

Enhancing the Urban Road Traffic with Swarm Intelligence: A Case Study of Córdoba City Downtown

José García-Nieto and Enrique Alba
Dept. Lenguajes y Ciencias de la Computación
University of Málaga, UMA
Campus de Teatinos, 29071, Málaga, Spain
Email: jnieto@lcc.uma.es, eat@lcc.uma.es

Ana Carolina Olivera
Dept. de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, UNS
Av. Alem 1253, 8000, Bahía Blanca, Argentina
Email: aco@cs.uns.edu.ar

Abstract—In current modern cities, the increasing number of traffic lights that control the vehicular traffic flow requires a highly complex scheduling. Thousands of red lights, that have to be optimally programmed, are nowadays operating in congested urban areas. Therefore, automatic intelligent systems are indispensable tools for optimally tackling this task. In this work, we propose a Swarm Intelligence approach that, coupled with the SUMO traffic simulator, is able to find successful cycle programs of traffic lights for large urban areas. In concrete, we have focused on a metropolitan area of the city downtown of Córdoba (in Spain). The experiments and comparisons with other techniques reveal that our proposed approach obtains significant profits in terms of traffic flow and global trip time.

Keywords—Traffic Light Scheduling, Particle Swarm Optimization, SUMO Microscopic Simulator, Differential Evolution.

I. INTRODUCTION

Nowadays, the high vehicular traffic in modern cities provokes severe problems related to the pollution, the congestion, the security, and the availability of parking places. Since changes in the urban area infrastructure are always expensive and 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 [9], [13]. The number of operating traffic lights consequently grows up and their joint scheduling becomes complex due to the huge number of combinations that appear. Therefore, the use of automatic intelligent systems for the optimal cycle programming of traffic lights is a mandatory task in the traffic network design.

Recent initiatives are focused in the use of simulators [6], [9], [13], since they provide an immediate and continuous source of information about the traffic flow. In the last few years, the main efforts were directed towards programming of convenient cycles of traffic lights [3], [10].

In this sense, the use of intelligent methods, and in concrete the use of metaheuristic algorithms [1], [2], have demonstrated their usefulness to the scheduling of traffic lights [5], [13], [11], [14]. However, the use of such an intelligent systems in the literature has been restricted to

optimize academic instances, or they are specifically adapted to limited areas in urban scenarios with a few cars and traffic lights. In addition, they were not compared against other different techniques nor human expert solutions at all.

All these motivated us to propose an optimization strategy by means of which a given metaheuristic algorithm can find successful cycle programs of traffic lights for heterogeneous and large urban instances. In particular, we have focused in this work on a Particle Swarm Optimization (PSO) algorithm [7] as optimizer for the traffic lights' cycle programs. The SUMO (Simulator of Urban Mobility) [8] microscopic simulator is used 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 given scenario instance. In this work, we have focused on enhancing the traffic flow in a metropolitan area of the city downtown of Córdoba (in Spain). The experiments and comparisons with other techniques: Differential Evolution (DE) [12], Random Search (RANDOM), and Sumo Cycle Program Generator (SCPG) [8], reveal that our proposed PSO obtains significant profits in terms of traffic flow and global trip time.

The remaining of this article is organized as follows. The Section II is devoted to formulate problem of the optimal cycle programs of traffic lights. In Section III, our optimization approach is described. Sections IV and V presents the experimental methodology and the results obtained, respectively. Conclusions and future work are given in Section VI.

II. PROBLEM FORMULATION: OPTIMAL CYCLE PROGRAMS 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, 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 can not 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.

A. Solution Encoding

For our approach, the OCP is encoded by means of a vector of integers following the SUMO structure of programming cycles [8], 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 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.

B. Fitness Function

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

$$fitness = \frac{TT + SW + (NV * ST)}{V^2 + P} \quad (1)$$

The main objective (Equation 1) 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 are finally wasting all the simulation time, which implies an additional penalization. Another 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.

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 through them. The proportion of colors in each phase (ph) of all the intersections tl is formulated in Equation 2.

$$P = \sum_{k=0}^{tl} \sum_{j=0}^{ph} s_{k,j} * \left(\frac{G_{k,j}}{r_{k,j}} \right), \quad (2)$$

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

III. OPTIMIZATION STRATEGY FOR OCP

The optimization strategy is composed by two main parts: the Particle Swarm Optimizer (PSO), and the simulation procedure with the SUMO traffic microsimulator.

