

Ant Colony Optimization for Real Time Traffic Lights Control on a Single Intersection

<https://doi.org/10.3991/ijim.v14i02.10332>

Nouha Rida (✉)

Mohammadia School of Engineers, Rabat, Morocco
nouharida@gmail.com

Mohammed Ouadoud

Abdelmalek Essaâdi University, Tetouan, Morocco

Abderrahim Hasbi

Mohammadia School of Engineers, Rabat, Morocco

Abstract—In this paper, we present a new scheme to intelligently control the cycles and phases of traffic lights by exploiting the road traffic data collected by a wireless sensor network installed on the road. The traffic light controller determines the next phase of traffic lights by applying the Ant Colony Optimization metaheuristics to the information collected by WSN. The objective of this system is to find an optimal solution that gives the best possible results in terms of reducing the waiting time of vehicles and maximizing the flow crossing the intersection during the green light. The results of simulations by the SUMO traffic simulator confirm the performance of the developed algorithm compared to the predefined time controller and other dynamic controllers.

Keywords—Traffic optimization, traffic light control, intelligent transportation system, ant colony optimization.

1 Introduction

The number of vehicles in the world continues to increase, leading to a saturation of the road network. This saturation represents the main cause of congestion, accidents, and pollution. These problems have major consequences for the economy and the daily life of the citizen. A possible solution to this problem is to effectively manage the road traffic, based on real-time traffic monitoring that will provide instant insight into the state of our roads. A simple example is an intersection, where traffic light management can be much more efficient by knowing the exact number of vehicles on each segment of road at any time; thus, the duration of green lights can be adapted according to the number of vehicles waiting.

In this paper, we propose an algorithm that uses ant colony optimization metaheuristics to plan and sequence the phases of a traffic light controller to adapt to instantaneous traffic variations in an isolated intersection.

The document is organized as follows: after a presentation, we present the context of the traffic lights control problem in section 2, then we show the analogy of traffic light control problem with scheduling problem in section 3. After in section 4 and 5 we present respectively related works and the Ant System Colony optimization metaheuristic. Thereafter, in section 6 we implement the dynamic traffic lights system based ACO for phases scheduling. Finally, in section 7 we evaluate our method via the SUMO simulator and we demonstrate its effectiveness by comparing the results to other methods.

2 Traffic Lights Control Problem Control in an ITS Context

In the traffic lights control in a intersection, a static light plan is useful when there is some historical information on the arrival of vehicles. It can be built and optimized using average traffic data that has been taken over a long period of time. The arrival of Intelligent Transportation Systems (ITS) provided access to information on vehicle arrival time, speed, etc. in real time. This latter information is less relevant when a static light plan is being used at a crossroads or it will not be able to adjust to the natural variability of the flow. As a result, it cannot react to changing traffic conditions. This means solutions considered moderately satisfactory. However, an adaptive light plan will be able to adjust to both stable and variable circulation conditions. It is in fact for this reason that it becomes more interesting to process the stochastic data of the warning system of traffic monitoring systems by using the plan of the false adaptive. Traffic monitoring systems provide accurate data in real time, allowing the adaptive plan to choose the flows of the highest priority vehicles for their data on a go-ahead basis.

The real-time management of a traffic light in an ITS context seeks to determine the sequence of green time of the different flows of a junction that minimizes total waiting time and maximizes vehicle throughput crossing an intersection by adjusting these decisions to changing traffic conditions. Several difficulties must be considered when solving this problem. We must, among other things, consider the compatibility between flows in order to avoid conflicts between vehicles. We must also ensure that the maximum green light time does not exceed a certain threshold. Finally, the problem to be solved is in a real-time management environment, so the chosen approach will need a model that can be resolved quickly so that the model information can be updated and the process can be restarted.

Our goal is to control traffic lights in real time to optimize the total waiting time of vehicles in the intersection and maximize the flow of vehicles crossing the intersection. The real-time management approach therefore consists to exploit the data of the road traffic status in each lane of the intersection that is collected by a road traffic surveillance system by the optimization algorithm, thus makes it possible to construct the phase-by-phase traffic light plan according to the observed traffic conditions observed in real time.

3 Analogy of Traffic Light Control Problem With Scheduling

Due to the properties of the problem being studied, the traffic lights regulation problem can be approached as a scheduling problem. We can model this isolated junction as a resource that can leave a certain flow of vehicles crossing the junction in parallel.

Each flow will be considered a task waiting to be processed. The flows are divided into different phases (compatible flows). Each flow has a waiting time and a queue that is proportional to the transit time of this flow (such as each task at a date rather than "release date" and a run time "processing time").

For example, the flows of the intersection model shown in Figure 2 can be divided into eight phases presented in Table 2. That is, flows in the same phase are considered as tasks that can be processed in the same time in parallel. Similarly, the time lost to start a queue of the flow to cross the intersection can be considered as a preparation time "establishment time" of the resource (here the intersection).

Thus, the problem of regulating an isolated junction can be treated as a problem of scheduling a resource (green light) and where certain tasks can be treated in parallel (flow of the same phase) this system of scheduling is shown in Figure 1.

