



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Swarm intelligence for traffic light scheduling: Application to real urban areas

J. García-Nieto^{a,*}, E. Alba^{a,1}, A. Carolina Olivera^b^a Dept. de Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, Málaga 29071, Spain^b Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Av. Alem 1253, 8000 Bahía Blanca, Argentina

ARTICLE INFO

Available online 20 May 2011

Keywords:

Traffic light scheduling
Particle swarm optimization
SUMO microscopic simulator of urban mobility
Cycle program optimization
Realistic traffic instances

ABSTRACT

Congestion, pollution, security, parking, noise, and many other problems derived from vehicular traffic are present every day in most cities around the world. The growing number of traffic lights that control the vehicular flow requires a complex scheduling, and hence, automatic systems are indispensable nowadays for optimally tackling this task. In this work, we propose a Swarm Intelligence approach to find successful cycle programs of traffic lights. Using a microscopic traffic simulator, the solutions obtained by our algorithm are evaluated in the context of two large and heterogeneous metropolitan areas located in the cities of Málaga and Sevilla (in Spain). In comparison with cycle programs predefined by experts (close to real ones), our proposal obtains significant profits in terms of two main indicators: the number of vehicles that reach their destinations on time and the global trip time.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays, most cities in the world suffer from an excessive vehicular traffic that provokes severe problems like pollution, congestion, security, parking, and many others. Since changes in the urban area infrastructure are usually not possible researchers often agree in that a correct scheduling of traffic lights can help to reduce these problems by improving the flow of vehicles through the cities (McCrea and Moutari, 2010; Sánchez et al., 2008; Spall and Chin, 1997). At the same time, as traffic lights are installed in cities and its number grows, their joint scheduling becomes complex due to the huge number of combinations that appear, and hence, the use of automatic systems for the optimal cycle programming of traffic lights is a necessary choice.

Current initiatives are focused in the use of simulators (Hewage and Ruwanpura, 2004; Karakuzu and Demirci, 2010; Lim et al., 2001) since they provide an immediate and continuous source of information about the traffic flow. Recent studies in the literature about traffic simulation focused on both, macroscopic (McCrea and Moutari, 2010) and microscopic (Sánchez et al., 2008; Tolba et al., 2005) traffic views. In the last few years, the main efforts are directed towards an accurate microscopic modeling of traffic flow (Karakuzu and Demirci, 2010; Sánchez et al., 2008) and the programming of convenient cycles of traffic lights (Brockfeld et al., 2001; Nagatani, 2010).

In this sense, the use of intelligent methods have demonstrated their usefulness to the optimization of cycle programs of traffic lights (Angulo et al., 2008; Sánchez et al., 2008). However, authors in general have addressed specific urban areas with few intersections and small number of traffic lights (Brockfeld et al., 2001), and most of them apply ad-hoc algorithms designed only for one specific instance (Angulo et al., 2008; Sánchez et al., 2008). The use of intelligent techniques for large and heterogeneous cases of study is still an open issue (Nagatani, 2010; Roupail et al., 2000).

All these motivations drive us to propose an optimization strategy here based in a particle swarm optimization (PSO) algorithm (Montes de Oca et al., 2009; Kennedy and Eberhart, 2001) that can find successful cycle programs of traffic lights. Several features led us to use PSO instead of other evolutionary methods: first, the PSO is a well-known algorithm shown to perform a fast converge to suitable solutions (Clerc and Kennedy, 2002). This is a highly desirable property for the optimal cycle program of traffic lights, where new immediate traffic light schedules should be required to face updating events in traffic scenarios. Second, the canonical PSO is easy to implement, and requires few tuning parameters (Clerc and Kennedy, 2002; Montes de Oca et al., 2009; Kennedy and Eberhart, 2001). Third, PSO is a kind of Swarm Intelligence algorithm that can inform us on future issues to deal with this problem using independent agents in the system for online adaptation (a future line of us).

A microscopic traffic simulator is then coupled with our PSO for the evaluation of cycle programs (codified as vector solutions) for the traffic lights that control the flow of vehicles through a

* Corresponding author.

E-mail addresses: jnieto@lcc.uma.es (J. García-Nieto), eat@lcc.uma.es (E. Alba), aco@cs.uns.edu.ar (A. Carolina Olivera).

given scenario instance. In this particular work we use SUMO (simulator of urban mobility) (Krajewicz et al., 2006).

As a first contribution of this work, our proposed PSO is tested with real data of two large and heterogeneous metropolitan areas with hundreds of traffic lights located in the cities of Sevilla and Málaga, in Spain. The results are analyzed under different road conditions. Secondly, in comparison with predefined cycle programs close to real ones, our PSO will be shown to obtain quantitative improvements in terms of two main objectives: the number of vehicles that reach their destinations and their global trip time.

The remaining of this article is organized as follows. In Section 2, a review of related works in the literature is presented. In Section 3, our optimization approach is described. Section 4 presents the experimental methodology used and the results obtained. Conclusions and future work are given in Section 5.

2. Literature overview

Recently, metaheuristic algorithms (Blum and Roli, 2003) have become very popular as optimization methods for solving traffic light scheduling problems. A first attempt corresponds to Rouphail et al. (2000), where a genetic algorithm (GA) was coupled with the CORSIM (Holm et al., 2007) microsimulator for the timing optimization of nine intersections in the city of Chicago (USA). The results, in terms of total queue size, were limited due to the delayed convergence behavior of the GA.

In Teklu et al. (2007), the impact of signal time changes with respect to the drivers were analyzed. More precisely, authors considered the problem of determining optimum signal timings while anticipating the responses of drivers as an instance of the network design problem (NDP). In order to solve the traffic equilibrium problem they used the SATURN package (simulation-assignment modeling software, Van Vliet, 1982). Authors applied a macroscopic point of view of the traffic flow and they employed a GA to compute the signal setting NDP (cycle time, offset, and green light times for stages). It is important to note that the chromosome (grey-code) encoding was done differently for each particular instance under study. The algorithm was tested with the city of Chester in UK, mainly addressing a complete GA parameter analysis, not actually the traffic problem.

In Sánchez et al. (2008), following the model proposed in Brockfeld et al. (2001), the authors designed a GA with the objective of optimizing the cycle programming of traffic lights. This GA was tested in a commercial area in Santa Cruz de Tenerife (Spain). In this work, every intersection was considered to have independent cycles. As individual encoding they used a similar binary (grey-code) representation to the one used in Teklu et al. (2007). The computation of valid states was done before the algorithm runs, and it strongly depended on the scenario instance tackled.

Turky et al. (2009) used a GA to improve the performance of traffic lights and pedestrians crossing control in a unique intersection with four-way two-lane. The algorithm solved the limitations of traditional fixed-time control for passing vehicles and pedestrians, and it employed a dynamic control system to monitor two sets of parameters.

A few works (three) related to the application of particle swarm optimization for the schedule of traffic lights also exist. One of the most representative was developed by Chen and Xu (2006), where they applied a PSO for training a fuzzy logic controller located in each intersection by determining the effective time of green for each phase of the traffic lights. A very simple network with two basic junctions was used for testing this PSO.