The PSO algorithm [7] is a population-based metaheuristic inspired by the social behavior of birds within a flock, and was 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*. We have followed the specification of the Standard PSO 2007 [4]. In this algorithm, each particle position x^i is updated each iteration g by means of the Equation 3.

$$x_{g+1}^i = x_g^i + v_{g+1}^i \quad (3)$$

where term v_{g+1}^i is the velocity of the particle, given by the Equation 4.

$$v_{g+1}^i = w \cdot v_g^i + U[0, \varphi_1] \cdot (p_g^i - x_g^i) + U[0, \varphi_2] \cdot (b_g^n - x_g^i) \quad (4)$$

In this formula, p_g^i is the best solution that the particle i has seen so far, b_g^n is the best particle of a neighborhood of n other particles (also known as the *social best*) randomly (uniform) selected from the swarm, 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 the acceleration coefficients that control the relative effect of the personal and social best particles, while $U[0, \varphi_k]$ is a uniform random value in $[0, \varphi_k]$, $k \in 1, 2$ which is sampled anew for each component of the velocity vector.

Since the OCP requires solutions encoded with a vector of integers (representing cycle phase durations), we have used the *quantisation* method provided in the standard specification of PSO 2007 [4]. This quantisation is applied to each new generated particle (in Equation 3), and transforms the continuous values of particles to discrete ones. It consist on a Mid-Thread uniform quantiser method as specified in Equation 5. The quantum step is set to $\Delta = 1$.

$$Q(x) = \Delta \cdot \lfloor x/\Delta + 0.5 \rfloor \quad (5)$$

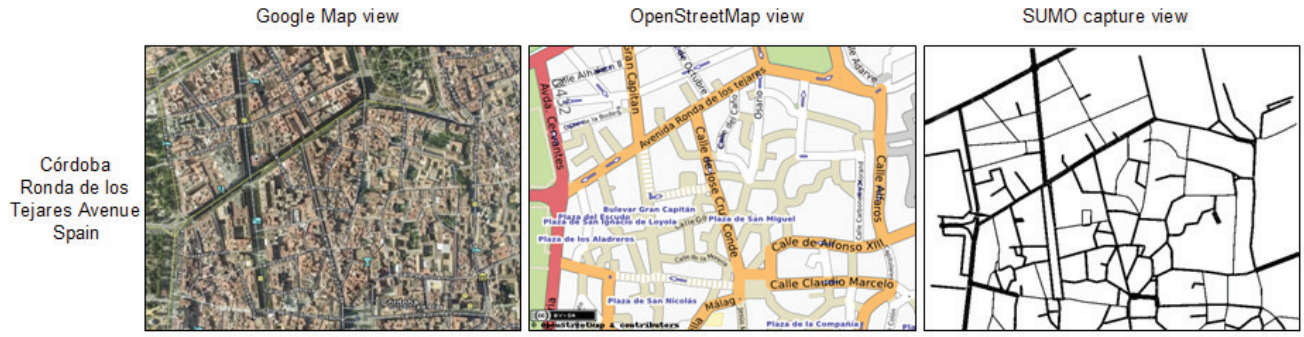


Figure 1. Process of creation of real-world instances for study. Around Ronda de los Tejares Avenue (37°53'15"N 4°46'57"E) instance view. After selecting our area of interest (Google Earth view), it is interpreted by means of the OpenStreetMap tool, and then exported to SUMO format

Algorithm 1 Pseudocode of Standard PSO 2007 for OCP

```

1: initializeSwarm()
2: while  $g < \text{maxIterations}$  do
3:   for each particle  $x_g^i$  do
4:      $b_g^n = \text{bestNeighbourSelection}(x_g^i, n)$ 
5:      $v_{g+1}^i = \text{updateVelocity}(w, v_g^i, x_g^i, \varphi_1, p_g, \varphi_2, b_g^n)$  //Eq. 4
6:      $x_{g+1}^i = Q(\text{updatePosition}(x_g^i, v_{g+1}^i))$  //Eqs. 3 and 5
7:      $\text{evaluate}(x_{g+1}^i)$  //SUMO Simulation and Eq. 1
8:      $p_{g+1}^i = \text{update}(p_g^i)$ 
9:   end for
10: end while

