Real-time Traffic Signal Control with Dynamic Evolutionary Computation

Kai Zeng, Yue-Jiao Gong (Corresponding Author) and Jun Zhang
Department of Computer Science, Sun Yat-sen University
Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
Key Laboratory of Software Technology, Education Department of Guangdong Province, P.R. China
gongyuejiao@gmail.com

***Abstract*—Nowadays real-time traffic signal control is a crucial issue with potential benefits in the fields of traffic control, environmental pollution, and energy utilization. In the literature, few related studies have been done with dynamic evolutionary algorithms. In this paper, we proposed a strategy using Collaborative Evolutionary-Swarm Optimization (CESO), which is able to track time-varying optimal solutions effectively. We use the simulator of urban mobility (SUMO), a popular traffic simulator to generate traffic flows. A grid traffic network is designed with several scenarios to simulate changes of traffic flows captured by traffic monitors. We test different traffic changes in the network using the proposed strategy and compare its performance with a traditional evolutionary algorithm. Experimental results show that our algorithm can obtain promising configuration of traffic light cycles and reduce the average delay time of all vehicles in various scenarios.**

Keywords-dynamic algorithms, real-time traffic signal control, Collaborative Evolutionary-Swarm Optimization (CESO).

I. INTRODUCTION

Traffic congestion is a common problem around the world. Since it is serious and closely related to many other problems like pollution, energy dissipation, and public safety, different methods have been developed to solve it. Traffic signal control is a cost-effective method that can bring substantial reductions in traffic delay [1], [2]. Considering large traffic flows and the increasing number of intersections, the control of traffic lights becomes more and more complicated. As evolutionary computation performs well in solving complex problems, many researchers have utilized them to optimize cycle program of traffic lights [2]-[11].

In this regard, current research efforts can be divided into two categories. In the first category, models with different assumptions of traffic flows are proposed to simulate the real traffic scene [3]-[7]. Researchers incorporate related algorithms into their models to calculate cycle program of traffic lights. Although these methods can obtain reasonable results, building a model that is suitable to any given traffic flow is difficult and sometimes impossible due to the uncertainty and complexity of real-world situations. Therefore, these methods are limited in scope.

On the other hand, some researchers employ simulators to simplify their model design and automatically generate needed traffic flows for their experiments [8]-[11]. These methods are practical and yet economical, since real traffic tests involve high operational costs and human resources. In addition, traffic simulators provide visual interfaces of traffic flows for researchers. In general, researchers use previously gathered data of traffic flows or self-designed instances as input. Accordingly, several modified algorithms are adopted to generate feasible solutions under certain evaluation criteria. However, in most cases, the input only represents several typical situations such as AM or PM rush hours or specific traffic flow patterns, and many algorithms spend a long time to reach an acceptable solution. Since the number and the path of vehicles vary all the time, the current acceptable schedule may become inferior in the next time-period.

This paper employs dynamic evolutionary computation since the control of traffic lights can be seen as a dynamic optimization problem (DOP) where traffic conditions change over time. In recent years, many modified evolutionary algorithms have been applied to solve DOPs. Compared with traditional evolutionary computation, these algorithms show desirable performance in solving problems in dynamic environments. All above has motivated us to propose a strategy based on Collaborative Evolutionary-Swarm Optimization [12] to find changing optimal cycle program of traffic lights. The reasons why we use this algorithm are listed as follows. First, the algorithm is originally designed for dynamic environments. This is a desirable property for time-varying traffic scenarios. Second, this algorithm is easy to implement than several existing dynamic evolutionary algorithms [13], [14]. Third, the population tracking the global optimum is composed of a particle swarm complied with PSO rules, while PSO is a famous algorithm which has shown fast convergence to suitable solutions. Thus it can respond to real-time requirements of dynamic environments rapidly.

The rest of this paper is organized as follows. In Section II, related algorithms and configuration in SUMO are presented. In section III, the details of the proposed strategy are described. Section IV describes experimental settings and corresponding results. Finally, conclusions are drawn in section V.