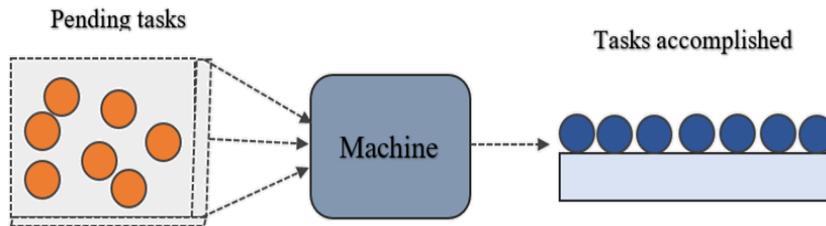


Fig. 1. Schedule multiple tasks to be run in a single machine

The problem of scheduling several tasks in a machine with the setup time and the date of availability is studied in [26]. Other constraints have also been discussed (see [27] and [28]).

However, for our traffic light control problem, the model has properties that are different, so a resolution algorithm will be needed.

We will solve the problem of control of traffic lights using a heuristic unlike an exact method for two simple reasons. First, the actual instances of the problem to be solved are of a very large size and are too difficult to solve accurately. Secondly, the time available to optimize the light plan is very short compared to the time needed to find the exact solution, because we have to optimize in real time. The Ant colony optimization heuristic has been chosen among various other possible heuristics, particularly because of its simplicity, its flexibility and above all the speed of execution.

4 Related Works

Dynamic real time control of traffic lights in road intersections is an important research focus in the intelligent transportation system. This control makes it possible to respond intelligently on the instantaneous variation of the traffic. The approach proposed in the literature focuses on two major objectives that are to maximize the flow crossing the intersection during green light and minimize the waiting time of vehicles. the dynamic controller uses traffic status information that can be collected in real time by inductive loops. [13], cameras [14], [15] [16], radars, VANET (ad hoc network for vehicles) [17], [18] or wireless sensor network [4], [19], [20], [21].

In the literature we find several theoretical models for adaptive management traffic lights: fuzzy logic [1], fluid mechanics [2], [19], [16] [9], neural networks [3], queues [4], genetic algorithm [3] and others.

In [1], a method based on fuzzy logic is employed. In this method, the authors present a table which defines the green light time according to the numbers of vehicles waiting. For example, for an intermediate flow with a number of vehicles arrives between 5/min and 10/min, we allow duration of 20s for green light.

The research work [5] proposes a light plan based on movement combinations can make simultaneously without any conflict. The authors use the same model of intersection presented in figure 4.1 with a topology of two sensors in such movement, which are separated by a variable distance that depends on the maximum green time. In this model of intersection, there are 4 lanes and 8 distinct combinations of non-conflict movements. This algorithm then selects the sequence of phases composing a cycle, according to several criteria: the presence of priority vehicles, the duration of the periods when there is no detection of new arrival, the cases of famine, the total waiting time and the length of the queues wait.

The model proposed in [5], however, is based on unrealistic assumptions, requiring vehicles to be of the same type and run at the same speed.

In [4], each lane equipped by two sensors is separated by a distance of 10 m. The authors presented each lane as an M/M /1 queue model and they use the Little law

$W = \frac{Q_L}{\lambda}$ to give equation 2.1 to calculate the queue length in each direction.

$$Q_{L_j} = Q_{L_{(j-1)}} + \lambda G_G - \mu G_G + \lambda G_R \quad (1)$$

Table 1. Notations

Notation	Meaning
W	The waiting time
Q_L	The queue length
λ	The vehicles arrival rate
J	Traffic cycle number
Q_{L_j}	The expected queue length for this cycle
$Q_{L(j-1)}$	The queue length from the previous cycle
λG_G	The arrival vehicles number in green light
λG_R	The arrival vehicles number in red light
μ	departure rate
G	The green light period for the phase
R	The red-light period for the phase

The algorithm proposed in [4] selects for each phase a combination of non-conflicting movements with the largest number of vehicles, in order to minimize the average queue length in the intersection.

In [6], the authors propose a study based on the GLD simulator to know the impact of the change of the wireless sensor network on the waiting time of the vehicles.

Firstly, the authors make a comparison between two topologies of the wireless sensor network, the first topology with one sensor per movement and the other with two sensors per movement. From the results of simulations with GLD simulator, the authors find that a topology with two sensors per direction gives a reduced waiting time compared to that with a sensor. After having arrived at this result, they studied the influence of the distance between the two sensors on the management of the road traffic as well as on the variation of the waiting time of the vehicles at the level of the intersection. They found that the distance between the two sensors does not have a big impact on the waiting time.

The method proposed in [19] selects the sequence of phases composing a cycle, according to several criteria: the presence of priority vehicles, the duration of the periods without detection of new arrivals, the total waiting time and the length of the queues. In [22] and [23], we have inspired the smallest job first method of task scheduling to be executed by a computer processor. We proposed the smallest phase first algorithm, which gives priority to phases with the smallest waiting line. Subsequently in [19] we introduced the waiting time in the algorithm proposed in [22].

A self-organization of traffic lights based on the historical data of the traffic status presented in [24] and [25].

In [7], the authors propose a topology of two sensors per movement. The first is just after the intersection and the other is placed before the intersection at a given distance. The data collected by these sensors are used to determine the sequence of phases for an intersection. To determine this sequence, the authors defined for each movement a score that classifies the movements according to the local score of the intersection and the state of the two neighboring intersections to this movement. The

local score of a movement is defined as a weighted sum of the waiting time and the number of vehicles waiting at the level of this movement.

5 Ant Colony Optimization

[8] and [9] initially proposed an ant colony optimization metaheuristic in order to solve the problem of traveling salesman. Their approach proceeds to the construction of solutions inspired by the synergy observed in the colonies of real ants ([10], [11], [12]).