More recently, Peng et al. (2009) presented a PSO with isolation niches to the schedule of traffic lights. In this approach, a custom microscopic view of the traffic flow was proposed for

the evaluation of the solutions. One single academic instance with a restrictive one-way road with two intersections was used to test the PSO. Nevertheless, this last study was focused on the capacity of isolation niches to maintain the diversity of the PSO population, and was not very involved with the problem itself.

Finally, in Kachroudi and Bhouri (2009) a multiobjective version of PSO is applied for optimizing cycle programs using a predictive model control based on a public transport progression model. In this work, private and public vehicles' models are used performing simulations on a virtual urban road network made up of 16 intersections and 51 links. Each intersection is then controlled by a traffic light with the same cycle time of 80 s.

All these approaches focused on different aspects of the traffic light scheduling. However, three common weak points can be found in all of them:

- They tackled limited vehicular networks with very few traffic lights and a small number of other elements (roads, intersections, directions, etc.). In contrast, our PSO can find optimized cycle programs for large scenarios with hundreds of traffic lights, vehicles, and other elements.
- They were designed for only one specific scenario. Some of them studied the influence of the traffic density. Our approach can be easily adapted to different scenario topologies.
- They were not compared against other techniques. Our PSO is compared here against two different approaches: a Random Search algorithm and the cycle program generator provided by SUMO.

3. PSO for traffic light scheduling

This section describes our optimization approach proposed for the optimal cycle programs of traffic lights. It details the solution encoding, the fitness function, and finally the global optimization procedure. Previous to this, basic notions about the PSO algorithm are given.

3.1. Particle swarm optimization

Inspired in the social behavior of birds within a flock, particle swarm optimization (Montes de Oca et al., 2009; Kennedy and Eberhart, 2001) is a population-based metaheuristic initially designed for continuous optimization problems. In PSO, each potential solution to the problem is called particle *position* and the population of particles is called the *swarm*. In this algorithm, each particle position x^i is updated each iteration g by means of

$$x_{g+1}^i \leftarrow x_g^i + v_{g+1}^i, \quad (1)$$

where term v_{g+1}^i is the velocity of the particle, given by the following equation:

$$v_{g+1}^i \leftarrow w \cdot v_g^i + \varphi_1 \cdot UN(0,1) \cdot (p_g^i - x_g^i) + \varphi_2 \cdot UN(0,1) \cdot (b_g - x_g^i). \quad (2)$$

In this formula, p_g^i is the best solution that the particle i has seen so far, b_g is the global best particle (also known as the *leader*) that the entire swarm has ever created, and w is the inertia weight of the particle (it controls the trade-off between exploration and exploitation). Finally, φ_1 and φ_2 are specific parameters which control the relative effect of the personal and global best particles, while $UN(0,1)$ is a uniform random value in $[0,1]$ which is sampled anew for each component of the velocity vector and for every particle and iteration.

Algorithm 1 describes the pseudo-code of PSO. The algorithm starts by initializing the swarm (Line 1), which includes both the positions and velocities of the particles. The corresponding p^i of

each particle is randomly initialized, and the leader b is computed as the best particle of the swarm (Line 2). Then, during a maximum number of iterations, each particle moves through the search space updating its velocity and position (Lines 5 and 6). The particles are then evaluated (Line 7), and their personal best positions p^i are also recalculated (Line 8). At the end of each iteration, the leader b is also updated.

Algorithm 1. Pseudocode of PSO.

```

1. initializeSwarm()
2. locateLeader( $b$ )
3. while  $g < \text{maxGenerations}$  do
4.   for each particle  $x_g^i$  do
5.     updateVelocity( $v_g^i$ ) do //Eq. (2)
6.     updatePosition( $x_g^i$ ) // Eq. (1)
7.     evaluate( $x_g^i$ )
8.     update( $p_g^i$ )
9.   end for
10.  updateLeader( $b_g$ )
11. end while

```

The particle swarm optimization algorithm is currently employed in multitude of engineering problems (Alba et al., 2008; García-Nieto and Alba, 2010; Kennedy and Eberhart, 2001; Parsopoulos, 2005) showing a successful performance, even in comparison with other modern optimization techniques (Alba et al., 2007a; García-Nieto et al., 2010). We also hope so in the cycle program of traffic lights problem using real data.

3.2. Cycle program of traffic lights

A urban traffic scenario is basically composed by: intersections, traffic lights, roads, directions, and vehicles moving through their previously specified routes. The traffic lights are located in

intersections, and control the flow of vehicles by following their programs of color states and cycle durations. In this context, all traffic lights located in the same intersection are governed by a common program, since they have to be necessarily synchronized for traffic security. In addition, for all the traffic lights in an intersection, the combination of color states during a cycle period is always kept valid (Leung et al., 2004) and it must follow the specific traffic rules of intersections, in order to avoid vehicle collisions and accidents. In this sense, we work only with valid combinations of color states for each intersection, which cannot be modified during the optimization process. This avoids invalid combinations of color states and restricts the optimization approach to work only with feasible states.

In this context, our main objective is to find optimized cycle programs (OCP) for all the traffic lights located in a given urban area. Specifically, cycle programs are refereed to the time span a set of traffic lights (in a junction) keep their color states. At the same time, these programs have to coordinate traffic lights in adjacent intersections with the aim of improving the global flow of vehicles circulating according to traffic regulations.

An example of this mechanism can be observed in Fig. 1 where the intersection with $\text{id}="i"$ contains seven phases with durations 40, 5, 40, 10, 36, 6, and 22 s (simulation steps). In these phases, the states have fourteen signals (colors), corresponding each one of them to one of the fourteen traffic lights located in the studied intersection. These states are the valid ones generated by the simulator (SUMO in this work) attending to real traffic rules. In this instance, the fifth phase contains the state " rrrr GGr rrrr GGr " meaning that four traffic lights are in green (G), and the ten others are in red (r) during 36 s. The following phase changes the state of the four traffic lights to other valid combination, for example, " GGGG yyr GGGG yyr " (y means yellow) during 6 s, and so on. The last phase is followed by the first one, and this cycle is repeated during all the simulation time. All the intersections in