II. RELATED WORK

A. Particle Swarm Optimization(PSO)

Particle swarm optimization is a well-known population-based algorithm [15], which was first intended for simulating the social behavior of a bird flock or fish school. The movements of the particles are guided by their own best known position as

This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China No.2013AA01A212, in part by the NSFC for Distinguished Young Scholars 61125205, in part by the NSFC No. 61332002 and No.61300044.

978-1-4799-4173-5/14 $31.00 © 2014 IEEE
DOI 10.1109/IIAI-AAI.2014.104

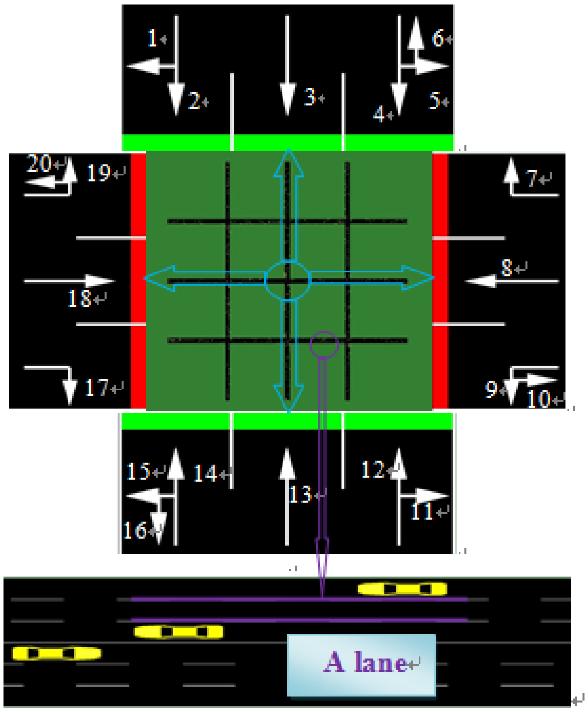


Fig. 1. The designed network and a set of traffic lights which are numbered 1-20 in an intersection.

well as the best known position of the entire swarm. In each generation, the jth dimension of each particle is updated according to the following equations:

$$x_i^j = x_i^j + v_i^j \tag{1}$$

$$v_i^j = \omega v_i^j + c_1 r_1 \left(p_i^j - x_i^j\right) + c_2 r_2 \left(g^j - x_i^j\right) \tag{2}$$

Where x_i is the position of particle i, ω is the inertia weight which controls the influence degree of the previous velocity to the current velocity. p_i is the best solution of particle i, g is the global best particle. r_1 and r_2 are random numbers uniformly distributed in [0, 1]. c_1 and c_2 are the acceleration coefficients that determine the influence of p_i and g to particle i, respectively.

B. *Differential Evolution(DE)*

DE [16] is population-based stochastic algorithm designed for global optimization. For each target vector $X_{i,\,G}$, i=1, 2, …, NP, a mutant vector $V_{i,\,G+1}$ is generated by

$$V_{i,\,G+1} = X_{r1,\,G} + F \cdot (X_{r2,\,G} - X_{r3,\,G}) \tag{3}$$

where random integers $r1$, $r2$, $r3 \in \{1, 2, \ldots, NP\}$ are mutually different and F controls the amplification of $(X_{r2,\,G} - X_{r3,\,G})$. In order to increase the diversity of the population, a trial vector $u_{i,G+1}$ is generated according to

$$u_{i,\,G+1}(j) = \begin{cases} V_{i,\,G+1}(j), & \text{if rand}(0, 1) \le Cr \text{ or } j = jrand \\ X_{i,\,G}(j), & \text{otherwise.} \end{cases} \tag{4}$$

