied to find the optimal, or near-optimal, traffic light configuration in order to minimize the average waiting time. In the second step, the SUMO software is employed to evaluate the traffic light configurations created by the PSO, returning the waiting time of each vehicle. Figure 1 shows the simulation and optimization process. Every time the PSO finds a new configuration for the traffic lights (i.e., green light phase times), this configuration is sent as input to the SUMO simulator for testing. In a similar way, after every simulation run, the output results (i.e., the waiting time for each vehicle in the system) are sent back for further optimization.

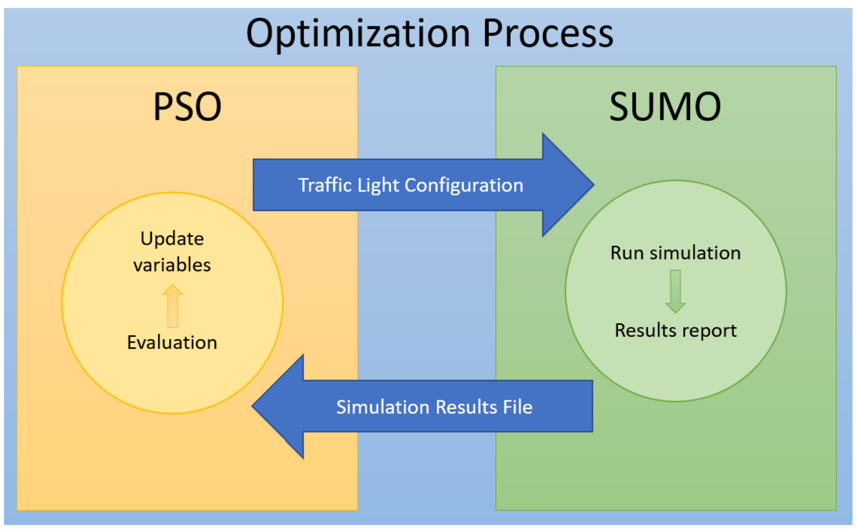

Fig. 1. Interaction between the optimization algorithm and the simulator.

5 Simulation-Optimization Results

First, the technical details of SUMO configuration and the PSO parameters are analyzed for the proposed simulation and optimization methodology. Then, the obtained results from their application are presented and discussed. A *python* code was developed to program the PSO algorithm, to call the SUMO simulator and also ensure correct communication between the two parts. A PC running Windows 10 operating system equipped with AMD Ryzen 7 4800H CPU @ 2.90 GHz, 16 GB RAM was used.

5.1 SUMO Input Configuration

As previously mentioned, this paper aims to study the one-lane-two-ways road with alternating circulation. Naturally, the first step was to represent the network in the SUMO tool, *netedit*. With the real world scenario in mind, the problem network was built as shown in Fig. 2.

Fig. 2. Network representation.

During the design process, several characteristics of the road were taken into account, such as the road length, the traffic light logic and the speed limits. Two different maximum speed limits were set, 30 km/h for the road between the traffic lights and 50 km/h for the remaining of the road. As usual, an all-red mandatory phase was set to ensure that the lane was empty before any of the traffic lights turned green to avoid collisions. Based on the time necessary to drive through the distance between the two traffic lights (200 meters), plus a safety margin, a time of 30 seconds was set for this phase. All this network settings were saved in the SUMO network file.

In order to create the SUMO route file the type of vehicle as well as the routes were defined as follows. Only one type of car was outlined, the passenger vehicle, with default attribute values with some minor changes to its acceleration and deceleration abilities, 3 m/s^2 and 6 m/s^2, respectively. Note that the heavy vehicles are prohibited to cross the bridge due to its dimensions and weight. In this scenario, there were only two different routes, one where the vehicles can only travel from the left to the right, $route01$, and another that is reversed from the first one, $route02$. The arrival frequency was arbitrarily defined, although the model is able to calculate it based on real average demand data. Thus, a car arrives every ten seconds for $route01$ and for $route02$ a frequency of one car every thirteen seconds was defined. Considering the total duration of the simulation of one hour of traffic (3600 seconds), the arrival time of each vehicle during the simulation was assigned according to a random uniform distribution. Therefore, a total of 628 vehicles belong to the simulation network being 352 to circulate on $route01$ and 276 to travel on $route02$. A simulation stopping criterion was defined for when all vehicles leave the network to make sure all waiting times are considered on the output results.

5.2 PSO Implementation Details

In all experiments, the number of particles was set to 20, as used in an extensive PSO inertia weight study [13], the algorithm stopping condition was set to 50 iterations, to limit execution times, and each experiment was executed 30 times since PSO is a stochastic algorithm. Furthermore, the parameters used were chosen after a small exercise with different values and according to previous studies [5, 18].