Originally, the goal of this problem is to find the shortest tour for a given set of cities. A matrix providing distances, d_{ij} , between all pairs of cities is used to estimate

the length of a tour. For the commercial traveler problem, each ant represents an agent [8] and when moving from city i to city j , it leaves a trace on the path (ij) . In addition, an agent chooses the next city to visit based on the traces of pheromones and some heuristic information. At the initialization stage, all the arcs are initialized to the same amount of pheromone and each ant is placed in a randomly chosen city among all the cities. With a probability $1 - q$, where $0 \leq q_0 \leq 1$ is a parameter of the decision rule.

An ant k chooses the next city j by being on the city J , using a probability $P_{ij}^k(t)$ as follows:

$$j = \begin{cases} \arg \max_{l \in \text{tabou}_k} \left\{ [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta \right\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases} \quad (2)$$

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_l^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad (3)$$

Where, J_l^k presents cities that have not been visited by ant k . The coefficients α and β are parameters that make it possible to control the relative importance of the two elements. An ant k has a memory tabou_k , allows it to memorize the cities already visited in order to force it to form an acceptable solution.

After each choice of a city, the amount of pheromone $\tau_{ij}(t)$ is modified according to the following equations:

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0 \quad (4)$$

Where ρ is the evaporation rate and τ_0 represent the initial amount of pheromone deposit in each arc.

After all the ants have built a circuit, a global update of the pheromones is made for the arcs of the circuit, which represents the best solution (equation 5).

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t) \quad (5)$$

With:

$$\Delta\tau_{ij}(t) = \frac{1}{L_{cycle}^*} \quad (6)$$

L_{cycle}^* Presents the length of the best solution of the cycle.

If the total number of ants is m and the number of cities to visit is n , a cycle is realized when each of the m ants completes a tour of the n cities.

6 Traffic Light System

The solution proposed in this work is an intelligent real-time traffic signal management system in an isolated intersection (cf. Figure2) equipped with a wireless sensor network to collect traffic status information.

Our solution consists of three parts:

- A traffic data collection unit that is provided by a wireless sensor network and which is presented in the following section.
- A unit for phases scheduling of a traffic light cycle and which allows to choose the directions that will have the next green light.
- And the last unit allows to specify the duration of the green light of the phase which was chosen in the part of the scheduling.

6.1 Traffic light data collection

Our system (cf. Figure2) consists mainly of four directions (N, S, E, W), each of which contains two lanes (go straight and turn left). Vehicles are always allowed to turn right without restricting traffic.

To feed traffic management support systems traffic knowledge is an essential element in real time. A wireless sensor network is responsible for collecting traffic data such as: the lengths of the queues and the waiting time in each route. The network is composed of a number of magnetic sensor nodes installed in the middle of the road.

These sensors are responsible for detecting and classifying vehicles by evaluating the distortion of the Earth's magnetic field produced by the presence of ferrous objects.

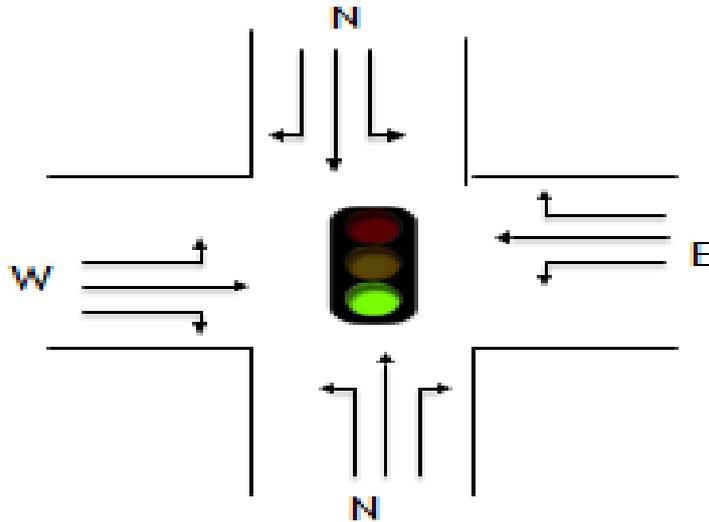


Fig. 2. Intersection model

In this work, we used a topology of two magnetic sensors per lane (cf. figure3): one located near the traffic light to count the number of vehicle departures and the other, installed at a distance from the first sensor to detect the arrival of vehicles.

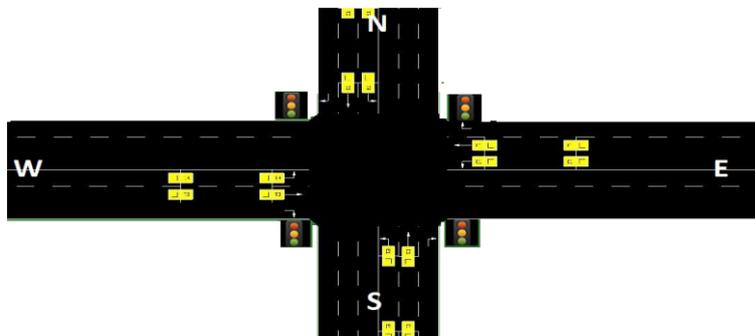


Fig. 3. WSN topology for an intersection

WSN captures the traffic flow data and communicates it to the base station, through their wireless radio interfaces. The base station is responsible for producing the traffic signal management decision by executing the traffic light control algorithm proposed thereafter.