```

Algorithm 1 describes the pseudo-code of the Standard PSO 2007 for OCP. The algorithm starts by initializing the swarm (Line 1). The corresponding elements of each particle (solutions) are initialized with random values representing the phase durations. These values are within the time interval $[5, 60] \in \mathbb{Z}^+$, 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). Then, for a maximum number of iterations, each particle *flies* through the search space updating its velocity and position (Lines 4, 5, and 6), it is then evaluated (Line 7), and its personal best position p^i is also updated (Line 8). Finally, the best particle found so far is returned.

The simulation procedure is then used for 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 compute the required values in Equation 1.

When the strategy 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 (Line 7 at

Algorithm 1) requires only one simulation procedure since vehicle routes in SUMO are generated deterministically. Each new cycle program is statically loaded for each simulation procedure. 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.

IV. EXPERIMENTAL FRAMEWORK

As we are interested in developing an optimization solver capable of dealing with close-to-reality and generic urban areas, we have generated a scenario by extracting actual information from real digital maps, as illustrated in Figure 1. This scenario covers approximately 0.75 km^2 of the Córdoba city downtown (Spain), comprising: Ronda de los Tejares Avenue, Alfaro Street, Claudio Marcelo Street and Cervantes Avenue. It is composed of 349 intersections between streets and roundabouts, of which, only 30 intersections were equipped with a number of traffic lights between 4 and 16, and constituting a global number of 152 traffic lights (solution dimension).

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 (a maximum in urban areas). The routes were previously generated by following random paths. The simulation time was set to 500 seconds (iterations of microsimulation). 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. In order to study the scalability of our approach in terms of traffic density, we have set three different numbers of vehicles in the scenario with: 100, 300, and 500 circulating vehicles, thus resulting in three different scenario instances.

Regarding the experimental setup, we have used the implementation of the Standard PSO 2007 provided by MALLBA [1], a C++ based framework of metaheuristic algorithms for solving optimization problems.

Table I
PSO AND DE PARAMETERS

Solver	Parameter	Value
PSO	Swarm Size	100
	Particle Size (N. Traffic Lights)	152
	Local and Social Coefficients ($\varphi_1 = \varphi_2$)	2.05
	Neighborhood size (n)	3
	Inertia Weight (w)	0.7213
DE	Population Size	100
	Individual Size (N. Traffic Lights)	152
	Mutation Constant (F)	0.5
	Crossover Probability (Cr)	0.9

For the sake of a fair comparison, we also experimented with another metaheuristic algorithm, the Differential Evolution (DE) [12], and with a Random Search Algorithm (RANDOM). These two also implemented in the scope of the MALLBA library. The DE is a distance-vector optimization algorithm with an internal operation close to the one of PSO. In this work, the DE was also provided with the quantisation *Mid-Thread* method (as used in Standard PSO 2007) for dealing with integer values in the solution codification. Thus, performing the same experimentation procedure, we expect to obtain some insights into the power of our proposal (how much intelligent it is) regarding a technique without any heuristic information in its operation (RANDOM), and with regards to another metaheuristic (DE).

For PSO, DE, and RANDOM, we have carried out 30 independent runs. The swarm (population) size was set to 100 particles (individuals) performing 200 iteration steps, hence resulting a number of 20,000 solution evaluations (SUMO simulations) per run and instance. The Random Search Algorithm also performs 20,000 solution evaluations. The remaining parameters are summarized in Table I.

The simulation phase is carried out by executing (in the evaluation of solutions) 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, 1GB RAM, and Linux Debian O.S. They operate under a Condor middleware platform that acts as a distributed task scheduler (each task dealing with one independent run).

Finally, SUMO provides a deterministic algorithm for generating cycle programs (SUMO Cycle Program Generator - 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 intersections 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 SCPG can be found in [8].