TABLE I
DETAILS OF EIGHT PHASES

Phase	States(1-20)
1	GGGGGGrrrrGGGGGGrrrr
2	yyyyGGrrrryyyyGGrrrr
3	rrrrGGrrrrrrrrGGrrrr
4	rrrryyrrrrrrrryyrrrr
5	rrrrrrGGGGrrrrrrGGGG
6	rrrrrryyGGrrrrrryyGG
7	rrrrrrrrGGrrrrrrrrGG
8	rrrrrrrryyrrrrrrrryy

where Cr is the crossover probability and $jrand$ is a uniformly distributed integer value $\in [1, D]$ which ensures that $u_{i,\,G+1}$ gets at least one parameter from $V_{i,\,G+1}$. Then, a selection operator is performed if and only if the trial individual yields a better result than (the target individual) in the value of the fitness function, which is given by

$$X_{i,\,G+1} = \begin{cases} u_{i,\,G+1}, & \text{if } f(u_{i,\,G+1}) \text{ is better than } f(X_{i,\,G}) \\ X_{i,\,G}, & \text{otherwise.} \end{cases} \tag{5}$$

C. *Collaborative Evolutionary-Swarm Optimization(CESO)*

CESO is a simple method for tracking moving optima in dynamic environments by combining the search abilities of an evolutionary algorithm for multimodal optimization and a particle swarm optimization algorithm. It uses two populations of equal size. The first population used by CESO, called the CRDE population, is evolved by Crowding Differential Evolution (CDE) [17] algorithm which is a very efficient method for detecting multiple optima in static environments. The only modification of CDE to the original DE is the individual replacement. Usually, the parent producing the offspring is substituted, whereas in CDE the offspring replaces the most similar individual among the population. The similarity measure used is Euclidean distance between two candidate solutions. On the other hand, the second population called SWARM is a particle swarm updated according to PSO rules as mentioned in (1) and (2). An exponential crossover scheme is used in the original CESO algorithm [17]. We use the binomial crossover scheme as (4) in the modified CESO algorithm described in section III.

The best individuals of CRDE and SWARM are denoted by *cbest* and *gbest*. In order to detect changes, *cbest* is re-evaluated: it is considered that a change takes place if the new fitness value differs from the old one. The process of CESO is described as follows:

Step 1. Randomly initialize SWARM and CRDE;

Step 2. If a change appears or the distance θ between the best individuals of two populations is less than a threshold value *th*, go to step 3; otherwise, go to step 4.

Step 3. The CRDE individuals replace the SWARM population.

Step 4. Update SWARM and CRDE and evaluate populations.

Step 5. If *gbest* is better than *cbest*, *gbest* replaces *cbest*; otherwise, go to step 6.

TABLE II
SIMULATION AND CESO PARAMETERS

Parameter	Value
Simulation time	100s
Simulation area	0.5625km^2
Vehicle speed	0-5m/s
Number of traffic lights in a node	20
Maximum number of evaluations	3000
Individuals in a population	10
Individual size	18
Acceleration coefficients c_1 and c_2	1.49445
Inertia weight w	0.729
Crossover probability Cr	0.9
Mutation constant F	0.5

Step 6. If final condition is not satisfied, go to step 2; otherwise the algorithm stops.

The diversity of the search is maintained by the CRDE and the SWARM actually acts as a local search operator. When a change is detected in the environment or the distance between *cbest* and *gbest* is smaller than a threshold value *th*, the CRDE population will replace the SWARM population. By re-starting the search with particles scattered over the search space, the SWARM presents a good potential to locate the global optimum. At the end of iteration *gbest* replaces *cbest* if it is has a better fitness value. If *cbest* is replaced by *gbest*, it goes to step 3 and the SWARM is replaced by the CRDE. In this way, the SWARM will not easily get trapped in the local optimum of the current environment.

D. Configuration in SUMO

SUMO is an open source traffic simulation package including net import and demand modeling components [18]. We designed a grid-like network as shown in Fig. 1, which has nine intersections. An edge between two intersections is 250 m with three lanes in it. Each intersection contains a set of traffic lights and eight phases. Each phase shows the corresponding color states of all the traffic lights and the phase duration of this state. *G* and *r* mean a green light and a red light respectively. A yellow light is marked by *y*, in this state, vehicles will start to decelerate if far away from the junction, otherwise they pass.