6.2 ACO algorithm for traffic phases scheduling

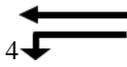
A traffic movement is a possible direction of movement of vehicles in an intersection. It is represented by a symbol consisting of a combination of two characters. The two characters represent the cardinal directions (S for south, N for north, E for east and W for west) for the source lane and the destination lane. The set P shows all the possible movements of the studied intersection model.

$P = \{ \{WE, WN\}, \{WN, ES\}, \{SN, SW\}, \{NE, SW\}, \{NS,NE\}, \{NS, SN\}, \{EW, ES\}, \{EW, WE\} \}$

Each traffic light controller defines a time called a cycle, which represents a sequence of up to four phases. Each of the phases is a combination of two movements (eg ES, EW, SN, etc.) that occur simultaneously. During a cycle, all the movements of the intersection should have the green light.

Table 1 and 4 show respectively the possible phases of the studied intersection model, and an example of a signal plane cycle for this model.

Table 2. Possible phases of the studied intersection model

1 	2 	3 	4 
5 	6 	7 	8 

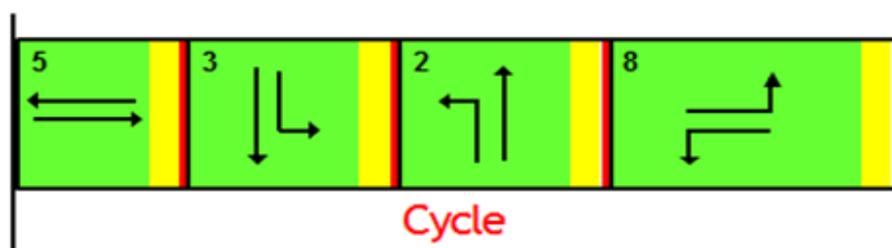


Fig. 4. Example of a traffic lights cycle

In this part, we develop an algorithm that executes at the end of each cycle phase and chooses the phase that will have the next green light among the eight phases presented in figure 3.

The main objective of this algorithm is to build a dynamic traffic light cycle plan that represents an optimal solution to reduce the waiting time of the vehicles and to maximize the flow crossing the intersection during the green light.

The proposed approach is an implementation of Ant Colony Optimization metaheuristic previously presented.

For the TSP problem, the ant colony optimization algorithm allows to specify the shortest tour of the cities. For our problem, we seek to determine the sequence of phases to minimize the total waiting time of vehicles in a given intersection, hence the obligation to adapt the ACO algorithm of TSP to give an effective solution to our problem. Below, we present the OCF adaptations to the problem.

Initialization: The initiation phase is the last running phase that is produced in the last sequence planning.

Visibility matrix: For the TSP, the distance matrix represents the distance between the cities, and the visibility matrix is constructed from the inverse of the elements of the distance matrix. For our problem, we use two visibility matrices to imply the two criteria that influence the waiting time in an intersection and that are the length of the queue and the waiting time of the vehicles in each movement.

By exploiting the data collected by the wireless sensors, the waiting time and the number of vehicles (the length of the queue) can be calculated according to equations 7 and 8.

$$Q_j(t) = AR_j + AG_j + Q_j(t - 1) - DP_j \tag{7}$$

Where:

$Q_j(t)$: The queue length for lane j and at time t

AR_j : Vehicles arriving during the red light and for the lane j .

AG_j : Vehicles arriving during green light and for the lane j .

$Q_j(t - 1)$: Vehicles are remaining from the last green light.

DP_j : Number of departures during current green light.

AR , AG and DP are calculated from the information received by the two sensors.

To calculate the waiting time W , there are two cases:

- If there are no vehicles waiting from the last green light, the waiting time is the time of the first vehicle arrived during this red time
- If there are still vehicles from the last green light, the waiting time is the current red-light time

Equation 8 gives the waiting time value for each lane as a function of FA , time of the first vehicle arrived, RT the time of the current red light and K , which is an index equal to 1 if there are vehicles waiting from the last selection of the green light and 0 otherwise.

$$W = \begin{cases} FA & \text{if } K = 0 \\ RT & \text{if } K = 1 \end{cases} \tag{8}$$

We use waiting time and queue lengths values to find the compromise that the algorithm seeks to establish between these two criteria. The coefficients β and γ allow controlling the relative importance of these two criteria.

Equations 2 and 3 that define the ACO transition rules are changed to 9 and 10, to fit our problem.

$$j = \begin{cases} \arg \max_{l \in tabou_k} \left\{ [\tau_{il}(t)]^\alpha \cdot [W_{il}]^\beta \cdot [Q_{il}]^\gamma \right\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases} \quad (9)$$

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [W_{ij}]^\beta \cdot [Q_{ij}]^\gamma}{\sum_{l \in J_l^k} [\tau_{il}(t)]^\alpha \cdot [W_{il}]^\beta \cdot [Q_{il}]^\gamma} \quad (10)$$

Candidates List: In the TSP problem, the candidates are the neighboring cities of the city i selected. In the traffic management problem dealt with here, the candidates are phases that can construct a traffic cycle. A phase consist two movements that can have green time at the same time without creating a conflict at the intersection.

For the intersection model shown in Figure 1, we have twelve possible movements. And if we eliminate the right turn motions that are still allowed, there will remain the following eight movements:

$$S = \{WE, WN, SN, SW, NS, NE, EW, ES\}$$

Some of these movements may have the green light at the same time without generating a blockage at the intersection and they are presented in the conflict matrix (Table 2), where 1 means that the vehicles of both movements can pass simultaneously while 0 expresses the movements that can create a conflict.