Table II
MAXIMUM, MEDIAN, MINIMUM, AND MEAN FITNESS OBTAINED BY PSO FOR CÓRDOBA INSTANCE. THE FITNESS VALUES OBTAINED BY DE, RANDOM, AND SCPG ALGORITHMS ARE ALSO PROVIDED

Num. of Vehicles	Value	PSO	DE	RANDOM	SCPG
100	Max.	1.38E+00	1.57E+00	1.96E+00	1.23E+00
	Med.	1.09E+00	1.36E+00	1.87E+00	1.23E+00
	Min.	9.10E-01	1.19E+00	1.76E+00	1.23E+00
	Mea.	1.10E+00	1.36E+00	1.87E+00	1.23E+00
	Std.	1.12E-01	9.67E-02	5.61E-02	0.00E+00
300	Max.	1.53E+00	2.24E+00	2.05E+00	1.22E+00
	Med.	1.16E+00	1.38E+00	1.95E+00	1.22E+00
	Min.	9.33E-01	1.19E+00	1.80E+00	1.22E+00
	Mea.	1.17E+00	1.46E+00	1.94E+00	1.22E+00
	Std.	1.51E-01	2.82E-01	6.98E-02	0.00E+00
500	Max.	1.78E+00	3.64E+00	3.08E+00	2.17E+00
	Med.	1.42E+00	3.14E+00	2.83E+00	2.17E+00
	Min.	1.09E+00	1.51E+00	2.40E+00	2.17E+00
	Mea.	1.45E+00	2.75E+00	2.83E+00	2.17E+00
	Std.	1.65E-01	7.62E-01	1.44E-01	0.00E+00

V. RESULTS AND COMPARISONS

In this section, a performance analysis and a statistical comparison of the studied algorithms are carried out. An analysis concerning the obtained cycle programs is also performed from the point of view of the problem domain.

A. Performance Analysis of Algorithms

Table II contains the maximum (Max.), median (Med.), mean (Mea.), minimum (Min.), and standard deviation (Std.) of the distributions of fitness resulted for all the compared algorithms: PSO, DE, RANDOM, and SCPG. These results were obtained for the Córdoba city instance with traffic densities: 100, 300, and 500 vehicles. We can observe in this table that PSO obtains the best results (marked in bold) in general, followed by DE and SCPG. In fact, the improvement of PSO with regards to the other three algorithms increases with the density of the traffic. It can be graphically observed in Figure 2, where the boxplots of the distribution fitness of PSO, DE and RANDOM are plotted. The results of SCPG are represented with a ♦ point since this technique always returns the same deterministic result for a given instance. The distributions of PSO always show better median values than the best values plotted by the other techniques. Specifically for density 500 vehicles (the largest instance), the lower quartile of PSO is better than the minimum value of DE and SCPG. It is worth checking that RANDOM always obtained the worst distribution values, but for the median fitness of DE for density 500.

Furthermore, with the aim of providing these comparisons with statistical significance, we have applied a *Signed Ranked (Wilcoxon)* test [15] to the numerical distributions of the results. We have used this non-parametric test since the resulting distributions usually violate the condition of normality required to apply parametric tests (*Kolmogorov-Smirnov p-value* < 0.023). The confidence level was set to

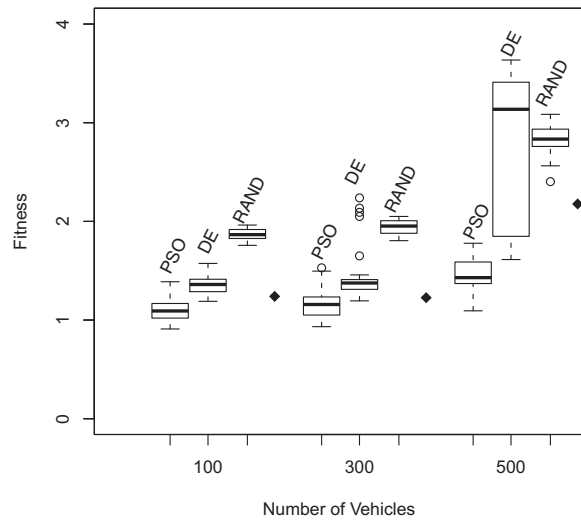


Figure 2. Boxplot representation of distributions of the results for the Córdoba instance with 100, 300, and 500 vehicles. The results of SCPG are represented with a \blacklozenge point since this technique always returns the same deterministic result for a given instance

95% ($\alpha=0.05$), which allows us to ensure that all these results are statistically different if they result in $p\text{-value} < 0.05$.