In order to better understand PSO particles behaviour in this context, four experiments where carried out. This experiments were mainly focused on the effect that different particle velocity parameters can have on PSO convergence. Table 1 shows the parameters used in each experiment, mainly different values of inertia weight (w), cognitive parameter (c_1) and social parameter (c_2). In

Table 1. PSO parameters for the different experiments

Experiment	w	c_1	c_2
1	0,4	2	2
2	0,9	2	2
3	0,9 to 0,4	2	2
4	1	2,05	2,05

Experiment 1 and 2, two different fixed values of the inertia weight were used to find out the effect of a relative low and high inertia respectively. After that, in Experiment 3, a linear decreasing inertia weight was used between the two previous values $[0,4; 0,9]$. In the last experiment, a constriction factor of $K = 0,729$ was used to further assess if lower values of particle velocity help the algorithm to find best solutions.

5.3 Results and Discussion

The results obtained with the four experiments are presented in Table 2. The second to fourth columns show the average values over the 30 runs for the average waiting time, the standard deviation and the average time spent for each run. The last three columns present the values obtained for the best run, in particular the best average waiting time and the green phase times of each traffic light, y_1 and y_2, respectively.

Table 2. Simulation-optimization results.

	Average values among 30 runs			Best Solution		
Experiment	Avg W_{time}	Std Dev	Avg Runtime	W_{time}	y_1	y_2
1	39,294	0,471	1242,888	**38,71**	44,782	32,170
2	39,746	0,298	1230,873	39,14	45,278	31,754
3	39,376	0,341	1273,502	38,77	44,427	32,547
4	**39,228**	0,443	**1209,352**	38,73	44,951	31,997

From Table 2, the proposed strategy that obtained better results on average was Experiment 4 (in bold). However, the best solution was obtained in Experiment 1 resulting in a time of $y_1 = 44{,}782$ and $y_2 = 32{,}170$ seconds for the green phase times of each traffic light. All experimental results seem similar in terms of running time, although a slightly shorter runtime can be observed in Experiment 4, perhaps due to the delay of the last vehicles leaving the system being lower, ending the simulations earlier.

To better analyze the behaviour of the particles during the optimization, Fig. 3 depicts the particle positions along the iterations for the best run of each experiment. In the first three experiments, some particles are still far from the best solution found, specially in Experiment 2 where the high inertia caused a slower convergence. The effect of the social and cognitive parameters is the likely cause of some particles escaping the best solution area. In Experiment 4, the particles seem to continuously converge towards one point, the global one.

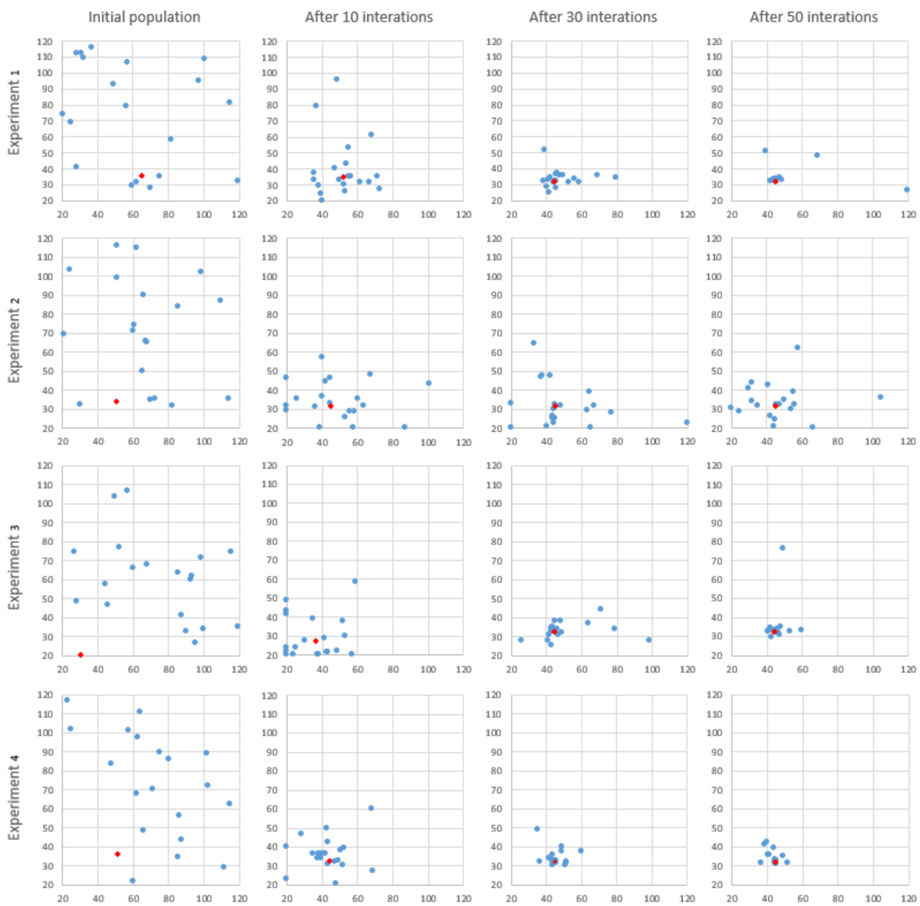

Fig. 3. Comparison of the population position of the best run, of each experiment on different iterations. The red diamond marks the global best found so far.

