American Journal of Intelligent Systems 2018, 8(1): 12-21
DOI: 10.5923/j.ajis.20180801.03

Best Vehicles Flow Traffic Light Controller Module Based on Particle Swarm Optimization (PSO)

Emad I. Abdul Kareem*, **Ayat Ismail Mejbel**

Computer Science, Mustansiriyah University, Iraq

Abstract A traffic light control module base on PSO algorithm has been given to determining the optimal set of adjacent streets that are the candidate to choose the green period time providing the best vehicle flow. In our previous work [19] a visual traffic light monitoring module has been introduced. This module able to define the traffic conditions (crowded, normal and empty). The proposed control module should be able to integrate with the previous monitoring module to develop a new complete intelligent traffic light scheme. The controller module shows its ability to select a set of streets. The green period time will be devoted to these selected streets to reach the optimal vehicle flow through the traffic light's intersections. The results show that the proposed control module improving the flow ratio about 72% to 96% with a different number of traffic lights.

Keywords Transportation System, Traffic Light Controller System, Particle Swarm Optimization (PSO)

1. Introduction

Fast transportation systems and rapid transit systems are nerves of economic developments for any nation. All developed nations have a well-developed transportation system with efficient traffic control on road, rail and air. Transportation of goods, industrial products, manpower and machinery are the key factors which influence the industrial development of any country. Mismanagement and traffic congestion results in long waiting times, loss of fuel and money.

It is therefore utmost necessary to have a fast, economical and efficient traffic control system for national development. The monitoring and control of city traffic are becoming a major problem in many countries. With the ever increasing number of vehicles on the road, the Traffic Monitoring Authority has to find new methods of overcoming such a problem [1, 2].

The measures taken are development of new roads and flyovers in the middle of the city; building of several rings such as the inner ring road, middle ring road and outer ring road; introduction of city trains such as the light rapid transit (LRT), and monorails; restricting of large vehicles in the city during peak hours; and also development of sophisticated traffic monitoring and control systems. Growing numbers of road users and the limited resources provided by current infrastructures lead to ever increasing, traveling times [3, 4].

One way to improve traffic flow and safety of the current transportation system is to apply automation and intelligent control methods to roadside infrastructure and vehicles [5]. A real time intelligent traffic light control module using Swarm algorithm has been presented in this work to configure many intersections in a city. Different number of intersections and streets have been used to evaluate the proposed module.

The proposed traffic light module shows its ability to improve the vehicle flow through these intersections and streets. The vehicles flow improvement reached to 96%.

2. Particle Swarm Optimization (PSO)

PSO is a heuristic global optimization method, originally provided in1995 by Kennedy and Eberhart. Particle Swarm Optimization (PSO) incorporates amalgamate swarming behaviors observed in schools of fish and flocks of birds. Since PSO is a population-based optimization tool, it can be implemented and applied easily to solve various optimization problems or the problems that can be transformed to optimization problems. The main power of PSO algorithms is its fast convergence as compared with many global optimization algorithms such that Simulated Annealing (SA), Genetic Algorithms (GA), and other global optimization algorithms. In PSO, there are other features such that no complex mathematical functions, no costly mathematical and doesn't require a large amount of memory [7].

* Corresponding author:
mmimad72@yahoo.com (Emad I. Abdul Kareem)
Published online at http://journal.sapub.org/ajis

Starting from a randomly distributed set of particles (potential solutions), the algorithms try to improve the solutions according to a quality measure (fitness function). The improvisation is performed through moving the particles around the search space by means of a set of simple mathematical expressions which model some antiparticle communications. These mathematical expressions, in their simplest and most basic form, suggest the movement of each particle toward its own best experienced position and the swarm's best position so far, along with some random perturbations. There is an abundance of different variants using different updating rules [8]. The general structure of a canonical PSO algorithm [9] (see algorithm 1).

```
Procedure Particle Swarm Optimization
Begin
        Initialize  x_i ,v_i and xbest_i for each particle i;
        While (not termination condition) do
        Begin
                For each particle i
                Evaluate objective function;
                Update xbest_i
         End
         For each i
                    Set g equal to index of neighbor with best xbest_i;
                    Use g to calculate v_i;
                    Update x_i = x_i + v_i;
                    Evaluate objective function;
                    Update xbest_i
                End
            End
    End
```

Algorithm 1. The General structure of a canonical PSO algorithm [19]

The optimal values for the parameters depend mainly on the problem at hand and even the instance to deal with and on the search time that the user wants to spend on solving the problem [10]. In fact, the main issue is to provide balance between exploration and exploitation tendencies of the algorithm. Total number of particles, total number of iterations, inertia weight and/or constriction factor, and cognition and social behavior coefficients (c1 and c2) are the main parameters that should be considered in a canonical PSO. The total number of iterations could be replaced with a desired precision or any other termination criterion. In general, the search space of an n-dimensional optimization problem can be conceived as a n-dimensional hypersurface. The suitable values for a, metaheuristic's parameter depend on relative ruggedness and smoothness of this hyperspace.

In general, there are two different strategies for the parameter value, selection, namely, off-line parameter initialization and online parameter tuning [10]. In off-line parameter initialization, the values of different parameters are fixed before the execution of the metaheuristic. These values are usually decided upon through empirical study. It should be noted that deciding about a parameter of a metaheuristic algorithm while keeping the others fixed (i.e., one-by-one parameter selection) may result in misleading observations since the interactions of the parameters are not taken into account. However, it is the common practice in the literature since examining combinations of the algorithm parameters might be very time-consuming. To perform such an examination, when desired, a meta optimization approach may be performed, i.e., the algorithm parameters can be considered as design variables and be optimized in an overlying level.