In effect, for all the instances (100, 300, and 500 traffic densities) and for all the pair-wise *Signed Ranked* comparisons (PSO vs. DE, PSO vs. RANDOM, and so on), the differences between the distributions resulted with $p\text{-values} \ll 0.05$, except for DE vs. RANDOM (500 vehicles) with $p\text{-value} = 0.285$. Therefore, we can claim that our PSO obtained statistically better results than the other three compared algorithms: DE (evolutionary algorithm), RANDOM (random stochastic search) and SCPG (deterministic). This also means that our algorithm is intelligent and competent when compared to evolutionary algorithms and greedy information with human knowledge, respectively.

B. Analysis of the Obtained Cycle Programs

Finally, in this section we focused on the cycle programs obtained as solutions by our PSO, and the possible profits they can offer to the actual users in this field.

Figure 3 plots the trace of the mean trip time employed by the 500 vehicles in the resulted solutions of PSO through all its 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 seconds, the complete simulation time. For this reason, SCPG solutions showed a trip time of 848 seconds while PSO solutions obtained a minimum trip duration of 700 seconds, which involves an improvement of 17,45% respect to the SCPG solution.

30 Traffic Logics - 500 Vehicles

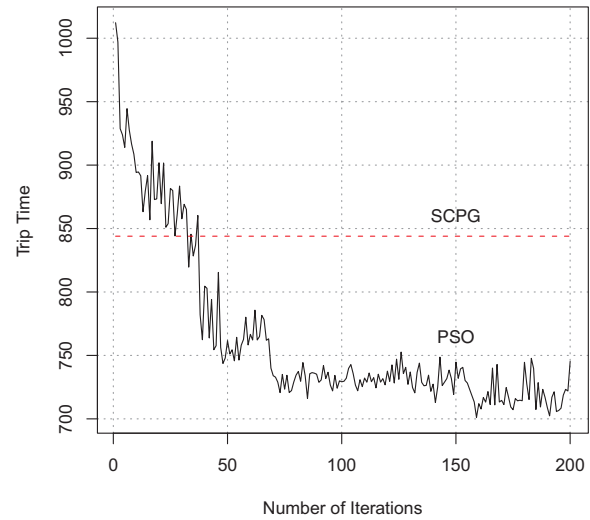


Figure 3. Mean trip time of vehicles to complete their respective routes. These values have been calculated for each simulation performed through a representative run of PSO. Dotted straight line represents SCPG results. Y axis shows the trip time in seconds

From a visual perspective, and with the aim of better understanding the final implications of using (or not using) an optimized cycle program, Figure 4 shows the simulation traces of the traffic flow resulted from solutions generated by both, SCPG (right) and PSO (left). The pictures were captured at the end of the simulation time (500 seconds), and correspond to two simulation procedures of the scenario instance with 500 vehicles. The particular area corresponds to the intersection of Góngora with Cruz Conde streets, a usual congested point in Córdoba. 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. The result of using our PSO cycle program is that all intersections are conveniently unblocked at the end of the study.

VI. CONCLUSIONS

In this work, we propose a Swarm Intelligence approach that, coupled with the SUMO traffic simulator, it is able to find successful cycle programs of traffic lights for large urban areas. In concrete, we have focused on a metropolitan area of the city downtown of Córdoba (in Spain). A series of experiments and comparisons have been carried out with the aim of checking how efficient our PSO performs. The main conclusions that can be extracted are as follows:

- 1) Our optimization structure with PSO performed efficiently for the studied instances. In comparison with DE, RANDOM, and SCPG, our PSO showed the best performance, with statistical confidence.
- 2) In the scope of the scenario studied here, we can claim that our PSO scales adequately in terms of the density