To summarize the table, we have eight possible combinations of the movements that can have the green light at the same time and which are the following phases:

$$P = \{\{WE, WN\}, \{WN, ES\}, \{SN, SW\}, \{NE, SW\}, \{NS, NE\}, \{NS, SN\}, \{EW, ES\}, \{EW, WE\}\}$$

Table 3. The conflict directions matrix

	WN	WE	SN	SW	EW	ES	NS	NE
WN		1	0	0	0	1	0	0
WE	1		0	0	1	0	0	0
SN	0	0		1	0	0	1	0
SW	0	0	1		0	0	0	1
EW	0	1	0	0		1	0	0
ES	1	0	0	0	1		0	0
NS	0	0	1	0	0	0		1
NE	0	0	0	1	0	0	1	

Table 4. Ant Colony Optimization Algorithm for Traffic light control Problem (ASTL algorithm)

```

Step 1: Initialization
t=0 c=0      t: time counter ; c: cycle counter
Step 2: departure phase
For each ant k:
    Select a starting city and add it to its list tabouk
Step 3: Construct a sequence for each ant
For i from 1 to n:
    For j from 1 to m:
        Select the next phase  $j, j \notin tabou_k$ , from the
        candidates according to:

$$j = \begin{cases} \operatorname{argmax}_{i \in \text{candidates}} \left\{ [\tau_{ij}(t)]^\alpha \cdot [W_{ij}]^\beta \cdot [Q_{ij}]^\gamma \right\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases}$$

        where J is selected according to the probability

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [W_{ij}]^\beta \cdot [Q_{ij}]^\gamma}{\sum_{i \in J^k} [\tau_{ii}(t)]^\alpha \cdot [W_{ii}]^\beta \cdot [Q_{ii}]^\gamma}$$

        Save the chosen phase in tabouk.
        Update pheromone intensity for the selected arc (i,j):

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0$$

Step 4: Global update of trace intensity
Calculate the total waiting time obtained, Lk for each ant k
For all arc (i,j) ∈ Cycle*, global update of the trace according to:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t)$$


$$\Delta \tau_{ij}(t) = \frac{1}{L_{\text{cycle}^*}}$$

t = t + 1 ; c = c + 1
Step 5: Sequence
If (c < cmax)
    Clean the tabouk and start again at step # 2.
Else:
    Stop
    
```

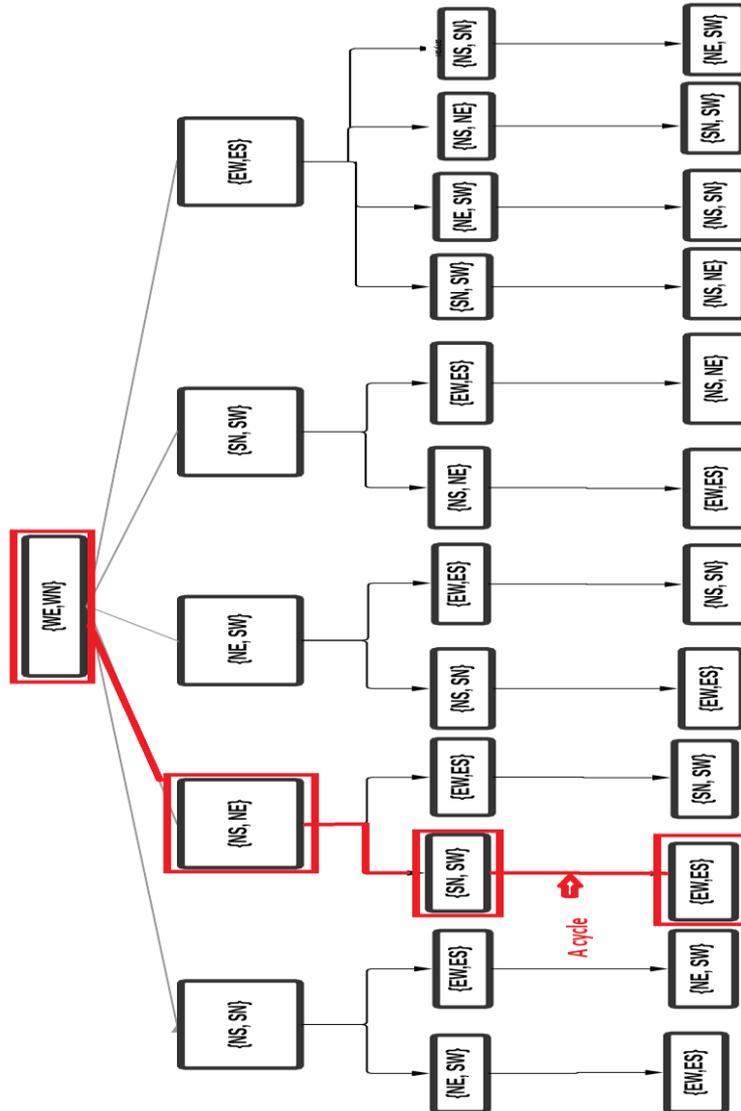


Fig. 5. Possible cycles with the initiation phase $\{WE, WN\}$

However, these eight phases are not all candidates for the ACO algorithm that we adopted since only four phases with different movements are needed to build a cycle. Above (cf. figure 5) is an example of possible candidates for the initiation phase

{WE, WN}. The diagram shows that for every possible solution, all movements exist only once in the solution.