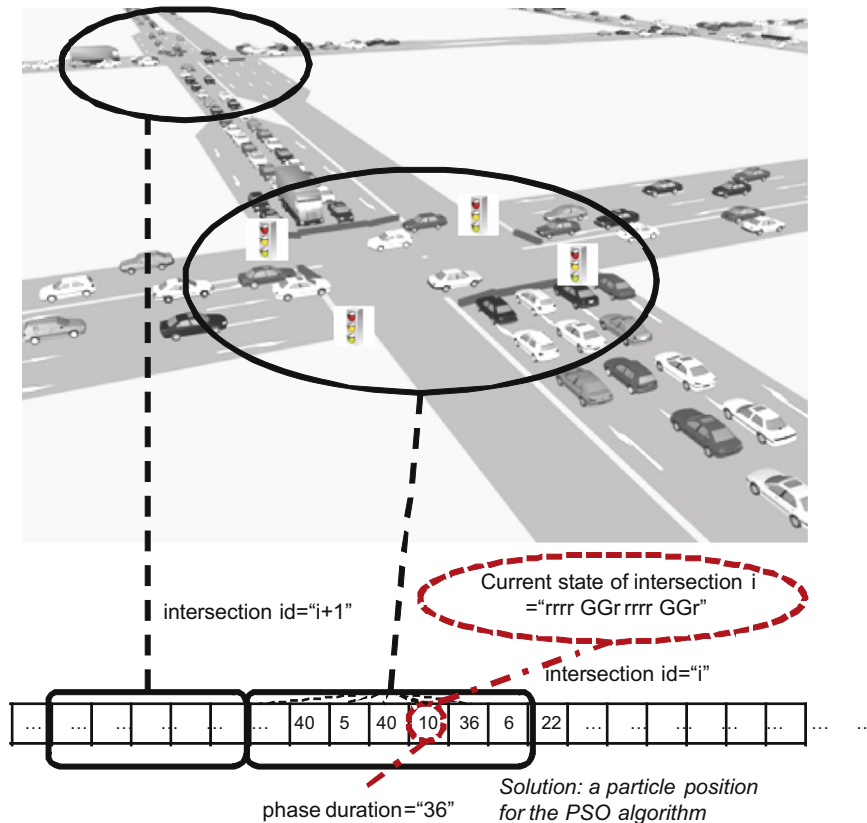


Fig. 1. Cycle program (phase duration) of traffic lights within intersections. Integer codification inside a PSO tentative solution.

the complete scenario perform their own programming cycles of phases at the same time, hence conforming the global schedule of traffic lights. As commented before, computing OCP consists in optimizing the combination of phase durations of all traffic lights (in all intersections) with the aim of improving the global flow of vehicles circulating in a urban scenario instance.

A final indication in this sense concerns the behavior of the vehicles involved in a SUMO simulation, that depends on both road directions and speed. SUMO employs a space-discrete extended model as introduced by Krauß (1998). In this model, the streets are divided into cells and the vehicles circulating through the streets go from one cell to another if both, the sense and the direction are allowed. The speed of each vehicle depends on its distance to the vehicle in front of it, with a preestablished maximum speed typical of urban areas (50 km/h in our study).

3.3. Solution encoding

In our approach, the OCP is encoded by means of a vector of integers (see Fig. 1) following the SUMO structure of programming cycles, where each element represents a phase duration of one state of the traffic lights involved in a given intersection.

In spite of its simplicity, this solution representation allows our PSO to take into account the interdependency of variables, not only between phase durations of a state of traffic lights in an intersection, but also between traffic lights in adjacent intersections.

3.4. Fitness function

In order to evaluate each cycle program generated by our PSO the following fitness function is used, which considers the information obtained from the events happening during the simulation:

$$fitness = \frac{TT + SW + (NV * ST)}{V^2 + P}. \quad (3)$$

The main objective (Eq. (3)) is to maximize the number of vehicles that reach their destinations (V) and minimize the global trip time of all the vehicles (TT), during the simulation time (ST). The number of vehicles that arrive to their destinations is squared (V^2) in order to prioritize it over the other terms and factors. Obviously, the number of vehicles that do not reach their destinations (NV) has to be minimized. The global trip time concerns an aggregation of the trip time of vehicles that reach their destinations during the simulation process. On the contrary, vehicles with uncompleted travels consume all the simulation time, which implies an additional penalization.

Another important term that we have considered (to minimize) concerns the state of the traffic lights in each precise moment, since it influences the time that each vehicle must stop and wait (SW), with the consequent delay over its own trip time. On the one hand, a prolonged state of traffic lights in red could collapse the intersection where it is, and even other close intersections. However, a prolonged state in green could improve the traffic flow in a given area or direction, but also makes the traffic flow of other areas and directions worse. On the other hand, we observed in some specific simulations that large waiting times (in certain traffic lights) could lead stopped vehicles to avoid immediate collapsed intersections, and hence contributing to alleviate them. The fact of remaining stopped in a traffic jam could yield a higher delay in the trip time of vehicles, and even increment the number of vehicles that do not reach their destinations during the simulation time. This behavior led us to believe that more waiting time did not necessarily entail more

time to finish the trip, and hence, to explicitly add this value (SW) to the fitness function.

Finally, the balanced proportion of colors in the phase duration of the states should promote those states with more traffic lights in green located in streets with a high number of vehicles circulating, and traffic lights in red located in streets with a low number of vehicles moving. The proportion of colors in each phase (ph) of all the intersections tl can be formulated as follows:

$$P = \sum_{k=0}^{tl} \sum_{j=0}^{ph} s_{kj} * \left(\frac{G_{kj}}{r_{kj}} \right), \quad (4)$$

where G_{kj} is the number of traffic lights in green, and r_{kj} is number of traffic lights in red in the phase state j (with duration s_{kj}) and in the intersection k . The minimum value of red_{kj} is 1 in order to avoid division by 0.

3.5. PSO for OCP

The proposed optimization strategy is then composed by two main parts: an optimization algorithm and a simulation procedure.

The optimization part is carried out by means of the PSO algorithm which has been specially adapted to optimize cycle programs for traffic lights. Concretely, three main modifications have been carried out as follows:

1. The *initial swarm* is composed by a number of particles (solutions) initialized with a set of random values representing the phase durations. These values are within the time interval $[5, 50] \in \mathbb{Z}^+$, and constitute the range of possible time spans (in seconds) a traffic light can kept a signal color (only green or red, the time for yellow is a constant value). We have specified this interval by following several examples of real traffic light schedules provided by the City Council of Málaga (Spain).
2. The *velocity calculation* has been softly modified in order to deal with integer combinatorial values by truncating (with floor and ceiling functions) all positions of the new velocity vector as Eq. (5) shows:

$$v_{g+1}^i = \begin{cases} \lfloor v_{g+1/2}^i \rfloor & \text{if } UN(0,1)^i \leq \lambda, \\ \lceil v_{g+1/2}^i \rceil & \text{otherwise.} \end{cases} \quad (5)$$

In this formula, $v_{g+1/2}^i$ is the intermediate velocity value obtained from Eq. (2). The parameter λ determines the probability of performing ceil or floor functions in the velocity calculation ($\lambda = 0.5$ for this study). The specific choice of randomly applying of ceil/floor (instead of simply approximating the velocity value to ceil or floor functions) can alleviate the problem of vanishing velocities that may appear after some iterations as the (real) velocity vector approaches zero.

3. The *inertia weight* changes linearly through the optimization process by using the following equation:

$$\omega \leftarrow \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \cdot g}{g_{total}}. \quad (6)$$

This way, at the beginning of the process a high inertia (ω_{max}) value is introduced, which decreases until reaching its lowest value (ω_{min}). A high inertia value provides the algorithm with exploration capability and a low inertia promotes exploitation.