The main drawback of the off-line approaches is their high computational cost since the process should be repeated for different problems and even for different instances of the same problem. Moreover, appropriate values for a parameter might change during the optimization process. Hence, online approaches that change the parameter values during the search procedure must be designed. Online approaches may be classified into two main groups [10]: dynamic approaches and adaptive approaches. In a dynamic parameter updating approach, the change of the parameter value is performed without taking into account the search progress. The adaptive approach changes the values according to the search progress (which it used in this work).

Like the other Evolutionary Algorithms (EA), a PSO algorithm is a population based on a search algorithm with random initialization and there is an interaction between population members. In PSO, each particle flies through the problem space and has ability of remembering the previous best position, outrun from generation to another [11].

3. Literature Survey

(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.

(Madhavi Arora, and V. K. Banga, 2012) In this paper, we have discussed two techniques for traffic light control. Firstly, we have discussed morphological methods of edge detection for real time traffic control and then fuzzy logic. If we compare two methods we find that fuzzy logic is simple to implement than morphology method because morphology method is very lengthy procedure, even because it is edge detection method it does not perform well during night time, edges of certain vehicles will not able detect due to dark at night time, but fuzzy logic only counts the number of vehicles not deal with edges, it gives more accurate results, if we see cost factor then morphological method is less costly than fuzzy because morphology method only needs high quality camera not sensors which is less costlier.

(Emad, I. Abdul Kareem, et al., 2014) this work uses small associative memory. It will learn all street traffic conditions. The controller uses a virtual data about the traffic condition of each street in the intersection. Thus, in an image processing module this video camera will provide visual information. This information will be processed to extract data about the traffic jam. This data represented in a look - up table, then smart decisions are needed when the intersection management determines the street case of each street at the intersection based on this look- up table.

(Ali A.J et al., 2014) in this study, they plan to construct a Simulation Model (SM) incorporate with the Road Network Design Method (RNDM), Genetic Algorithm system (GAs), and Optimization method. In order to extract a general diagnosis for the traffic congestion problem in the main urban areas in Malaysia, and to find out the optimal solutions for transportation system problems in Malaysia. The results of this study used to apply optimal transportation policies with the aim of providing useful insights on critical issues about the impact of such policies, and to help as a guide for the Malaysian decision makers to elaborate the optimal transportation policies.

(Khaled, 2016) in this study, a new Artificial Intelligence Techniques (AIT) and Simulation Model (SM) are applied in order to elicit a general diagnosis for the traffic congestion problem in Kuala Lumpur and Kuantan. An integrated model involves a Neural Network (NN), Fuzzy Logic (FL), Genetic Algorithm (GA), and Simulation Model (SM) is used. The current traffic demand data will be captured by strategically placed cameras. By receiving and processing data, they plan to use our integrated model to adjust traffic lights, timing to optimize traffic flow in coordinated traffic light systems, in order to minimize the traffic congestion through controlling traffic lights. The results of this study used to suggest and apply more efficient transportation policies with the aim of providing useful insights on traffic congestion problem, and to assist the Malaysian decision makers to elaborate the best transportation policies.

4. The Proposed Traffic Light Control Module

The proposed control module will be integrated with our previous work [17] which it presented an intelligent a traffic light monitor module. This monitor module supplies a three street condition of each street in the intersection (i.e. empty, normal and crowded). As well as, an intersections, traffic light map need to be supported to the proposed traffic light control module. Though, the input of the proposed control module should be the intersections, traffic light map in addition to the outcome of the monitoring module. These two inputs will be used by the PSO algorithm to find the optimum set of adjacent streets to assign the green period time to them providing optimal vehicles flow.

4.1. Handling Mechanism of Best Vehicles Flow

In this method the proposed module will take the flow of vehicles into account. In this method (see algorithm 1), the green period time will be given to the set of streets with optimal flow of vehicles through a set of selected street. This algorithm needs to use an adjustment array to represent streets adjustment. Choosing adjacent set streets mean more vehicles' flow during the green period time.

This method needs to optimize a random selected streets to achieve an optimal set of adjacent streets. These selected streets will be as particle in PSO algorithm. A reward will be given for each pair of adjacent streets in a particle. Thus, the fitness function of the PSO will sum all the given rewards for each particle. As shown in equation (1).

$$\text{Fitness Function } P(i)(j) = \begin{cases} \text{If } p(i)(j) \text{ adjust with } p(i)(j+1) \text{ Fitness Function } P(i)=p(i)+ \text{reward...(1)} \\ \text{Otherwise...no action} \end{cases}$$

Where i: i th particle.

j: j th offspring for i the particle.

Reward: the reward value that should be added to the fitness function. When two streets in the particle are adjacent.