We will now present the ant colony optimization algorithm (cf. algorithm 1) for solving the phases scheduling problem of a signaling controller. This algorithm returns the phases sequence of the cycle with the optimal waiting time. Because of our system is dynamic and it changed in each moment we execute the algorithm to determine the phase, which will have the green light at each end of the current green light.

7 Simulation Results and Discussion

We use the SUMO Simulation of Urban Mobility [29] to evaluate our ASTL algorithm. We opt for a simple intersection with three movements per lane; go straight, turn left and turn right, it will not be considered in the simulation because vehicles are always allowed to go right. For each 60-second period, we calculate with the five approaches the total waiting time of the vehicles in the intersection and the average queue length.

In order to properly analyze the results, we compared our dynamic plan with the ASTL method to three other planes:

- Static plan with four phases of duration of 25 or 20 seconds and which is illustrated below:

```
<tlLogic id="0" type="static" programID="0" offset="0">
  <phase duration="20" state="GrrGrrGrrGGG"/>
  <phase duration="25" state="GrrGGGGrrGrr"/>
  <phase duration="25" state="GrrGrrGGGGrr"/>
  <phase duration="25" state="GGGGrrGrrGrr"/>
</tlLogic>
```

- TSTMA dynamic plan presented in [4] and based on the choice of the phase with the largest queue.
- And dynamic plan with the TOPIOCA algorithm presented in [7], which defines a weighted sum of the waiting time and the number of vehicles as a criterion for choosing the next phase.

Several types of traffic have been created in order to better evaluate the solution according to the variations of traffic during the day. These are summarized in the table below.

Table 5. The traffic intensities used in the simulation

Scenario	traffic intensity	Number of vehicles
S1	Low	300
S2	intermediate	700
S3	high	1500

We tested our algorithm and other approaches at different traffic conditions. These tests were then analyzed using performance criteria to measure the potential benefits of the proposed approach.

The program was coded in python and the tests were done on a PC equipped with Intel® Core™ i7-4702MQ @ 2.20GHz processor.

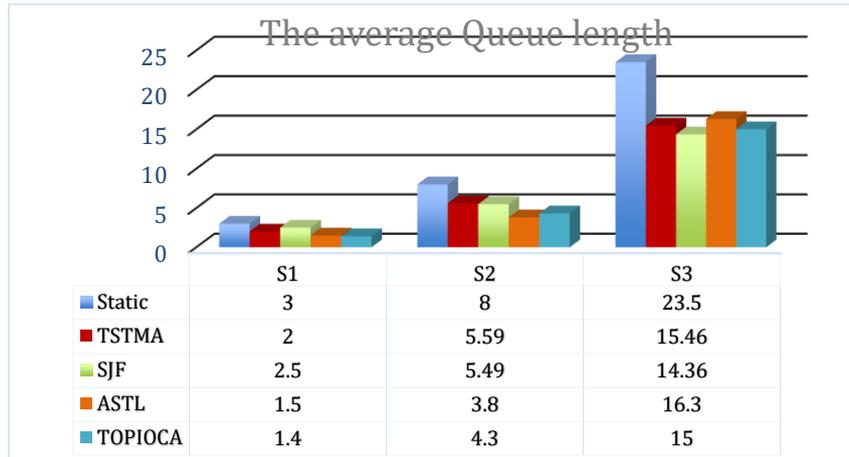


Fig. 6. Average Queue length comparison between fixed-time control, TSTMA, SJF, local TOPIOCA and our proposed algorithm (ASTL)

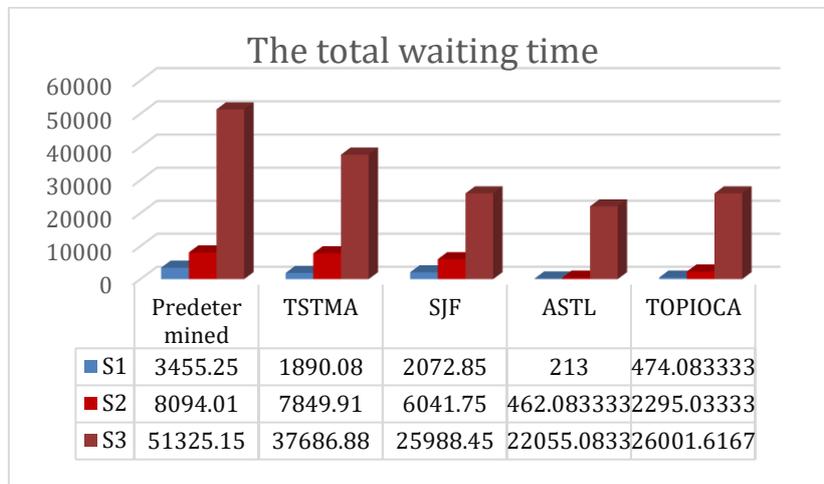


Fig. 7. Total waiting time with fixed-time control, TSTMA, SJF, local TOPIOCA and our proposed algorithm (ASTL) and with the three traffic scenarios (S1, S2, S3).

Figure 7 shows the average queues lengths in the intersection with the four approaches to traffic control and the three traffic intensity scenarios. The results obtained indicate that:

- For scenario S1, which has low intensity traffic, the simulations show that the results of the ASTL and TOPIOCA method are better than the other approaches. Specifically, for ASTL, the average queue is 1.5 vehicles, 14 with TOPIOCA and for the other methods is between 2 with SJF and 3 with the static plane.
- For the S2 scenario, it can be noted that for this scenario too, the ASTL and TOPIOCA adaptive methods give the shortest average queues with 3.8 on average for ASTL, 4.3 for TOPIOCA and the others are between 4.46 and 8 vehicles.
- For the S3 scenario, which shows high intensity traffic, the SJF method gives the best result in terms of reduction of the queue with 14.35 on average compared to 15 with TOPIOCA, 15.36 for TSTMA, 16.3 with ASTL and 23.5 with a static plane.