The simulation procedure is the way of assigning a quantitative quality value (fitness) to the solutions, thus leading to optimized cycle programs tailored to a given urban scenario instance. This task is tackled by the SUMO microscopic traffic simulator, which accepts new cycle programs of traffic lights and

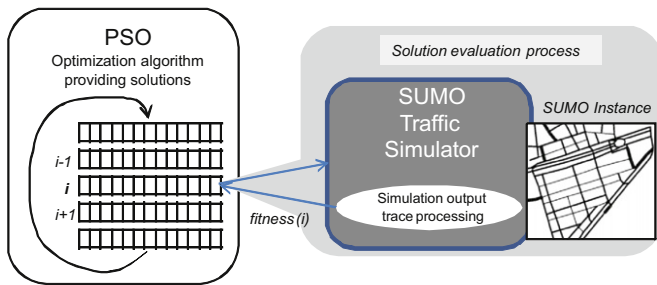


Fig. 2. Optimization strategy for the cycle program configuration of traffic lights. The algorithm invokes SUMO for each solution evaluation.

compute the required values in Eq. (3). Fig. 2 illustrates the complete optimization strategy used in this work.

When PSO generates a new solution, it is used for updating the cycle program. Then, SUMO is started to simulate the instance with streets, directions, obstacles, traffic lights, vehicles, speed, routes, etc., under the new defined schedule of cycle programs. After the simulation, SUMO returns the global information necessary to compute the fitness function. Each solution evaluation requires only one simulation procedure since vehicle routes in SUMO were generated deterministically. In fact, as suggested in Sánchez Medina et al. (2005), stochastic traffic simulators obtain similar results to deterministic ones, the latter allowing huge computing savings.

We must notice that each new cycle program is statically loaded for each simulation procedure. Our aim here is not to generate cycle programs dynamically during an isolated simulation as done in agent-based algorithms (Krajewicz et al., 2005), but obtaining optimized cycle programs for a given scenario and timetable. In this sense, what real traffic light human schedulers actually demand are constant cycle programs for specific areas and for preestablished time periods (rush hours, nocturne periods, etc.), which led us to take this focus.

4. Experiments and results

In this section we present the experimental framework followed to assess the performance of our PSO algorithm for creating OCP. First, we describe the scenario instances generated specifically for this work, the implementation details of our approach, and the parameter settings. Later, results and comparisons are presented. A study of resulted cycle programs is also carried out in order to show the real benefits of using our proposal.

4.1. Instances

As we are interested in developing an optimization solver capable of dealing with close-to-reality and generic urban areas, we have generated two scenarios by extracting actual information from real digital maps. These two scenarios cover similar areas of approximately 0.75 km², and they are physically located in the cities of Sevilla and Málaga in Spain. The information used concerns: traffic rules, traffic element locations, buildings, road directions, streets, intersections, etc. Moreover, we have set the number of vehicles circulating, as well as their speeds by following current specifications available in the Mobility Delegation of the City Hall of Málaga (<http://movilidad.malaga.eu/>). This information was collected from sensorized points in certain streets obtaining a measure of traffic density in several time intervals. In the case of Sevilla we consulted the Mobility Delegation of Sevilla Council (<http://www.trajano.com/>).

In Fig. 3, the selected areas of the two cities are shown with their corresponding snapshots of Google Earth, OpenStreetMap, and SUMO. This figure illustrates the process of generating the traffic network instances. The specific features of these areas are as follows:

1. **Sevilla:** Located in the popular district of Nervión in the city center of Sevilla (Fig. 3, left), it is made up of intersections between streets including from 4 to 17 traffic lights. The complete area shows a regular organization with almost all the junctions connecting between three and four streets. The main avenues crossing this neighborhood are: Menéndez Pelayo, Eduardo Dato, San Francisco Javier, Montoto, Galván, and Buharías.
2. **Málaga:** In the zone between the city center and the harbor. This irregular scenario (Fig. 3, right) is composed by streets with different widths and lengths, and several roundabouts. It contains junctions including from 4 to 16 traffic lights each one of them. The main avenues found in this area are: Alameda Principal, Andalucía, Manuel Agustín Heredia, Colón, and Aurora.

We have chosen these two scenarios since they constitute different metropolitan areas with heterogeneous structures and traffic organizations. The number of studied intersections is 70 for the two instances, and circulating 500 vehicles through each one of them. We have to notice that in spite of having in both instances a similar number of intersections (70), the number of traffic lights is not exactly the same, since they contain different intersection shapes (368 traffic lights in Sevilla and 312 ones in the case of Málaga).

In the simulations, each one of the vehicles performs its own route from origin to destination circulating with a maximum speed of 50 km/h (typical in urban areas). The routes were previously generated by following random paths. The simulation time was set to 500 s (iterations of microsimulation) for each instance. This time was determined as a maximum time for a car to complete its route, even if it must stop in all the traffic lights it finds. When a vehicle leaves the scenario network, it will not appear again.

4.2. Experimental setup

We have used the implementation in C++ of the PSO algorithm provided by the MALLBA (Alba et al., 2007b) framework. The simulation phase is carried out by executing (for the evaluation of particles) the traffic simulator SUMO release 0.12.0 for Linux. The experiments were performed in computers of the laboratories of the Department of Computer Science of the University of Málaga (Spain). Most of them are equipped with modern dual core processors, 1 GB RAM, and Linux Debian O.S. They operate under a Condor (Thain et al., 2005) middleware platform that acts as a distributed task scheduler (each task dealing with one independent run of PSO).

For each scenario instance we have carried out 30 independent runs of our PSO. The swarm (population) size was set to 100 particles performing 300 iteration steps, hence resulting a number of 30,000 solution evaluations (SUMO simulations) per run and instance. As previously mentioned, the particle size directly depends on the number of traffic lights of each instance. The remaining parameters are summarized in Table 1. These parameters were set after preliminary executions of our approach with the Málaga instance. Specific parameters of PSO were selected as recommended in the study about the convergence of this algorithm in Clerc and Kennedy (2002).



Fig. 3. Process of creation of real-world instances for study. Urban centre of Sevilla (37° 38'14"S 5° 9'23"O) and Málaga (36° 43'60"N 4° 25'87"O) instance views. After selecting our area of interest (Google Earth view), it is interpreted by means of the OpenStreetMap tool, and then exported to SUMO format.

Table 1
Simulation and PSO parameters.

Solver phase	Parameter	Value
Simulation details	Simulation time (steps)	500 s
	Simulation area	0.75 km ²
	Number of vehicles	500
	Vehicle speed	0–50 km/h
	No. of studied intersections	70
PSO parameters	Max. no. of evaluations	30,000
	Swarm size	100
	Particle size (no. traffic lights)	368
		312
	Local coefficient (φ_1)	2.0
	Social coefficient (φ_2)	2.0
	Maximum inertia (w_{max})	0.5
	Minimum inertia (w_{min})	0.1
	Velocity truncation factor (λ)	0.5

Additionally, we have implemented a Random Search algorithm, also in the scope of the MALLBA library, with the aim of establishing comparisons against our PSO. Thus, performing the same experimentation procedure with PSO and Random Search algorithm we expect to obtain some insights into the power of our proposal (how much intelligent it is). The pseudocode of the Random Search algorithm (RANDOM from now on) is shown in Algorithm 2. The maximum number of evaluations was set to 30,000, as for PSO.