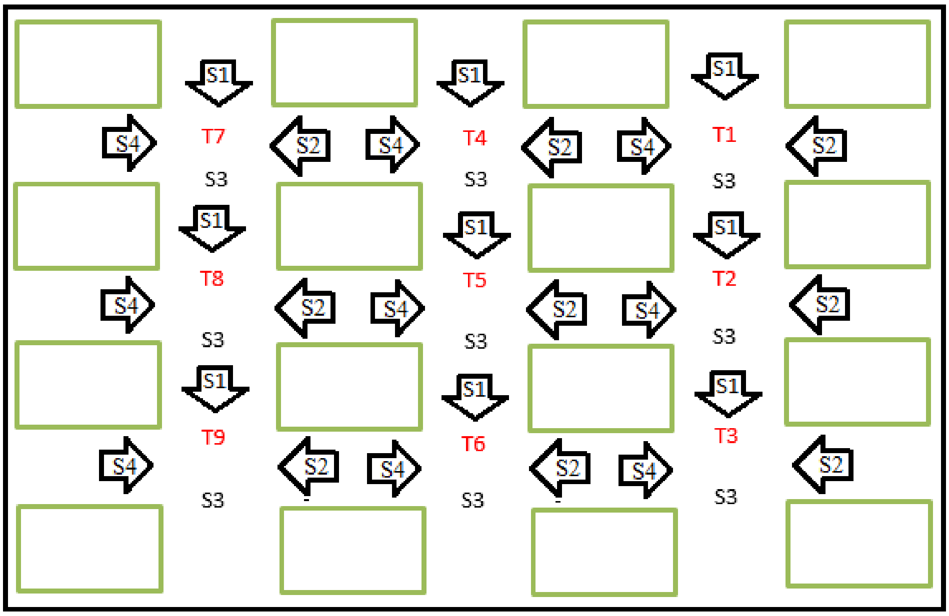

Figure 1. Example of adjustment street array

4.2. The Proposed Control Module Algorithm

The green period time will be given to the optimal series of streets which it the output of the algorithm. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*. After finding the two best values, the particle updates its velocity and positions with the following equation (2) and (3).

$$y[\,] = v[\,] + c1 * rand(\,) * (pbest[\,] - present[\,]) + c2 * rand(\,) * (gbest[\,] - present[\,]) \quad (2)$$

$$present[\,] = present[\,] + v[\,] \quad (3)$$

v [ ] is the particle velocity, *present [ ]* is the current particle (solution). *pbest[ ]* and *gbest[ ]* are defined as stated before. *rand ( )* is a random number between (0,1). *c1*, *c2* are learning factors. Usually *c1* = *c2* = 2. The pseudo code of the procedure is as follows:

Algorithm : Traffic control module based on PSO

Input One-dimensional array contains the case of each street in the intersection 1... N, where N<=36.

Output the particle with Best fitness value (PBest)

Step 1: For each particle

Initialize particle randomly. {which it a set of streets}

Step 2: For each particle do

Step 2.1: Calculate fitness value. {using equation (1)}

Step 2.2: If the fitness value is better than the best fitness value (*pBest*) in history set current value as the new *pBest.*

Step 3: Choose the particle with the best fitness value of all the particles as the *gBest*

Step 4: For each particle do

4.1: Calculate particle velocity according equation (2).

4.2: Update particle position according equation (3).

5. Result and Analysis

The proposed control module has been evaluated using different number of traffic light and streets.

5.1. One Traffic Light Cycle

This test has been guided by three scenarios. For each scenario, there are different number of traffic light and streets, the results have been taken for one cycle without taking time into account.

5.1.1. Scenario No. 1

In this scenario, 9 Traffic lights and 36 streets have been considered. Figure (2) shows that the proposed controller has been improved the flow ratio of the vehicles through 9 traffic light has been improved. This improvement can be found by increasing the number of adjacent streets to more than 98%. This improvement has been increased as soon as the number of PSO iterations is increased. The saturation phenomenon has been reached when the number of iterations increased more than 60 iterations.

5.1.2. Scenario No. 2

In this scenario, 16 Traffic lights and 64 streets have been considered. The same as the previous scenario, Figure (3) shows that the proposed controller has been improved the flow ratio of the vehicles through 16 traffic light intersections has been improved. This improvement can be found by increasing the number of adjacent streets to more than 93%. The saturation phenomenon has been reached when the number of iterations increased more than 70 iterations.