From the results of the total vehicles waiting time for the three traffic scenarios presented in Table 5, it can be noted that it is much smaller for ASTL than with the other methods. Table 7 below shows the percentage improvement in ASTL's adjusted waiting time compared to other approaches.

Table 6. The performance OF ASTL algorithm

	Static	TSTMA	SJF	TOPIOCA
S1	93 %	88 %	89 %	55 %
S2	94 %	93 %	92 %	79 %
S3	57 %	41 %	15 %	15 %

The time saving in the intersection is very important with ASTL for the simulations with the first two scenarios S1 and S2, saving between 88% and 94% compared to the TSTMA, SJF and the static plane. For Scenario S3, traffic conditions are much higher in intensity than other scenarios and the time savings are also important in this case. It is between 15% with SJF and TOPIOCA and 57% with a static plan.

In summary, among the methods tested in our simulations none of these methods perform well for all types of traffic as regards the reduction of queues. For each intensity of the traffic we found a method more efficient than the others and which is not the best for another intensity. Our ASTL model, which uses the ant colony optimization system, performs well by reducing average queue lengths for medium intensity traffic, and significantly improves vehicle waiting time at an isolated intersection for all types of traffic.

8 Conclusion

In this paper, we have developed an adaptive optimization method of controlling traffic lights at an isolated intersection using information provided by a wireless sensor network. We have used the metaheuristic ant colony optimization with several variants to solve the problem.

After testing and analyzing the simulation results, we concluded that we have a very effective method to reduce the waiting time of vehicles in the intersection. We

have demonstrated that our model saves time up to 57% compared to the static plan and in dense traffic conditions with high traffic intensity. Our future research focuses on the application of this model in other types of intersections and with vehicle priority constraints.

9 References

- [1] Zou, F., Yang, B. and Cao, Y., 2009, August. Traffic light control for a single intersection based on wireless sensor network. In *2009 9th International Conference on Electronic Measurement & Instruments* (pp. 1-1040). IEEE. <https://doi.org/10.1109/icemi.2009.5273994>
- [2] Liu, B. and Liu, W., 2011, May. Evaluation of traffic control methods at traffic circles. In *2011 Chinese Control and Decision Conference (CCDC)* (pp. 3371-3377). IEEE. <https://doi.org/10.1109/ccdc.2011.5968698>
- [3] Wei, W. and Zhang, Y., 2002. FL-FN based traffic signal control. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)* (Vol. 1, pp. 296-300). IEEE. <https://doi.org/10.1109/fuzz.2002.1005004>
- [4] Yousef, K.M., Al-Karaki, M.N. and Shatnawi, A.M., 2010. Intelligent traffic light flow control system using wireless sensors networks. *J. Inf. Sci. Eng.*, 26(3), pp.753-768.
- [5] Zhou, B., Cao, J., Zeng, X. and Wu, H., 2010, September. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In *2010 IEEE 72nd Vehicular Technology Conference-Fall* (pp. 1-5). IEEE. <https://doi.org/10.1109/vetecf.2010.5594435>
- [6] Tubaishat, M., Qi, Q., Shang, Y. and Shi, H., 2008, January. Wireless sensor-based traffic light control. In *2008 5th IEEE Consumer Communications and Networking Conference* (pp. 702-706). IEEE. <https://doi.org/10.1109/ccnc08.2007.161>
- [7] Faye, S., Chaudet, C. and Demeure, I., 2012, October. Un algorithme distribué de contrôle des feux de circulation sur plusieurs intersections par un réseau de capteurs sans fil. In *Nouvelles Technologies de la Répartition/Colloque francophone sur l'ingénierie des protocoles (NOTERE/CFIP)* (p. 1). <https://doi.org/10.1522/1388250>
- [8] Dorigo, M., Maniezzo, V. and Colormi, A., 1991. Positive feedback as a search strategy.
- [9] Dorigo, M., 1992. Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano*.
- [10] Deneubourg, J.L., Pasteels, J.M. and Verhaeghe, J.C., 1983. Probabilistic behaviour in ants: a strategy of errors?. *Journal of theoretical Biology*, 105(2), pp.259-271. [https://doi.org/10.1016/s0022-5193\(83\)80007-1](https://doi.org/10.1016/s0022-5193(83)80007-1)
- [11] Deneubourg, J.L. and Goss, S., 1989. Collective patterns and decision-making. *Ethology Ecology & Evolution*, 1(4), pp.295-311. <https://doi.org/10.1080/08927014.1989.9525500>
- [12] Moscato, P., 1999, January. Memetic algorithms: A short introduction. In *New ideas in optimization* (pp. 219-234). McGraw-Hill Ltd., UK.
- [13] Zhu, Y., Liu, X., Li, M. and Zhang, Q., 2013. POVA: Traffic light sensing with probe vehicles. *IEEE Transactions on Parallel and Distributed Systems*, 24(7), pp.1390-1400. <https://doi.org/10.1109/tpds.2012.233>
- [14] Rithesh, R.N., Vignesh, R. and Anala, M.R., 2018. Autonomous Traffic Signal Control using Decision Tree. *International Journal of Electrical & Computer Engineering (2088-8708)*, 8(3).