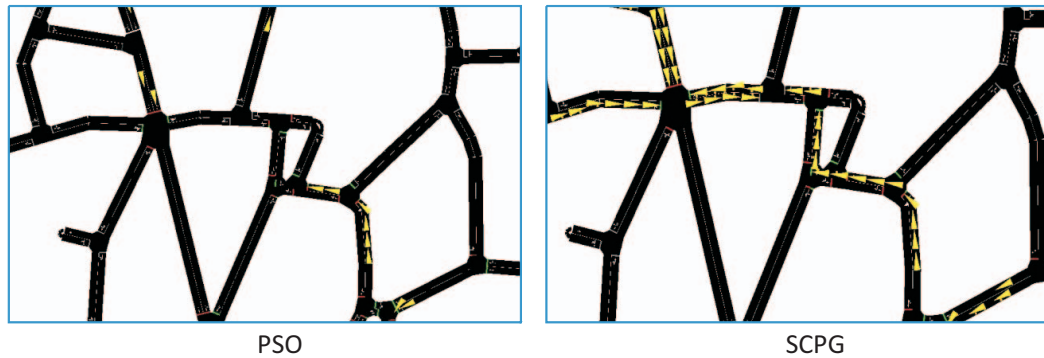


Figure 4. Snapshots at the end of the simulations resulting from cycle programs generated by PSO and SCPG. Triangles are cars not arriving at their destinations. It is noticeable that the SCPG leaves a dense traffic while PSO has solved all the traffic jams. This particular area corresponds to the intersection of Góngora with Cruz Conde streets

of traffic: with 100, 300, and 500 vehicles.

- 3) The final solutions obtained by our PSO can improve the number of vehicles that reach their destinations and the mean trip time, for all the instances. In particular, with 300 vehicles, the improvement obtained is around 17.45% in the trip time, regarding SCPG. All this means an actual improvement in city traffic.

As future work, we will be tackling the optimal cycle program with other metaheuristic algorithms. We are also interested in creating new larger dimension instances, as close as possible to real scenarios of a whole city.

ACKNOWLEDGMENT

Authors acknowledge funds from the Junta de Andalucía (CICE), under contract P07-TIC-03044 (DIRICOM project <http://diricom.lcc.uma.es>) and Spanish Ministry of Sciences and Innovation (MICINN) and FEDER under contract TIN2008-06491-C04-01 (M* project <http://mstar.lcc.uma.es>). José García-Nieto is supported by grant BES-2009-018767 from the MICINN. Ana Carolina Olivera is supported by a post-doctoral fellowship from the Consejo Nacional de Investigaciones Científicas y Técnicas de Argentina (CONICET, www.conicet.gov.ar).

REFERENCES

- [1] E. Alba, G. Luque, J. García-Nieto, G. Ordonez, and G. Leguizamón. Mallba: a software library to design efficient optimisation algorithms. *Int. J. of Innovative Computing and Applications 2007 (IJICA)*, 1(1):74–85, 2007.
- [2] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [3] E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg. Optimizing traffic lights in a cellular automaton model for city traffic. *Phys. Rev. E*, 64(5):56–132, 2001.
- [4] M. Clerc et al. Standard PSO 2011. Technical Report [online] <http://www.particleswarm.info/>, Particle Swarm Central, January 2011.
- [5] J. García-Nieto, E. Alba, and A. C. Olivera. Swarm intelligence approach for the traffic light scheduling: Application to real urban areas. *Engineering Applications Of Artificial Intelligence*, -In Press, 2011.
- [6] C. Karakuzu and O. Demirci. Fuzzy logic based smart traffic light simulator design and hardware implementation. *Applied Soft Computing*, 10(1):66 – 73, 2010.
- [7] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.
- [8] D. Krajzewicz, M. Bonert, and P. Wagner. The open source traffic simulation package SUMO. *RoboCup 2006 Infrastructure Simulation Competition*, 2006.
- [9] J. McCrea and S. Moutari. A hybrid macroscopic-based model for traffic flow in road networks. *European Journal of Operational Research*, 207(2):676 – 684, 2010.
- [10] T. Nagatani. Effect of speed fluctuations on a green-light path in a 2d traffic network controlled by signals. *Physica A: Statistical Mechanics and its Applications*, 389(19):4105 – 4115, 2010.
- [11] L. Peng, M. Wang, J. Du, and G. Luo. Isolation niches particle swarm optimization applied to traffic lights controlling. In *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3318 –3322, dec. 2009.
- [12] K. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing. Springer, 2005.
- [13] J. Sánchez, M. Galán, and E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: “Las Ramblas” area in Santa Cruz de Tenerife. *IEEE Transactions on Evolutionary Computation*, 12(1):25 –40, feb. 2008.
- [14] F. Teklu, A. Sumalee, and D. Watling. A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering*, 22:31–43, 2007.
- [15] R. Wilcox. *New statistical procedures for the social sciences*. Hillsdale, 1987.