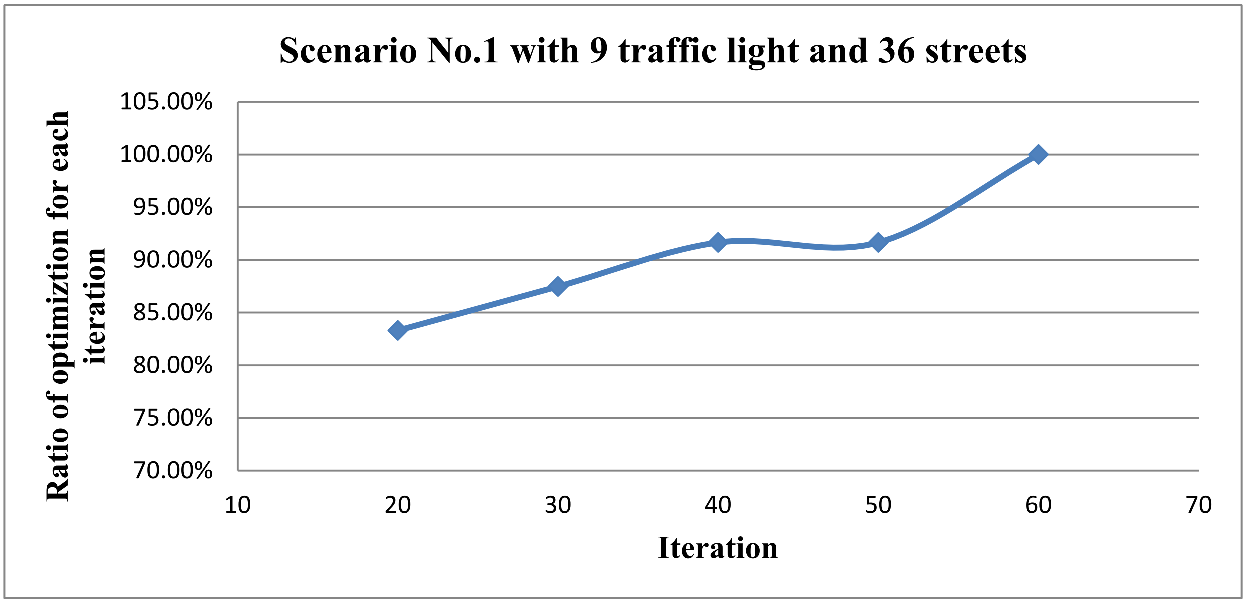

Figure 2. Scenario No. 1 with 9 Traffic lights and 36 streets

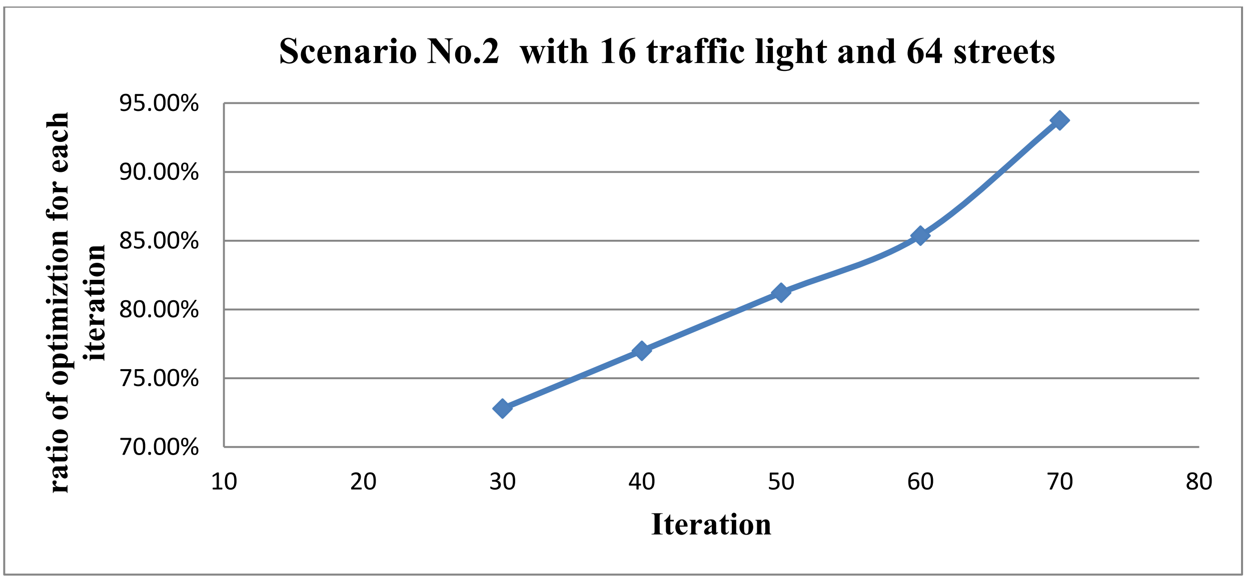

Figure 3. Scenario No. 2 with 16 Traffic lights and 64 streets

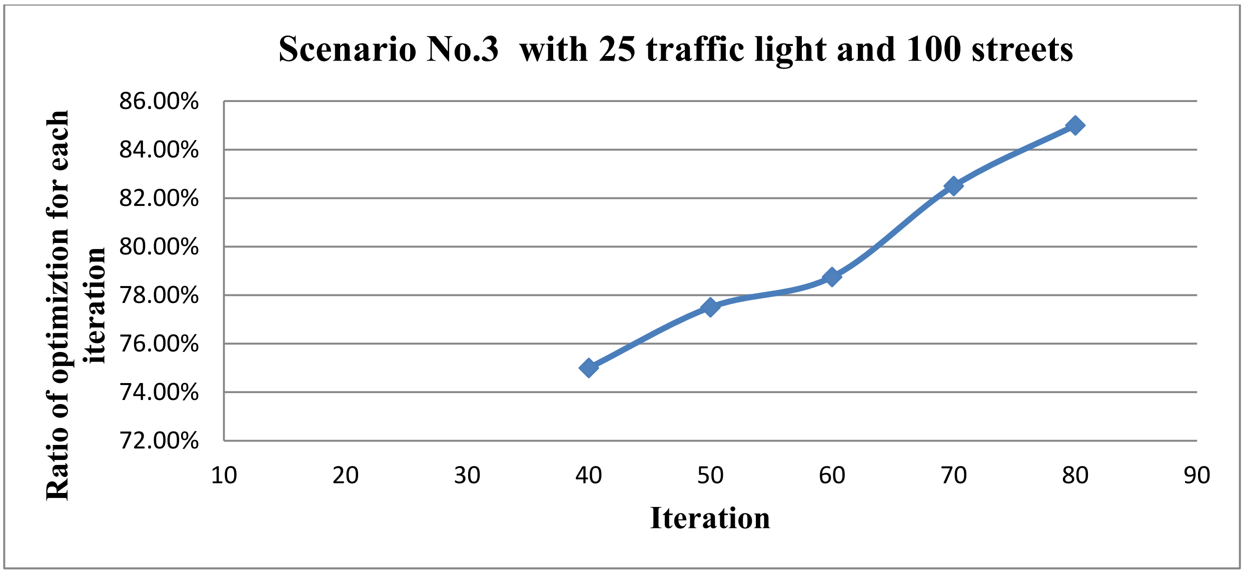

Figure 4. Scenario No. 3 with 25 Traffic lights and 100 streets

5.1.3. Scenario No. 3

In this scenario, 25 Traffic lights and 100 streets have been considered. Figure (4) shows that the proposed controller has been improved the flow ratio of the vehicles through 25 traffic light has been improved. This improvement can be found by increasing the number of adjacent streets to more than 85%. This improvement can be found by increasing the number of adjacent streets. The saturation phenomenon has been reached when the number of iterations increased more than 80 iterations.