Different cycle programs of traffic lights are tested in a simulation. In this study, at the beginning of each simulation, vehicles are uniformly distributed among the map with their maximum possible speeds (determined by their positions, traffic rules and distances). They have their trips defined by starting edges and destination edges. After a while, a new scenario captured by traffic monitors emerges and the corresponding optimal cycle program of traffic lights changes.

III. THE PROPOSED STRATEGY

This section describes the optimization strategy using modified CESO algorithm. Implementations of the solution encoding, the fitness function, the optimization procedure and the design reasons are presented.

TABLE III
DIFFERENT TRAFFIC CHANGES

Instance	Number of vehicles
I_1	500,589,667,725,770
I_2	770,725,667,589,500
I_3	589,500,725,667,770
I_4	500,540,589,667,725,770
I_5	770,725,667,589,540,500
I_6	589,500,540,725,667,770
I_7	500,540,565,589,631,667,701,725,749,770
I_8	770,749,725,701,667,631,589,565,540,500
I_9	589,500,725,540,667,749,565,701,631,770

A. Solution Encoding

The global staging of traffic lights has been encoded via a vector of integers. In this representation, phase duration of phase 1 and 5 of each intersection are sequentially stored in the vector (18 dimensions), that is, (p_{11}, p_{15}, …, p_{91}, p_{95}) in a vector (p_{ij} means the phase duration of phase *j* of intersection *i*). The rest (phase 2-4 and 6-8) use fixed phase durations and are not recorded in the vector. The reason for this representation is as follows. Since we aim to reduce the average delay time of all vehicles in the map, intuitively, phases which contain fewer green lights should be shortened. It can be seen that phase 1 and 5 contain more green lights (G) than other phases. For traffic safety, phase 2-4 and phase 6-8 cannot be removed, so we fix it to a certain time (in this paper 3s).

B. Fitness Function

As mentioned above, each vector codifies the cycle program of the traffic light programs. The fitness function is

$$f = \frac{T_{total}}{V} \tag{6}$$

where T_{total} is the total delay time of a scenario and V is the total vehicles. We use this criterion since the average delay time has a significant impact on traffic congestion.

C. Optimization Procedure

The optimization procedure is shown in Fig. 2. The optimization part is conducted by a modified CESO algorithm with details described below.

1) Two populations (SWARM and CRDE) are randomly initialized with a set of integer values. These values are set within $[5, 15] \in N$ since a too large value may result in prolonged delay time of one direction, whereas a too small value may cause security problems. Different combinations of phase durations will bring different traffic conditions.

2) SWARM is reinitialized when *gbest* is equal to *cbest* rather than estimating the distance between *cbest* and *gbest*. In this way, the algorithm avoids the setting of threshold value *th* and the SWARM has more time to explore. In addition, SWARM is also updated when a new scenario emerges. Different from the original algorithm, we do not need to test whether a change happens.

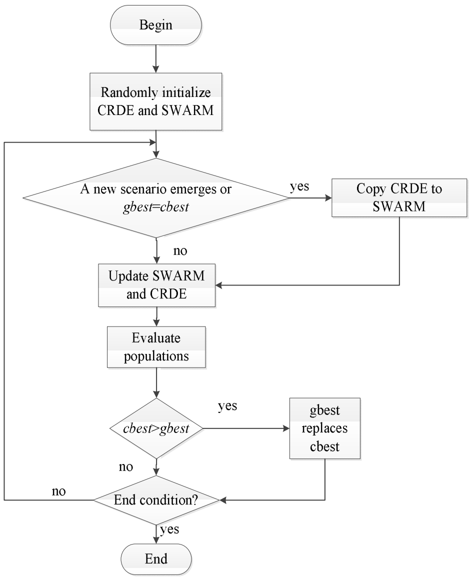


Fig. 2. Flowchart of the modified CESO algorithm.