Algorithm 2. Pseudocode of RANDOM.

```

1. initializeSolution( $x$ )
2.  $i \leftarrow 0$ 
3. while  $i < \text{Max\_Number\_of\_Evaluations}$  do
4.   generate( $x_i$ ) //new solution
5.   if  $f(x) \geq f(x_i)$  then
6.      $x \leftarrow x_i$ 
7.   end if
8.    $i \leftarrow i + 1$ 
9. end while

```

SUMO provides a deterministic algorithm for generating cycle programs (SCPG). Then we also compare the cycle programs obtained by our PSO against the ones obtained by SUMO. This last algorithm basically consists in assigning to the phase durations of the traffic logics fresh values in the range of [6,31], according to three different factors:

1. the proportion of green states in the phases,
2. the number of incoming lanes to the intersection, and
3. the braking time of the vehicles approaching to their traffic lights.

Further information about this algorithm can be found in Krajzewicz et al. (2006).

4.3. Results and comparisons

In this section, we are first interested to analyze the internal performance of our PSO. Graphically, Fig. 4 plots the trace

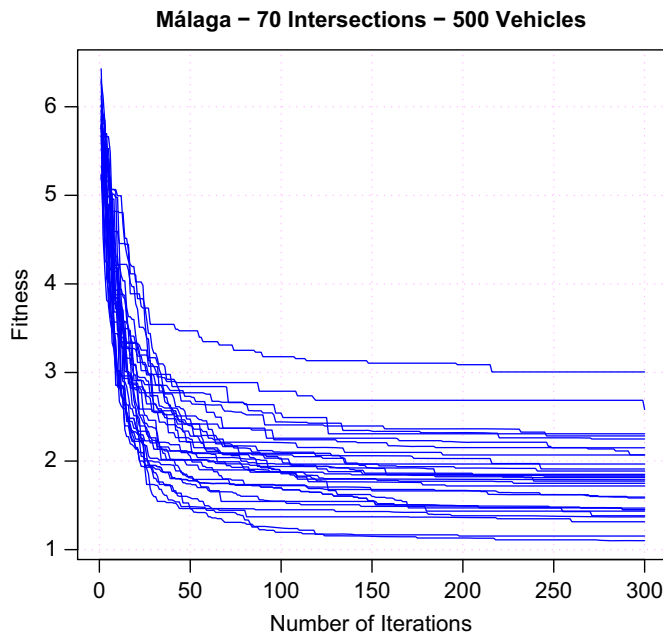


Fig. 4. Trace progress of the best fitness values in 30 independent runs of PSO tackling the Málaga instance.

progress of the obtained in 30 independent runs of our technique when solving the Málaga instance. In this figure, we can observe that for all executions our algorithm practically converged after the first 200 iterations (20,000 evaluations), using the remaining time to only slightly refine solutions. In addition, all the computed solutions are close each other in quality, but different among them. They are almost all in the range of fitness values between 1 and 3. In terms of convergence and robustness, these are desirable features since we can offer to the expert a varied set of accurate cycle programs in a first stage of optimization.

Concerning each individual execution, a representative example can be observed in Fig. 5 where the absolute frequency of the fitness distribution of the entire swarm through the optimization process is plotted. In concrete, it illustrates one of the thirty independent runs of our PSO tackling the Málaga instance. We can notice that the initial particles are diverse and with high fitness values (≈ 8), although during the second half of the optimization process they get convergence on solutions with low fitness values (≤ 1.5).

Table 2 contains the maximum, median, minimum, mean, and standard deviation of the fitness values obtained (out of 30 independent runs) by the proposed PSO for the two scenario instances: Sevilla and Málaga. Additionally, the values obtained by the RANDOM algorithm, and the results of the SCPG are also provided in order to allow comparisons. We can clearly observe in this table that PSO obtained the best results (marked in boldface). Furthermore, the maximum values of our proposal are lower than the mean values showed by both, RANDOM and SCPG algorithms.

With the aim of providing these comparisons with statistical meaning we have applied different *t-test* (Sheskin, 2007) to the numerical values and distributions of the results. We have used this parametric test since the resulting distributions show the conditions of normality and equality of variances. The confidence level was set to 95% ($p\text{-value}=0.05$), which allows us to ensure that all these distributions are statistically different if they result in $p\text{-value} < 0.05$.

In effect, the *t-test of independent samples* applied to the mean values of the distributions of PSO and RANDOM (Table 2) resulted in $p\text{-values} \ll 0.05$, for the two instances: Málaga and Sevilla. In a

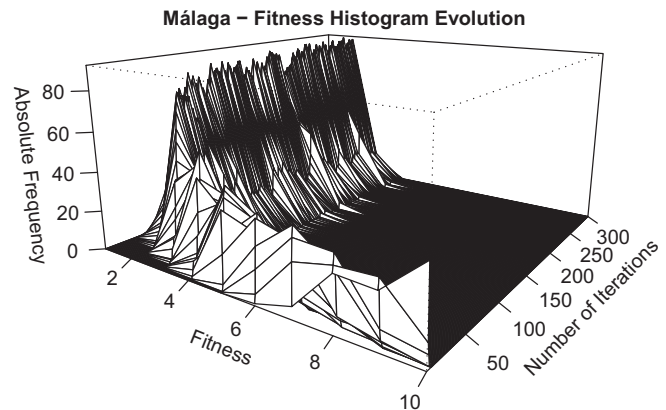


Fig. 5. Swarm fitness histogram through 300 iterations in the optimization of the Málaga scenario.

Table 2

Median fitness values obtained by PSO for Sevilla and Málaga instances. Median fitness obtained by RANDOM and by SCPG algorithms are also provided.

Instance	Value	PSO	RANDOM	SCPG
Sevilla	Maximum	6.53E+00	9.04E+00	7.24E+00
	Median	5.06E+00	8.45E+00	7.24E+00
	Minimum	3.08E+00	7.73E+00	7.24E+00
	Mean	4.95E+00	8.43E+00	7.24E+00
	Std.	6.68E-01	2.58E-01	0.00E+00
Málaga	Maximum	3.01E+00	4.67E+00	5.19E+00
	Median	1.80E+00	4.43E+00	5.19E+00
	Minimum	1.10E+00	3.63E+00	5.19E+00
	Mean	1.81E+00	4.37E+00	5.19E+00
	Std.	4.22E-01	2.55E-01	0.00E+00

similar way, the *t-test of a simple sample* applied to the mean of the distribution of PSO against the simple value of SCPG also resulted in $p\text{-values} \ll 0.05$. Therefore, we can claim that our PSO obtained statistically better results than the other two compared algorithms: RANDOM (stochastic search) and SCPG (deterministic). This also means that our algorithm is intelligent and competent when compared to greedy information and human expert guidelines.

From a graphical point of view, Fig. 6 shows the boxplots of the distribution results of PSO, and RANDOM. The results of SCPG are represented with a \blacklozenge point since this technique always returns the same deterministic result. As expected, the distributions of PSO show better lower quartiles, medians, and upper quartiles than RANDOM and SCPG. For the Sevilla instance, the RANDOM algorithm obtained worse results than SCPG. On the contrary, for the Málaga instance the results of RANDOM are better than the one of SCPG. Possibly, the higher number of traffic lights in the case of Sevilla makes the SCPG algorithm to obtain more successful cycle programs than RANDOM. This last evidently shows a limited performance for the most complex instance (Sevilla).