5.2. More than one Cycle Test

In this test the proposed control module should be configured the traffic light intersections all the time with more than one cycle. This means, the proposed control module should be reconfigured the traffic light intersections according to the shortest green period time in the previous streets sequence.

The selection of the best cycle time considers the volume, intersection configuration, approach speeds and coordination with nearby intersections. Thus, typical cycle times - 30s to 120s. On the other hand, the cycle time should be as short as possible in off-peak periods 40s - 60s. While Cycle times of 120s - 150s are often required in peak periods in urban areas [18].

Three types of tests have been presented according to the previous mentioned: first test will be taking during typical cycle times, the second will be during off-peak periods and the third test will be during peak periods in urban area using the same scenarios which are presented previously in the section 5.1.

5.2.1. Typical Cycle Times

In this test the shortest cycle should be 30s. This means, the proposed controller module should be reconfigured the traffic light intersections each 30s. For scenario 1, figure (5) shows that the controller module needs just 5.9s to improve the flow of the vehicles to more than 98% through 9 traffic and 36 streets. This improvement can be found by increasing the number of adjacent streets.

For scenario 2, figure (6) shows that the controller module can improve the flow of vehicles to more than 93% through 16 traffic lights and 64 streets. This improvement can be found by increasing the number of adjacent streets.

In the last scenario (scenario No. 3) figure (7) shows that the controller module can improve the flow of vehicles to more than 75% through 25 traffic lights and 100 streets. This improvement can be found by increasing the number of adjacent streets.

5.2.2. Off-peak Periods Time

In this test the shortest cycle should be 40s. This means, the proposed controller module should be reconfigured the traffic light intersections each 40s. For scenario 1, just like previous test, the controller module needs just 5.9s to improve the flow of the vehicles to more than 98% through 9 traffic and 36 streets (see figure 5). For scenario 2, (figure 8) shows that the controller module can improve the flow of vehicles to more than 93% through 16 traffic lights and 64 streets. This improvement can be found by increasing the number of adjacent streets.

For scenario No. 3 figure (9) shows that the controller module can be able to improve the flow of vehicles to more than 78% through 25 traffic lights and 100 streets. This improvement can be found by increasing the number of adjacent streets.