TABLE IV
MEDIAN FITNESS VALUES OBTAINED BY OUR CESO AND PSO FOR INSTANCES OF FIVE SCENARIOS

Instance	algorithm	scenario				
		1	2	3	4	5
I_1	CESO	10.53	**13.58**	**13.61**	**16.21**	**16.26**
	PSO	**10.48**	13.60	13.65	16.49	16.60
I_2	CESO	16.65	**16.34**	**13.54**	**13.58**	**10.35**
	PSO	**16.60**	16.49	13.65	13.60	10.48
I_3	CESO	**13.43**	**10.23**	**16.05**	**13.51**	**16.50**
	PSO	13.60	10.48	16.49	13.65	16.60

TABLE V
MEDIAN FITNESS VALUES OBTAINED BY OUR CESO AND PSO FOR INSTANCES OF SIX SCENARIOS

Instance	algorithm	scenario					
		1	2	3	4	5	6
I_4	CESO	10.71	**12.1**	**13.56**	**13.60**	**16.05**	**16.39**
	PSO	**10.53**	12.63	13.60	13.69	16.54	16.80
I_5	CESO	**16.60**	**16.41**	**13.44**	**13.53**	**12.28**	10.74
	PSO	16.80	16.54	13.69	13.60	12.63	**10.53**
I_6	CESO	**13.35**	**10.51**	**12.10**	**16.18**	**13.65**	**16.49**
	PSO	13.69	10.53	12.63	16.54	13.69	16.80

TABLE VI
MEDIAN FITNESS VALUES OBTAINED BY OUR CESO AND PSO FOR INSTANCES OF TEN SCENARIOS

Instance	algorithm	scenario									
		1	2	3	4	5	6	7	8	9	10
I_7	CESO	10.74	**12.31**	**12.58**	**13.35**	14.19	**13.59**	**14.01**	**16.31**	14.33	**16.53**
	PSO	**10.55**	12.88	12.91	13.68	**14.12**	13.76	14.86	16.63	**14.11**	16.94
I_8	CESO	**16.78**	**14.08**	**16.56**	**14.86**	**13.70**	14.74	**13.66**	**12.62**	**12.23**	10.64
	PSO	16.94	14.11	16.63	**14.86**	13.76	**14.12**	13.68	12.91	12.88	**10.55**
I_9	CESO	13.74	10.63	**16.37**	**12.15**	**13.68**	**14.08**	**12.89**	**14.73**	14.69	**16.71**
	PSO	**13.68**	**10.55**	16.63	12.88	13.76	14.11	12.91	14.86	**14.12**	16.94

3) The position of an individual of SWARM has been modified to deal with integer values. Thus, (1) is changed to:

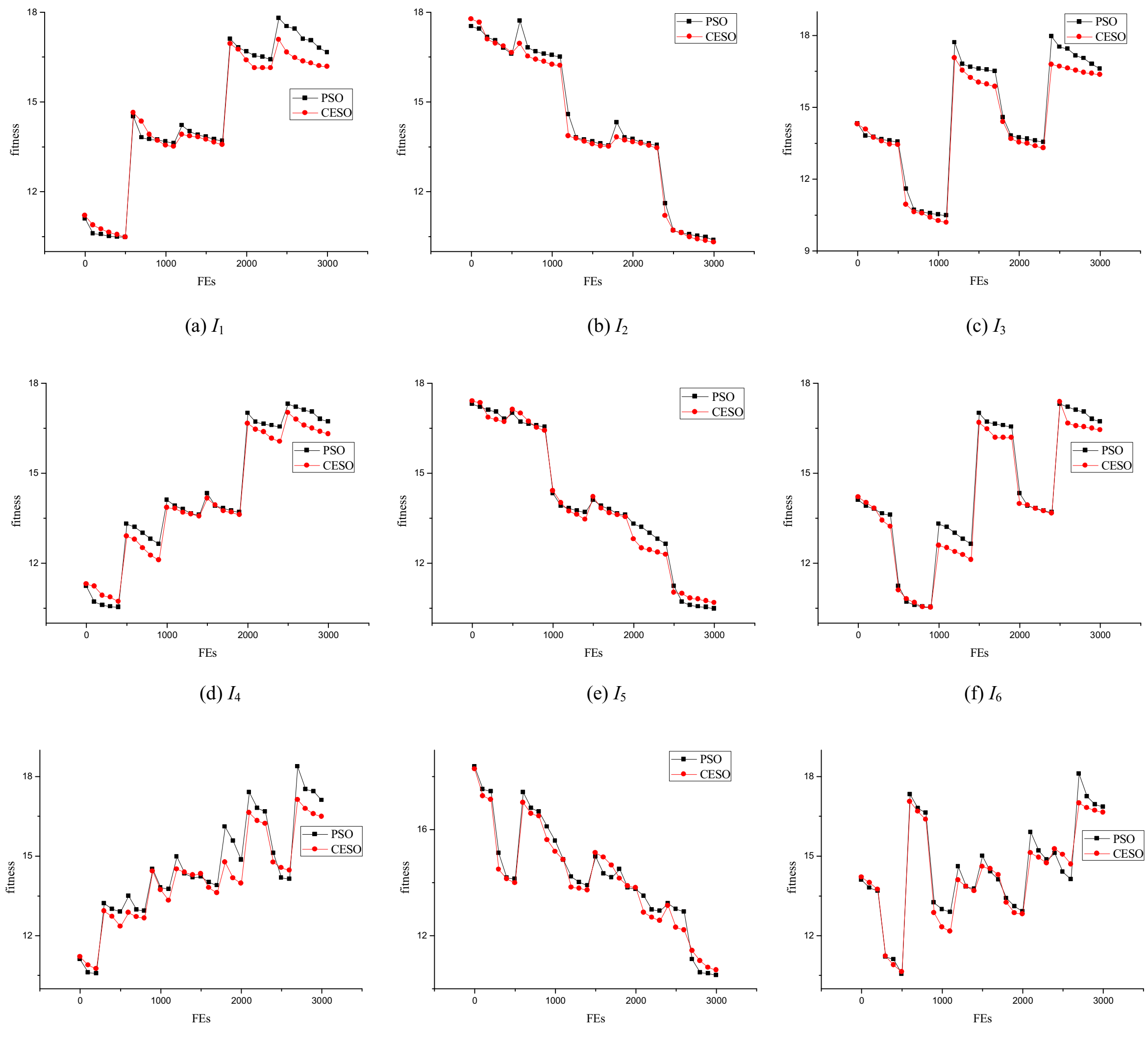
$$x_{i,t+1} = \lfloor (x_{i,t} + \mathrm{v}_{i,t+1}) \rfloor \tag{7}$$

When the new individual is generated, it is transferred to SUMO for updating the cycle program. After the simulation, it returns several files necessary to compute the average delay time of the scenario. A new scenario emerges after P FEs, where positions and number of vehicles change. In other words, P is the number of FEs needed to calculate the cycle program of the current scenario. The generated optimal cycle program is trasferred to the traffic system. Then the algorithm waits for the appearance of a new scenario (it is ignored when we conduct the experiments for convenience). Due to economic constraints, we obtain the new scenario by software rather than captured by new traffic monitors. New vehicles of starting and arriving and the information of previous scenario are taken into account when constructing the new scenario. Presumably, it is close to real-world situations, where the solution of the new scenario is related to the previous one. It is an ideal property for using dynamic evolutionary algorithms as they are designed for problems using previous information to calculate fitness function. In fact, as suggested in [19], if a change of the problem results in a totally new fitness landscape, nothing will beat a simple restart policy, since there is no information to transfer from one environment to another.