In the following, a comparison to a real scenario simulation considering green phase times of 120 seconds ($y_1 = 120$ and $y_2 = 120$) on each traffic light was conducted. The SUMO simulator was run one time with the same configuration values given in Sect. 5.1 and with these green phase time values. After the simulation, a value of $W_{\text{time}} = 52{,}04$ seconds was obtained for the average waiting time. Thus, when comparing this value with the best solution obtained with the simulation-optimization strategy, $W_{\text{time}} = 38{,}71$ seconds, there is a reduction of 25,6% in the waiting time. Figure 4 presents the evolution of the number of vehicles in the network during those two types of simulations.

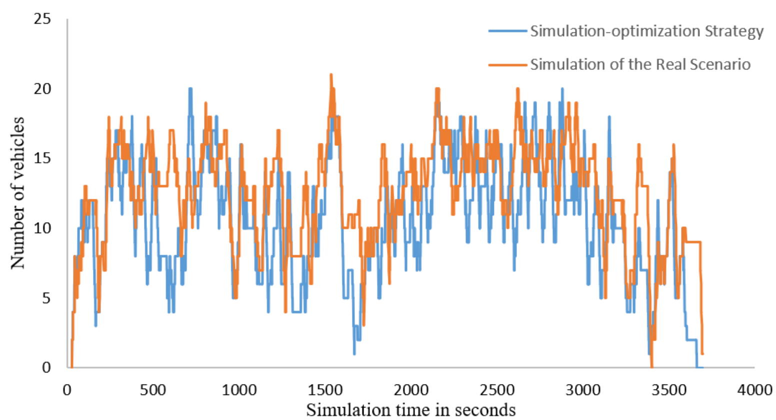

Fig. 4. Number of vehicles in the system along the simulation time of 3600 seconds, in the simulation-optimization strategy and in the simulation of the real scenario.

The results showed that the number of vehicles in the system was similar in both scenarios, although a reduction in the average waiting time should translate into reduced traffic congestion. However, there were vehicles that could not enter the system at the scheduled time because the queue at traffic lights reached and exceeded the network limits. Figure 5 represents the evolution of the number of vehicles waiting to enter the system for both scenarios, revealing a higher number of vehicles on the real scenario. Therefore, the simulation-optimization strategy effectively reduced traffic congestion.

The comparison between the waiting time along the simulation time of 3600 seconds in the simulation-optimization strategy and in the simulation of the real scenario can be seen in Fig. 6. There is a clear difference between the range of values of the two figures. In the simulation-optimization strategy, the waiting time is more evenly distributed across all vehicles, reaching a maximum of 107 seconds, while a large increase to 184 seconds is registered in the real scenario simulation.

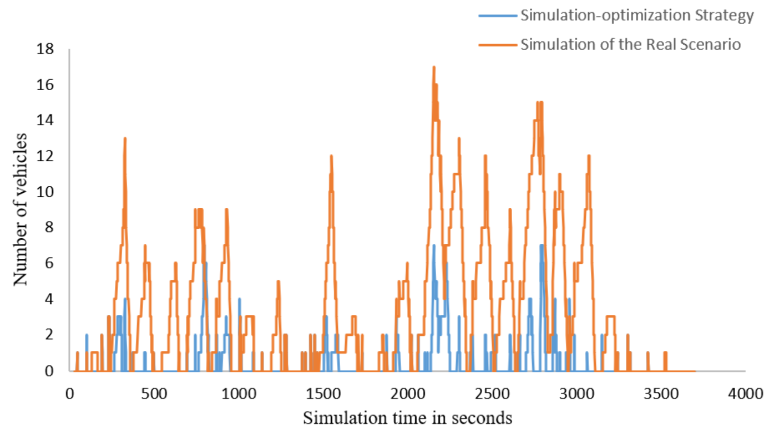

Fig. 5. Number of vehicles waiting in queue to enter the system along the simulation time of 3600 seconds, in the simulation-optimization strategy and in the simulation of the real scenario.