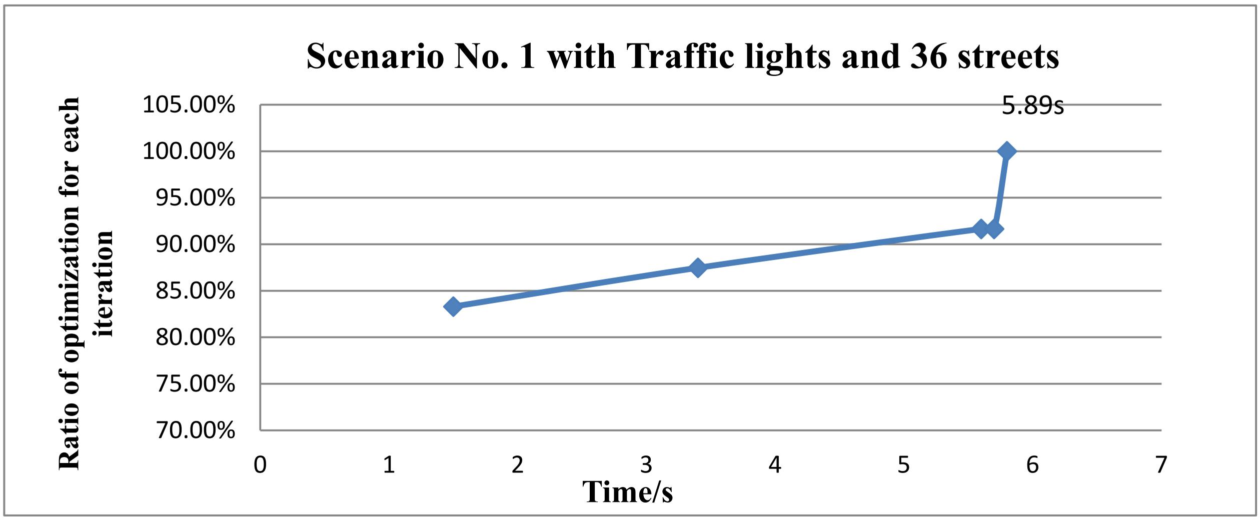

Figure 5. Scenario No. 1 with 9 Traffic lights and 36 streets

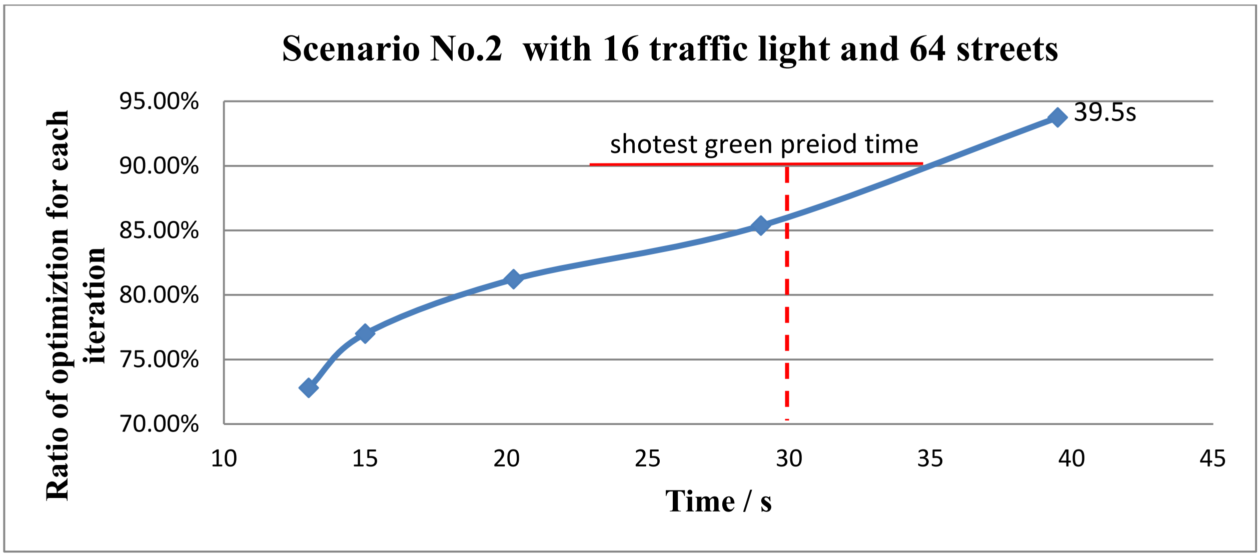

Figure 6. Scenario No. 2 with 16 Traffic lights and 64 streets

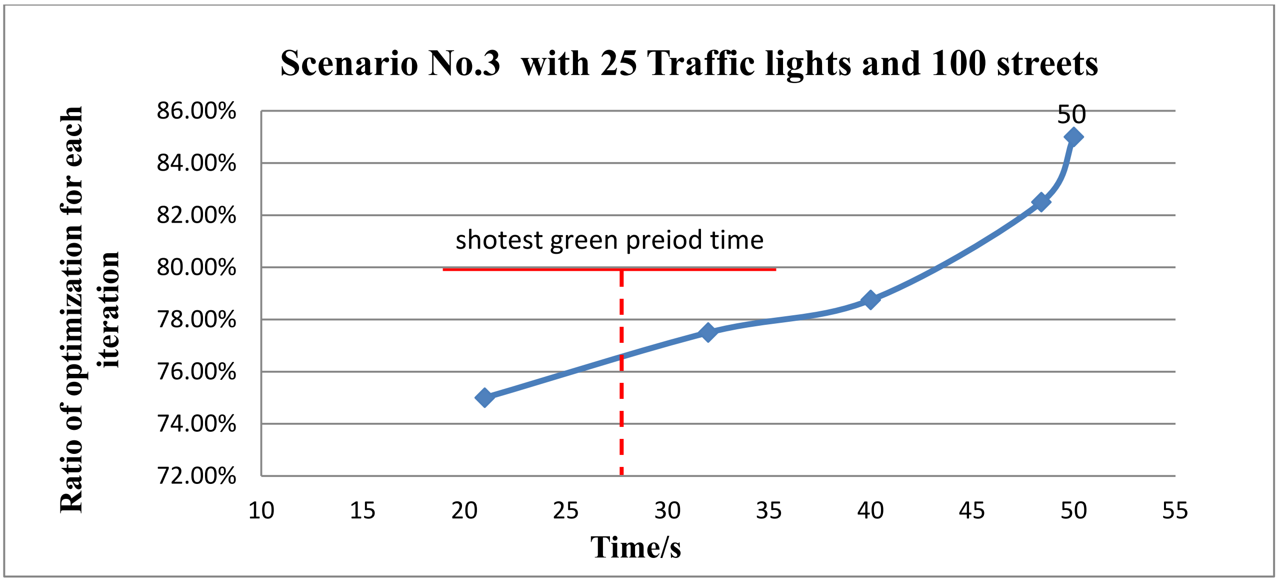

Figure 7. Scenario No. 3 with 25 Traffic lights and 100 streets

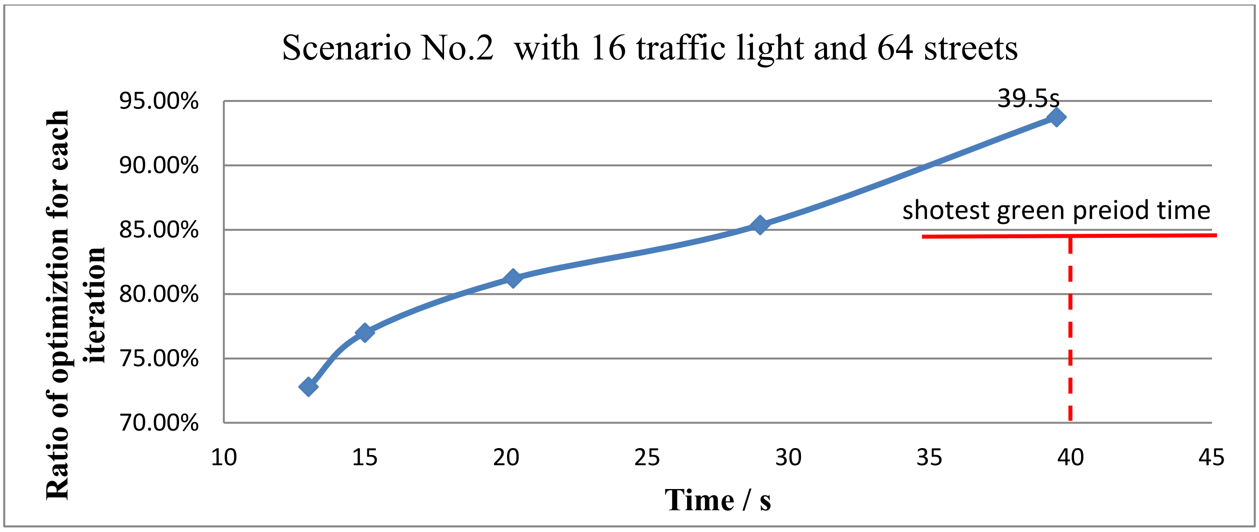

Figure 8. Scenario No. 2 with 16 Traffic lights and 64 streets

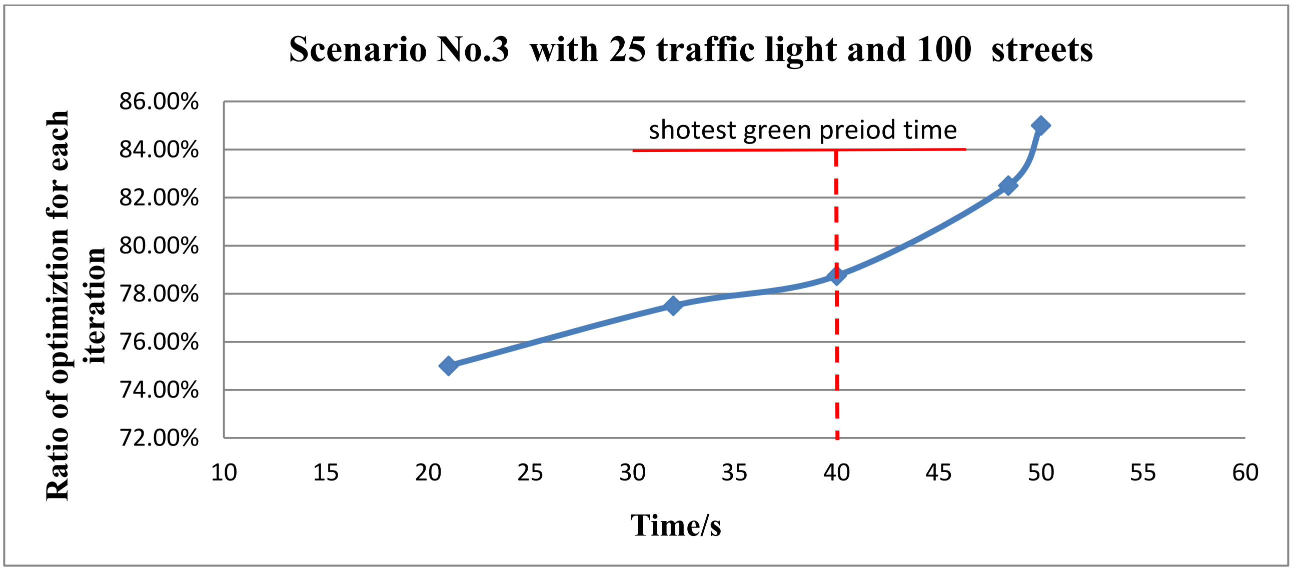

Figure 9. Scenario No. 3 with 25 Traffic lights and 100 streets

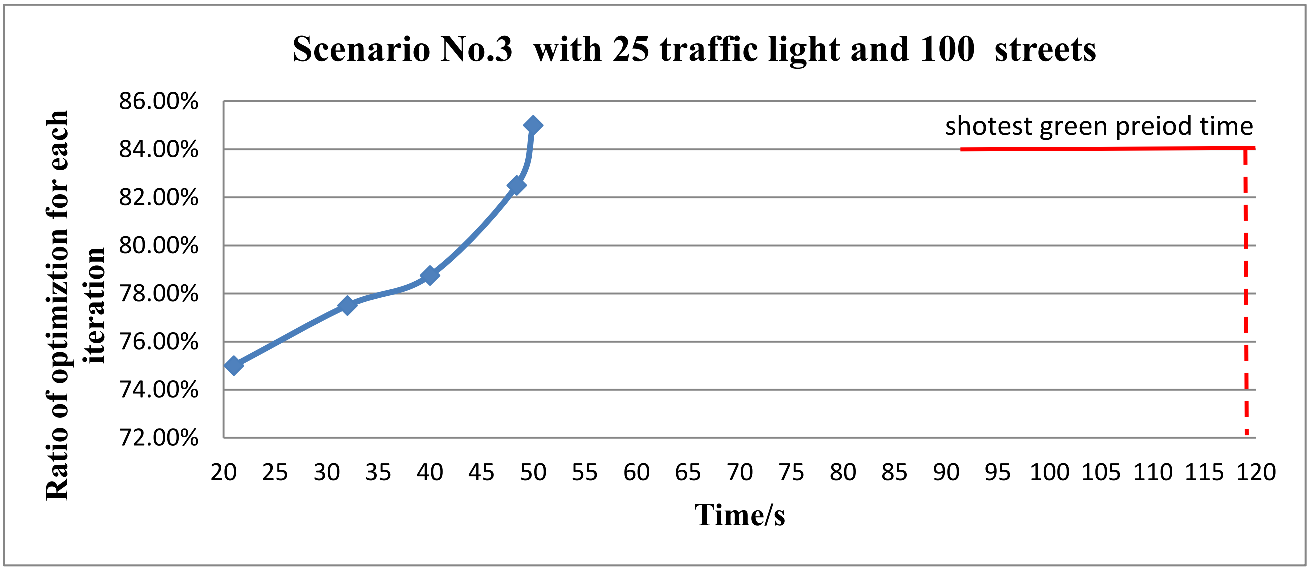

Figure 10. Scenario No. 3 with 25 Traffic lights and 100 streets

5.2.3. Peak Periods Time

In this test the shortest cycle should be 120s. This means, the proposed controller module should be reconfigured the traffic light intersections each 120s. For scenario 1 and 2, just like previous test, the controller module needs just 5.9s and 39s to improve the flow of the vehicles to more than 96% (see figure 8 and 9). This improvement can be found by increasing the number of adjacent streets. For scenario 3, just like the other two scenarios, the controller module can improve the flow of vehicles to more than 85% (see figure (10)) because the controller module needs just to 50s for improving. This improvement can be found by increasing the number of adjacent streets.