IV. EXPERIMENTAL STUDY

A. *Experimental Setup*

The simulation is carried out using the traffic simulator SUMO release 0.19.0 for windows. The experiments were performed on Intel Core i3-3240 CPU 3.40GHz, 4GB RAM, and VS2012. The population size (SWARM or CRDE) was set to 10 individuals performing 150 iteration steps, resulting in 3000 FEs per run. The remaining parameters are listed in Table II. Acceleration coefficients c_1 and c_2 were set to 1.49445 and inertia weight ω was set to 0.729 as recommended in [20]. The maximum velocity of each dimension of SWARM was set to 20% of the variable range. The number of scenarios per independent run was set to 5,6,10 respectively, in other words,



(a) I_1 (b) I_2 (c) I_3

(d) I_4 (e) I_5 (f) I_6

(g) I_7 (h) I_8 (i) I_9

Fig. 3. Convergence curves of best fitness values (median out of 30 runs) of CESO and PSO.

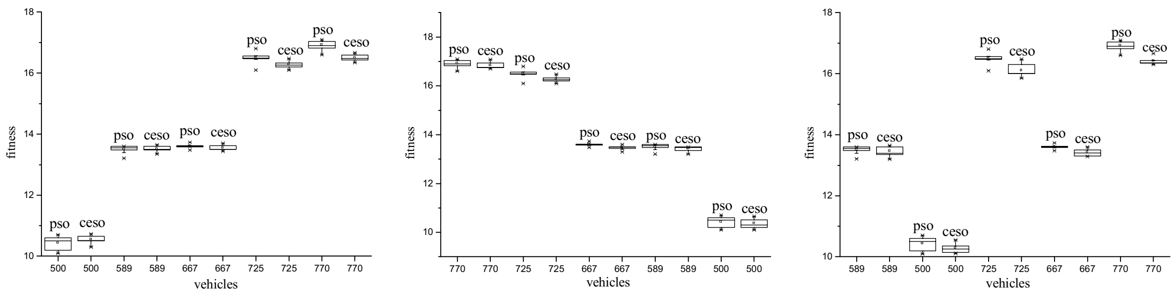
the new scenario emerged every 600, 500, 300 FEs. Different traffic changes are presented in Table III.

Besides, we also use PSO to compare with our strategy. For a fair comparison, the population size was set to 10 performing the same FEs between scenario changes. Acceleration coefficients c_1, c_2 and inertia weight w were set to the same as those in SWARM. It also uses (7) to deal with integer values.

B. *Results and Discussions*

Table IV-VI contains the median fitness values obtained by the proposed strategy for all instances. Additionally, the median fitness values obtained by PSO are also provided for every instance in order to permit comparisons. In these tables, we can observe that the proposed strategy obtained better fitness values than PSO for most of the scenarios. Relatively poor results occasionally appear. In fact, small changes in the number and position of vehicles may cause a big change in the optimal cycle program of traffic lights, which may lead to the poor performance of the proposed strategy as it uses previous information for the calculation of the current scenario. But typically two adjacent scenarios are relevant and the proposed strategy exhibits much stronger search ability in finding solutions to adjacent scenarios.

To give an insight into the performance of the two algorithms, the convergence curves are given in Fig. 3. Results were obtained based on 30 independent runs for every instance. In this figure, we can see that the proposed strategy converges faster than PSO most of the time for all instances.



(a) I_1 (b) I_2 (c) I_3

Fig. 4. Boxplots of two algorithms on I_1, I_2, I_3.

In addition, the performance of two algorithms on instances of five scenarios I_1-I_3 is further compared by the boxplots shown in Fig. 4. It can be seen that, for these instances, the results of the proposed strategy are better than PSO most of the time. To summarize, by using a dynamic algorithm, the proposed strategy can obtain better results than PSO in terms of average delay time of all vehicles in various scenarios.

V. CONCLUSION

In this paper, an optimization strategy is proposed using a Collaborative Evolutionary-Swarm Optimization (CESO) algorithm that can find time-varying optimal cycle program of traffic lights effectively. A popular traffic simulator SUMO is adopted to generate traffic flows and evaluate solutions. Nine instances of different traffic changes are tested. For universality, we designed ascending sequence, descending sequence and random sequence of number of vehicles captured by traffic monitors. Experimental results show that our strategy obtains better results than PSO in terms of the average delay time of all vehicles in various scenarios. Specifically, the proposed strategy is better than PSO most of the time.