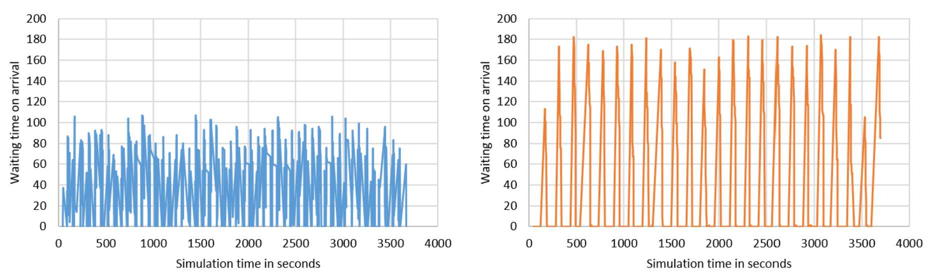

Fig. 6. The waiting time along the simulation time of 3600 seconds, in the simulation-optimization strategy (on the left) and in the real scenario simulation (on the right).

6 Conclusions and Future Work

Mobility needs, especially in large urban cities, are always on the rise due to the fast economic growth and to the increasing world population. Current stress on existing infrastructure causes traffic jams that are harmful to the health of all citizens with increasing greenhouse gas emissions and noise pollution. The adjustment of the traffic light cycle times to the network requirements is seen as a good solution to this problem.

This paper aimed to optimize the waiting time in the one-lane-two-ways traffic light problem. Thus, a simulation-optimization strategy, based on PSO algorithm and SUMO simulator, was used for solving the traffic light configuration problem. The objective was to minimize the average waiting time of the vehicles in a simulated system. The SUMO simulation software evaluates the traffic light configurations created by the PSO, returning the waiting time for each vehicle.

Different values to the inertia, social and cognitive parameters were used to assess the convergence of the PSO. Four experiments were conducted, and the best solution for the green phase times of each traffic light with values of $y_1 = 44{,}782$, $y_2 = 32{,}170$ and $W_{\text{time}} = 38{,}71$ seconds was obtained in Experiment 1. When comparing these values with a simulation of a real scenario, under the same conditions of the experiments, there was a reduction of the average waiting time by more than 25% when using the simulation-optimization strategy for optimizing the traffic light cycle times.

In the future, the effect of the traffic lights on the surrounding network will be taken into account. A study on the three-way intersection and its possible combinations will be carried out and the utilization of other optimization algorithms will be considered.

References

1. Abushehab, R.K., Abdalhaq, B.K., Sartawi, B.: Genetic vs. particle swarm optimization techniques for traffic light signals timing. In: 2014 6th International Conference on Computer Science and Information Technology (CSIT). pp. 27–35. IEEE (2014)
2. Celtek, S.A., Durdu, A., Alı, M.E.M.: Real-time traffic signal control with swarm optimization methods. Measurement **166**, 108206 (2020)
3. Chouikhi, N., Ammar, B., Rokbani, N., Alimi, A.M.: Pso-based analysis of echo state network parameters for time series forecasting. Applied Soft Computing **55**, 211–225 (2017)
4. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). vol. 3, pp. 1951–1957. IEEE (1999)
5. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512). vol. 1, pp. 84–88. IEEE (2000)
6. Elloumi, W., El Abed, H., Abraham, A., Alimi, A.M.: A comparative study of the improvement of performance using a pso modified by aco applied to tsp. Applied Soft Computing **25**, 234–241 (2014)
7. Garcia-Nieto, J., Olivera, A.C., Alba, E.: Optimal cycle program of traffic lights with particle swarm optimization. IEEE Transactions on Evolutionary Computation **17**(6), 823–839 (2013)
8. Gökçe, M.A., Öner, E., Işık, G.: Traffic signal optimization with particle swarm optimization for signalized roundabouts. simulation **91**(5), 456–466 (2015)
9. Gong, Y.j., Zhang, J.: Real-time traffic signal control for roundabouts by using a pso-based fuzzy controller. In: 2012 IEEE Congress on Evolutionary Computation. pp. 1–8. IEEE (2012)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4, pp. 1942–1948. IEEE (1995)
11. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of sumo-simulation of urban mobility. International journal on advances in systems and measurements **5**(3&4) (2012)

12. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: 2018 21st international conference on intelligent transportation systems (ITSC). pp. 2575–2582. IEEE (2018)
13. Nickabadi, A., Ebadzadeh, M.M., Safabakhsh, R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Applied soft computing **11**(4), 3658–3670 (2011)
14. Panovski, D., Zaharia, T.: Simulation-based vehicular traffic lights optimization. In: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). pp. 258–265. IEEE (2016)
15. Peñabaena-Niebles, R., Cantillo, V., Moura, J.L.: The positive impacts of designing transition between traffic signal plans considering social cost. Transport Policy **87**, 67–76 (2020)
16. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). pp. 69–73. IEEE (1998)
17. Shi, Y., Eberhart, R.: Parameter selection in particle swarm optimization. In: International conference on evolutionary programming. pp. 591–600. Springer (1998)
18. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). vol. 3, pp. 1945–1950. IEEE (1999)