4.4. Analysis of resulting traffic light schedules

In this section we focus on the cycle programs obtained by our PSO, and the possible profits they can offer to the actual users in this field. Then we show the real impact of using our optimization technique, able of computing realistic and comprehensive traffic light cycle programs.

In this context, for each iteration step of the PSO and for each particle in the swarm, we have saved the information obtained from each simulation (solution evaluation) about the number of

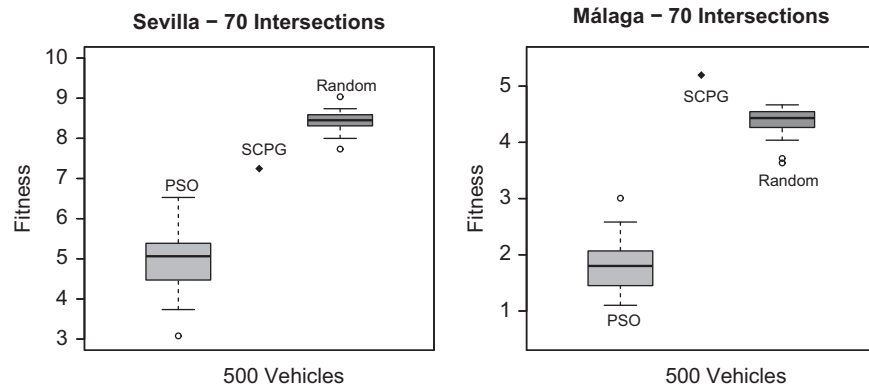


Fig. 6. Boxplots representation of distributions results of Sevilla (left) and Málaga (right) instances. The results of SCPG are represented with a ♦ point since this technique always returns the same deterministic result for a given instance.

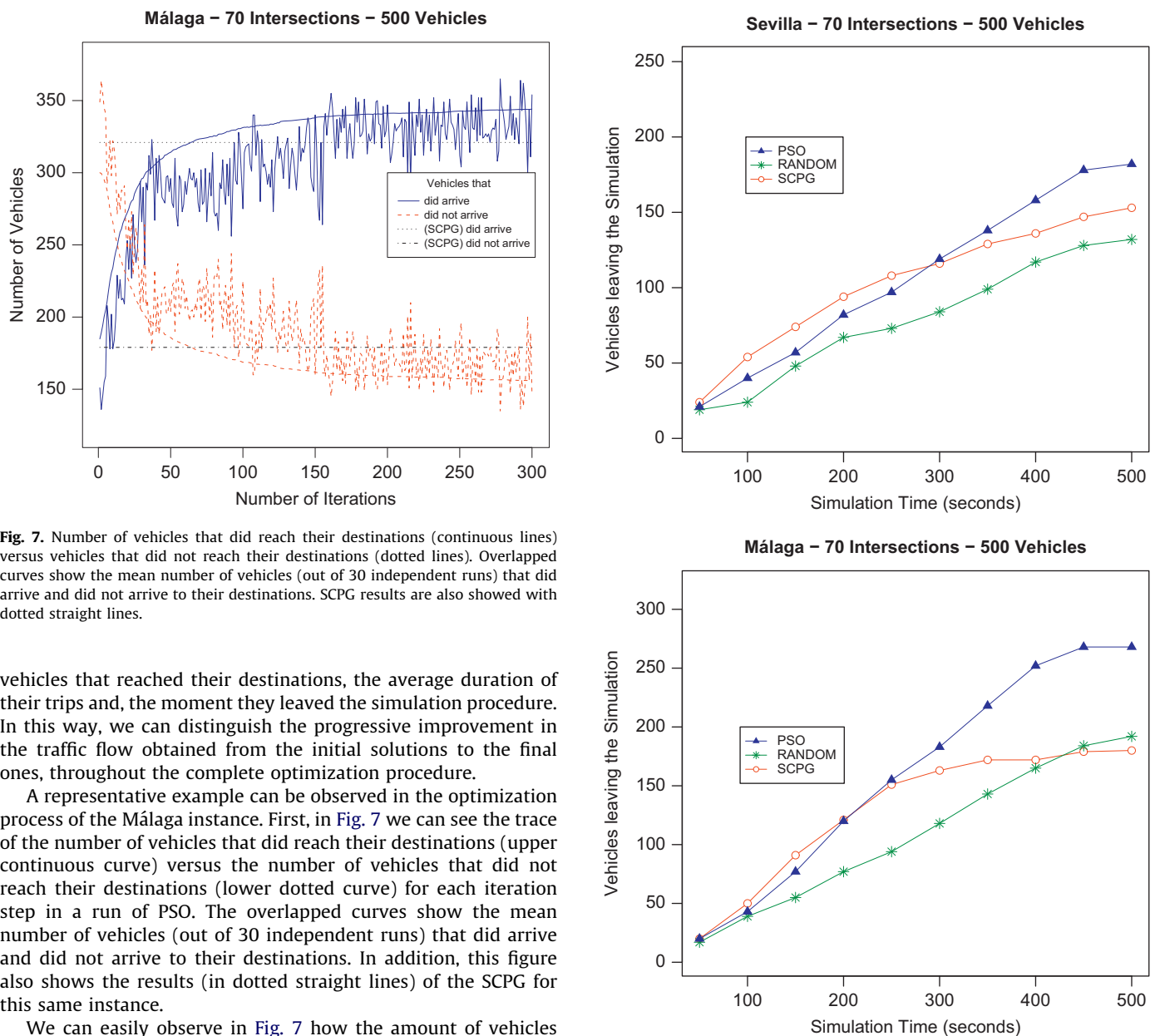


Fig. 7. Number of vehicles that did reach their destinations (continuous lines) versus vehicles that did not reach their destinations (dotted lines). Overlapped curves show the mean number of vehicles (out of 30 independent runs) that did arrive and did not arrive to their destinations. SCPG results are also showed with dotted straight lines.

vehicles that reached their destinations, the average duration of their trips and, the moment they leaved the simulation procedure. In this way, we can distinguish the progressive improvement in the traffic flow obtained from the initial solutions to the final ones, throughout the complete optimization procedure.

A representative example can be observed in the optimization process of the Málaga instance. First, in Fig. 7 we can see the trace of the number of vehicles that did reach their destinations (upper continuous curve) versus the number of vehicles that did not reach their destinations (lower dotted curve) for each iteration step in a run of PSO. The overlapped curves show the mean number of vehicles (out of 30 independent runs) that did arrive and did not arrive to their destinations. In addition, this figure also shows the results (in dotted straight lines) of the SCPG for this same instance.