REFERENCES

[1] S. Dipti, C. Min, and L. Ruey, "Neural networks for real-time traffic signal control," *IEEE Trans. Intell. Transp. Syst.,* vol. 7, no. 3, pp. 261–272, Sep.2006.

[2] J. Sánchez-Medina, 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 Trans. Evol. Comput.,* vol. 12, no. 1, pp. 25–40, Feb.2008.

[3] C.-J. Jin, W. Cheng, G.-R. Chang, H.-Y. Chang, and Hwang Hakchol, "Traffic signal timing and phase optimization for intersection based on IPSO," *10th World Congress on Intelligent Control and Automation (WCICA)*, 2012, pp. 711-714.

[4] M.-M. Zhang, S.-G. Zhao, L. Jie, and Y.-D. Qian, "Multi-phase urban traffic signal real-time control with multi-objective discrete differential evolution," *IEEE International Conference on Electronic Computer Technology*, 2009, pp. 296-300.

[5] R. Putha, L. Quadrifoglio, and E. Zechman, "Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions," *Computer‐Aided Civil and Infrastructure Engineering,* vol.27, no.1, 2012, pp. 14-28.

[6] D.-Z. Sun, R. F. Benekohal, and S. T. Waller, "Multiobjective traffic signal timing optimization using non-dominated sorting genetic algorithm," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003, pp. 198-203.

[7] H. Ceylan and M. G. H. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transportation Research Part B: Methodological,* vol.38, no.4, pp. 329-342, 2004.

[8] B. Park, C. J. Messer, and T. Urbanik II, "Enhanced genetic algorithm for signal-timing optimization of oversaturated intersections," *Transportation Research Record: Journal of the Transportation Research Board*, vol.1727, 2000, pp. 32-41.

[9] J. Lee, B. Abdulhai, A. Shalaby, and E.-H. Chung, "Real-time optimization for adaptive traffic signal control using genetic algorithms," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations,* vol. 9, no. 3, pp. 111-122, 2005.

[10] A. Stevanovic, P. T. Martin, and J. Stevanovic, "VisSim-Based Genetic Algorithm Optimization of Signal Timings," *Transportation Research Record: Journal of the Transportation Research Board*, vol.2035, pp.59-68, 2007.

[11] J. Garcia-Nieto, A. C. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optmization," *IEEE Transactions on Evolutionary Computation*, vol.17, no.6, pp.823-839, Nov. 2013.

[12] R. I. Lung and D. Dumitrescu, "A new collaborative evolutionary-swarm optimization technique," *Proceedings of the GECCO conference companion on Genetic and evolutionary computation,* 2007, pp. 2817-2820.

[13] L. L. Liu, S. X. Yang and D.W. Wang, "Particle swarm optimization with composite particles in dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.40, no.6, pp.1634-1648, Dec, 2010.

[14] S. Das, A. Mandal, and R. Mukherjee, "An Adaptive Differential Evolution Algorithm for Global Optimization in Dynamic Environments," *IEEE Transactions on Cybernetics*, in press.

[15] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 4, pp. 1942–1948.

[16] R. M. Storn, and K. V. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no.4, pp. 341-359, Dec.1997.

[17] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," *IEEE Congress on Evolutionary Computation,* 2004, pp. 1382-1389.

[18] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source trafficsimulation package SUMO," in *Proc. RoboCup Infrastruct. Simul.Competition*, 2006, pp. 1–5.

[19] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. of the Congr. on Evol. Comput,* 1999, pp. 1875-1882.

[20] Y. Shi, R. C. Eberhart, "Emprical study of particle swarm optimization," in *Proc.*, *IEEE Congr. Evol. Comput.*, 1999, pp.1945-1950.