- [15] Odeh, S.M., 2013. Management of an intelligent traffic light system by using genetic algorithm. *Journal of Image and Graphics*, 1(2), pp.90-93. <https://doi.org/10.12720/joig.1.2.90-93>
- [16] Collotta, M., Bello, L.L. and Pau, G., 2015. A novel approach for dynamic traffic lights management based on Wireless Sensor Networks and multiple fuzzy logic controllers. *Expert Systems with Applications*, 42(13), pp.5403-5415. <https://doi.org/10.1016/j.eswa.2015.02.011>
- [17] Zhu, Q., Peng, C., Shi, J., Duan, P., Bao, Y. and Xie, M., 2016. Cooperative traffic light control based on semi-real-time processing. *Journal of Automation and Control Engineering*, 4(1). <https://doi.org/10.12720/joace.4.1.40-46>
- [18] Salman, M., Ozdemir, S. and Celebi, F., 2018. Fuzzy traffic control with vehicle-to-everything communication. *Sensors*, 18(2), p.368. <https://doi.org/10.3390/s18020368>
- [19] Rida, N., Ouadoud, M., Hasbi, A. and Chebli, S., 2018, October. Adaptive Traffic Light Control System Using Wireless Sensors Networks. In *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)* (pp. 552-556). IEEE. <https://doi.org/10.1109/cist.2018.8596620>
- [20] Faye, S. and Chaudet, C., 2016. Characterizing the topology of an urban wireless sensor network for road traffic management. *IEEE transactions on vehicular technology*, 65(7), pp.5720-5725. <https://doi.org/10.1109/tvt.2015.2465811>
- [21] Collotta, M., Bello, L.L. and Pau, G., 2015. A novel approach for dynamic traffic lights management based on Wireless Sensor Networks and multiple fuzzy logic controllers. *Expert Systems with Applications*, 42(13), pp.5403-5415. <https://doi.org/10.1016/j.eswa.2015.02.011>
- [22] Rida, N. and Hasbi, A., 2018, October. Traffic Lights Control using Wireless Sensors Networks. In *Proceedings of the 3rd International Conference on Smart City Applications* (p. 14). ACM. <https://doi.org/10.1145/3286606.3286791>
- [23] Rida, N. and Hasbi, A., 2018, October. Dynamic Traffic Lights Control for Isolated Intersection Based Wireless Sensor Network. In *The Proceedings of the Third International Conference on Smart City Applications* (pp. 1036-1044). Springer, Cham. https://doi.org/10.1007/978-3-030-11196-0_84
- [24] Yousef, K.M.A., Shatnawi, A. and Latayfeh, M., 2019. Intelligent traffic light scheduling technique using calendar-based history information. *Future Generation Computer Systems*, 91, pp.124-135. <https://doi.org/10.1016/j.future.2018.08.037>
- [25] Cano, M.D., Sanchez-Iborra, R., Freire-Viteri, B., Garcia-Sanchez, A.J., Garcia-Sanchez, F. and Garcia-Haro, J., 2017, July. A self-adaptive approach for traffic lights control in an urban network. In *2017 19th International Conference on Transparent Optical Networks (ICTON)* (pp. 1-4). IEEE. <https://doi.org/10.1109/icton.2017.8025051>
- [26] J. J., H. Liu Z., C. T. Ng and T. C. E. Cheng (2006). "Single machine batch scheduling problem with family setup times and release dates to minimize makespan." *Journal of Scheduling* 9 (6): 499-513. <https://doi.org/10.1007/s10951-006-8776-2>
- [27] Cheng, TCE, JJ Yuan, and AF Yang (2005). "Computers and Operations Research 32 (4): 849-859.", [Liu, Z. and W. Yu (2000). "Discrete Applied Mathematics 105 (1-3): 129-136.
- [28] Brucker, P., & Kovalyov, M. Y. (1996). Single machine batch scheduling to minimize the weighted number of late jobs. *Mathematical Methods of Operations Research*, 43(1), 1-8. <https://doi.org/10.1007/bf01303431>
- [29] Behrisch, Michael, et al. "SUMO—simulation of urban mobility: an overview." *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011

10 Authors

Nouha Rida is currently a Ph.D. Student at Computer Networks, Modeling and ELearning Research Laboratory at Mohammadia School of Engineer, Mohammed V University, Morocco. She received her degree in engineering in computer science by “Ecole Nationale Supérieur de Mine of Rabat”. Her main research interests are related to the intelligent transportation system, road traffic optimization and ELearning.

Mohammed Ouadoud is a Ph.D. in Computer sciences, at the Laboratory of Informatics, Research Operational and Statistic Applied (LIROSA) at Faculty of Sciences, Abdelmalek Essaâdi University. In 2018, he completed his Ph.D. thesis in computer science. His dissertation research, focus on Modeling and Prototyping a Learning Management System Based on the IMD-LD and the Hybridization between Learning Theories. His current research focuses on E-learning, Software Engineering, Geomatics and Bigdata. He is a reviewer in several International journals.

Abderrahim Hasbi got his PhD degree in computer science from the University Mohamed 5 of Rabat Morocco. He’s full Professor in Computer Science at the Mohammadia School of Engineering of the University Mohamed 5 Agdal, Morocco. He’s member of the Network and Intelligent systems Group and he has many contributions researches.

Article submitted 2019-02-17. Resubmitted 2019-11-26. Final acceptance 2019-11-26. Final version published as submitted by the authors.