We can easily observe in Fig. 7 how the amount of vehicles that did arrive (did not arrive) to their destinations increases (decreases) as the algorithm gets the stop condition of 300 iterations. In fact, at the initial steps of the optimization process, the number of vehicles that reached their destinations was lower

Fig. 8. Evolution of the number of vehicles leaving the scenario during the simulation of the best cycle programs (solutions) found in one representative independent run of PSO and RANDOM, and SCPG for the Sevilla and Málaga instances.

than the ones resulting in the cycle program generated by SCPG. However, at the final steps of the PSO procedure, the evolved solutions show a high quality in terms of the traffic flow, since 349 vehicles of the initial 500 (69.8%) finalized their trips successfully. Moreover, a mean number of 340 vehicles completed their trips in the final solutions of PSO (average of 30 runs). This contrasts to the 314 vehicles that reached their destinations in the SUMO cycle program: the improvement obtained by our PSO over SCPG is 10.13%.

Following this analysis, we now compare the number of vehicles that leave the traffic network (reach their destinations) during the simulation of the final solutions. Then, Fig. 8 plots the evolution of the number of vehicles that leave the network (in Sevilla and Málaga) during the simulation of the best cycle programs found with PSO, RANDOM, and SCPG.

Similar traces can be observed in Fig. 8 for the two instances: at the initial steps of the simulation, PSO and SCPG deliver quite similar number of vehicles, even showing SCPG a better performance over PSO. However, during the second half of the simulation, the cycle programs of PSO get a higher number of cars that reach their destinations. The greedy SCPG cycle program deteriorates with the simulation time, even performing worse than

RANDOM for the Málaga instance. Evidently, the generation of traffic jams during the simulation procedure directly influences on the ability of cycle programs to manage them. Nevertheless, the learning process performed by PSO leads the generation of cycle programs able to avoid large queues of cars, hence improving the global traffic flow even at the final steps of the simulation when the possibility of congestion is higher.

From a different point of view, Fig. 9 plots the trace of the average trip time employed by the vehicles in the resulted solutions of PSO through all the iterations of an example run. In this case, the trip time becomes shorter as the algorithm approaches the stop condition. We must notice that, in the calculation of the trip time, the vehicles that did not arrive to their destinations took 500 s, the complete simulation time. For this reason, SCPG solutions showed an averaged trip time of 660 s while PSO solutions obtained a trip duration of 557 s, which represents an improvement of 15.7% respect to the SCPG solution.

Finally, with the aim of better understanding the final implications of using (or not using) an optimized cycle program, Fig. 10 shows the simulation traces of the traffic flow resulted from solutions generated by both, SCPG (left) and PSO (right). The pictures were captured at the final of the simulation time, and correspond to two simulation procedures of a selected area of the Málaga instance including: Andalucía avenue, Aurora avenue, and Guadalmedina street. As we can observe, the traffic density of the SCPG cycle program is clearly higher than the one of PSO, even showing the former several intersections with traffic jams. As to the PSO cycle program, all intersections are unblocked at the end of the study.

5. Conclusions

Based on PSO, we have proposed in this work an optimization technique able to find successful cycle programs of traffic lights. For this study we have used two extensive traffic networks located in the metropolitan center of the cities of Sevilla and Málaga (in Spain).

A series of analysis have been carried out from different points of view: the performance of the optimization technique and the quality of solutions in the domain of the traffic lights. From these, the following conclusions are extracted:

1. The proposed PSO shows a successful performance in big realistic traffic scenarios. For the two instances, our proposal obtained results statistically better than the two other compared algorithms: the SUMO cycle programs generator (SCPG) and a Random Search algorithm (RANDOM).
2. The final solutions obtained by PSO can improve the number of vehicles that reach their destinations and the mean trip time,

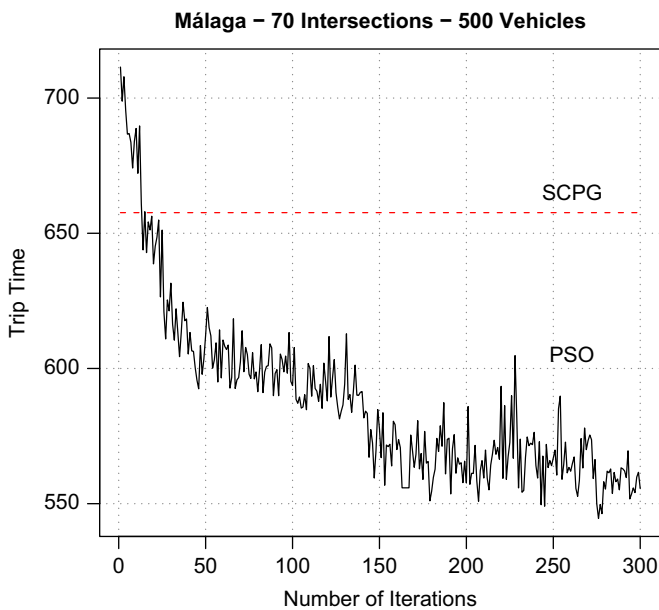


Fig. 9. Mean trip time of vehicles calculated for each one of the simulations performed through a representative run of PSO. SCPG results are also showed with a dotted straight line. Y axis represents the trip time in seconds.

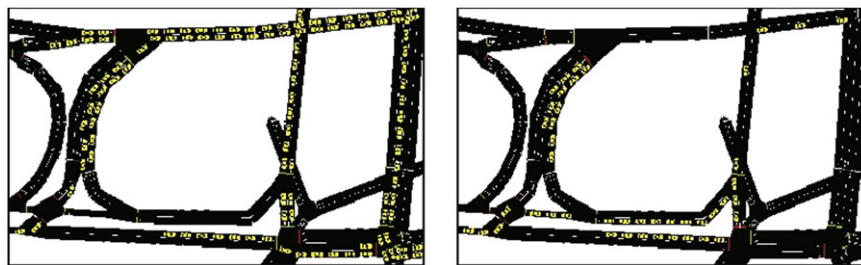


Fig. 10. Simulation traces of the traffic flow (cars in white) resulting from the cycle programs generated by both, SCPG (left) and PSO (right) in Málaga. The pictures show snapshots at the end of the simulation time. The reader can notice that the SCPG leaves a dense traffic while PSO has cleaned the routes and the traffic is very fluid and sparse.

for the two instances. In particular, for Málaga, the improvement obtained is around 10.13% in the number of completed trips and 15.7% in the trip time, regarding SCPG.

3. Moreover, an analysis of the traffic flow shows that the cycle traffic light schedules provided by our PSO improve the global traffic flow in the road network. All this means an actual improvement in traffic for citizens.

As future work, with the final aim of assisting the human expert in the decision making process, we will be tackling the optimal cycle program with current sophisticated versions of PSO (like Standards 2007 and 2011), other metaheuristic algorithms, and other optimization strategies like multiobjective or parallel versions. We are also interested in using other traffic simulators and create new large dimension instances, as close as possible to real scenarios of a whole city.

Acknowledgments

Authors acknowledge funds from the Spanish Ministry MICINN and FEDER Contract TIN2008-06491-C04-01 (M* <http://mstar.lcc.uma.es>) and CICE, Junta de Andalucía, Contract P07-TIC-03044 (DIRICOM <http://diricom.lcc.uma.es>). José García-Nieto is supported by Grant BES-2009-018767 from the MICINN. A. Carolina Olivera is supported by a post-doctoral fellowship from the CONICET of Argentina (www.conicet.gov.ar).