6. Proposed Module Complexity

The proposed module complexity depends on the number of computations required for a complete run of the PSO algorithm which are the sum of the computations required to calculate the cost of a candidate solution (based on current position of the particles) and the computations required to update each particle's position and velocity. Both of these are directly proportional to the number of iterations.

7. Conclusions

This research presents an intelligent control module. The proposed control module has been developed based on swarm technique. The proposed control module developed to optimize the traffic light configuration of a set of urban area's intersections. The results have been shown its ability to select one street from each traffic light according to each street condition in the intersection. The street condition should be offered from the monitor module which it integrated with the proposed control module. The green period time has been given to these selected streets to achieve the optimal vehicles flow through the traffic light's intersections. Its ability has been proven via three scenarios. These scenarios have been established by different number of traffic lights and streets. PSO improved the traffic flow through these traffic lights with ratio more than 96% for both scenarios 1 and 2 however, in scenario 3 the improvement was more than 95%. PSO needs from 80 to 90 iterations to reach the optimal series of streets. The results show that the proposed control module can be more efficient with peak periods because the traffic light cycle is relatively longer than the processing time that the proposed control module needs. Thus, the proposed control module is efficient as soon as the traffic light cycle is longer than PSO processing time.

REFERENCES

[1] Liu, "Routing finding by using knowledge about the road network", IEEE Transactions on System, man, and Cybernetics Part A: Systems and Humans. Vol. 27 No.4, 1997, pp 425-430.

[2] Sheu, "A composite traffic flow modeling approach for incident-responsive network traffic assignment", Physica A. Vol. 367. 2006, pp. 461-478.

[3] Abu-Lebdeh, G. and Ahmed, K., "Assessment of operational advantages of intelligent traffic control in congested conditions", Presented at the 9th ITS World Congress, Chicago, October 2002.

[4] Wangermann and Stengel, "Principled negotiation between intelligent agents: a model for air traffic management", Journal of Artificial Intelligent in Engineering. Vol. 12. 1998, pp. 177187.

[5] Roberto Horowitz, Pravin Varaiya "Control Design of an Automated Highway System", Proceedings of the IEEE, 2005 Available at: http://www.path.berkeley.edu/~varaiya/papers_ps.dir/ahsdesign.pdf.

[6] Hazem Ahmed, Janice Glasgow "Swarm Intelligence: Concepts, Models and Applications" Conference: Queen's University, School of Computing Technical Reports, At Kingston, Canada, Volume: Technical Report 2012-585, February 2012, DOI 10.13140/2.1.1320.2568.

[7] Bijaya Ketan Panigrahi, Swagatam Das, Ponnuthurai Nagaratnam Suganthan and Subhransu Sekhar Dash, "*Swarm, Evolutionary and Memetic Computing*". Springer-Verlag Berlin Heidelberg, ISSN 0302-9743, 2010.

[8] © Springer International Publishing AG 2017 A. Kaveh, Advances in Metaheuristic Algorithms for Optimal Design of Structures, DOI 10.1007/978-3-319-46173-1_2.

[9] Kennedy J (2006) Swarm intelligence. In: Handbook of nature-inspired and innovative computing. Springer, New York, pp 187–219.

[10] Talbi EG (2009) Metaheuristics: from design to implementation. Wiley, UK.

[11] Faez Hassan Ali, *"Improving Exact and Local Search Algorithms for Solving Some Combinatorial Optimization Problems* ". Ph.D thesis, al-mustansiriya university college of science, department of mathematics, 2015.

[12] 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.

[13] Madhavi Arora, V. K. Banga "*Real Time Traffic Light Control System Using Morphological Edge Detection and Fuzzy Logic"*. 2nd International Conference on Electrical Electronics and Civil Engineering (ICEECE'2012) Singapore April 28-29, 2012.

[14] Abdul Kareem E and Abbas S and Mahmood S. "*Intelligent traffic light controller based on MCA associative memory*". Science Journal of Circuits, Systems and Signal Processing. 2014; 3(6-1).

[15] Adham A and Abdul Rahman K, Hayyan M. "*An Integrated Model to Enhance the Transportation Methods in Malaysia: Review Paper*". Journal of Applied Science and Agriculture. 2014; 9(18).

[16] Abdul Rahman K. "An Artificial Intelligence Techniques and Simulation Model to Control a Traffic Jam System in Malaysia". Asian Journal of Business and Management. 2016; 4(1).

[17] Emad I Abdul Kareem, Aman Jantan (2011), *"An Intelligent Traffic Light Monitor System using an Adaptive Associative Memory",* IJIPM: International Journal of Information Processing and Management, Vol. 2, No. 2, pp. 23 ~ 39.

[18] Permit Writers Workshop, Signal Timing, http://cce.oregonstate.edu/sites/cce.oregonstate.edu/files/pw_sigtime.pdf, last visit 2017.

[19] Emad, I., Kareem, A., Jantan, A.: An intelligent traffic light monitor system using an adaptive associative memory. Int. J. Inf. Process. Manag. 2, 2(2.4), 23–39 (2011).