References

- Alba, E., García-Nieto, J., Jourdan, L., Talbi, E.-G., 2007a. Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, 284–290.
- Alba, E., Luque, G., García-Nieto, J., Ordóñez, G., Leguizamón, G., 2007b. Mallba: a software library to design efficient optimisation algorithms. *Int. J. Innovative Comput. Appl. (IJICA)* 1 (1), 74–85.
- Alba, E., García-Nieto, J., Taheri, J., Zomaya, A., 2008. New research in nature inspired algorithms for mobility management in GSM networks. In: *Lecture Notes in Computer Science of the Fifth European Workshop on the Application of Nature-inspired Techniques to Telecommunication Networks and Other Connected Systems, EvoWorkshops08*, Napoli, Italy, pp. 1–10.
- Angulo, E., Romero, F.P., García, R., Serrano-Guerrero, J., Olivas, J.A., 2008. A methodology for the automatic regulation of intersections in real time using soft-computing techniques. In: *Modelling, Computation and Optimization in Information Systems and Management Sciences*. Springer, pp. 379–388.
- Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput. Surveys (CSUR)* 35 (3), 268–308.
- Brockfeld, E., Barlovic, R., Schadschneider, A., Schreckenberg, M., 2001. Optimizing traffic lights in a cellular automaton model for city traffic. *Phys. Rev. E* 64 (5), 056132.
- Chen, J., Xu, L., 2006. Road-junction traffic signal timing optimization by an adaptive particle swarm algorithm. In: *ICARCV*, pp. 1–7.
- Clerc, M., Kennedy, J., 2002. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* 6 (1), 58–73.
- García-Nieto, J., Alba, E., 2010. Automatic parameter tuning with metaheuristics of the AODV routing protocol for vehicular ad-hoc networks. *Applications of Evolutionary Computation of Lecture Notes in Computer Science*, vol. 6025. Springer, Berlin, Heidelberg, pp. 21–30.
- García-Nieto, J., Toutouh, J., Alba, E., 2010. Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics. *Engineering Applications of Artificial Intelligence Advances in Metaheuristics for Hard Optimization New Trends and Case Studies* 23 (5), 795–805.
- Hewage, K.N., Ruwanpura, J.Y., 2004. Optimization of traffic signal light timing using simulation. In: *WSC '04: Proceedings of the 36th Conference on Winter Simulation*, Winter Simulation Conference, pp. 1428–1436.
- Holm, P., Tomich, D., Sloboden, J., Lowrance, C., 2007. Traffic analysis toolbox volume iv: guidelines for applying corsim microsimulation modeling software. Technical Report, National Technical Information Service—5285 Port Royal Road Springfield, VA 22161, USA—Final Report.
- Kachroudi, S., Bhouiri, N., 2009. A multimodal traffic responsive strategy using particle swarm optimization. In: *Proceedings of the 12th IFAC Symposium on Transportation Systems*, Redondo Beach, CA, USA, pp. 531–537.
- Karakuzu, C., Demirci, O., 2010. Fuzzy logic based smart traffic light simulator design and hardware implementation. *Appl. Soft Comput.* 10 (1), 66–73.
- Kennedy, J., Eberhart, R.C., 2001. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California.
- Krajzewicz, D., Brockfeld, E., Mikat, J., Ringel, J., Rössel, C., Tuchscheerer, W., Wagner, P., Wöslar, R., 2005. Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO.
- Krajzewicz, D., Bonert, M., Wagner, P., 2006. The open source traffic simulation package SUMO. RoboCup 2006 Infrastructure Simulation Competition.
- Krauß, S., 1998. Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. Ph.D. Thesis.
- Leung, J., Kelly, L., Anderson, J.H., 2004. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. CRC Press Inc., Boca Raton, FL, USA.
- Lim, G., Kang, J.J., Hong, Y., 2001. The optimization of traffic signal light using artificial intelligence. In: *FUZZ-IEEE*, pp. 1279–1282.
- McCrea, J., Moutari, S., 2010. A hybrid macroscopic-based model for traffic flow in road networks. *Eur. J. Oper. Res.* 207 (1), 676–684.
- Montes de Oca, M.A., Stützle, T., Birattari, M., Dorigo, M., 2009. Frankenstein's PSO: a composite particle swarm optimization algorithm. *Trans. Evol. Comput.* 13 (5), 1120–1132.
- Nagatani, T., 2010. Effect of speed fluctuation on green-light path in 2d traffic network controlled by signals. *Phys. A: Stat. Mech. Appl.* 389 (19), 4105–4115.
- Parsopoulos, K.E., Vrahatis, F.M.N., 2005. Unified particle swarm optimization for solving constrained engineering optimization problems. In: *Advances in Natural Computation*. Springer, pp. 582–591.
- Peng, L., Hai Wang, M., Ping Du, J., Luo, G., 2009. Isolation niches particle swarm optimization applied to traffic lights controlling. In: *Proceedings of the 48th IEEE: Decision and Control Chinese Conference CDC/CCC*, pp. 3318–3322.
- Rouphail, N.M., Park, B.B., Sacks, J., 2000. Direct signal timing optimization: strategy development and results. Technical Report. In: *XI Pan American Conference in Traffic and Transportation Engineering*.
- Sánchez Medina, J., Galán, M., Rubio Royo, E., 2005. Stochastic vs deterministic traffic simulator: comparative study for its use within a traffic light cycles optimization architecture. *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach of Lecture Notes in Computer Science*, vol. 3562. Springer, Berlin, Heidelberg, pp. 622–631.
- Sánchez, J., Galán, M., Rubio, E., 2008. Applying a traffic lights evolutionary optimization technique to a real case: Las Ramblas area in Santa Cruz de Tenerife. *IEEE Trans. Evol. Comput.* 12 (1), 25–40.
- Sheskin, D.J., 2007. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC.
- Spall, J.C., Chin, D.C., 1997. Traffic-responsive signal timing for system-wide traffic control. *Transport. Res. Part C: Emerging Technol.* 5 (3–4), 153–163.
- Teklu, F., Sumalee, A., Watling, D., 2007. A genetic algorithm approach for optimizing traffic control signals considering routing. *Comput. Aided Civil Infrastruct. Eng.* 22, 31–43.
- Thain, D., Tannenbaum, T., Livny, M., 2005. Distributed computing in practice: the condor experience. *Concurrency—Prac. Exp.* 17 (2–4), 323–356.
- Tolba, C., Lefebvre, D., Thomas, P., El Moudni, A., 2005. Continuous and timed petri nets for the macroscopic and microscopic traffic flow modelling. *Simul. Modelling Pract. Theory* 13 (5), 407–436.
- Turky, A.M., Ahmad, M.S., Yusoff, M.Z., Hammad, B.T., 2009. Using genetic algorithm for traffic light control system with a pedestrian crossing. *RSKT '09: Proceedings of the Fourth International Conference on Rough Sets and Knowledge Technology*. Berlin, Heidelberg, pp. 512–519.
- Van Vliet, D., 1982. SATURN—a modern assignment model. *Traffic Eng. Control* 23, 578–581.