

# Swarm-Based Controller for Traffic Lights Management

Federico Caselli, Alessio Bonfietti<sup>(✉)</sup>, and Michela Milano

DISI, University of Bologna, Bologna, Italy  
alessio.bonfietti@unibo.it

**Abstract.** This paper presents a Traffic Lights control system, inspired by Swarm intelligence methodologies, in which every intersection controller makes independent decisions to pursue common goals and is able to improve the global traffic performance. The solution is low cost and widely applicable to different urban scenarios. This work is developed within the COLOMBO european project. Control methods are divided into macroscopic and microscopic control levels: the former reacts to macroscopic key figures such as mean congestion length and mean traffic density and acts on the choice of the signal program or the development of the frame signal program; the latter includes changes at short notice based on changes in the traffic flow: they include methods for signal program adaptation and development. The developed system has been widely tested on synthetic benchmarks with promising results.

## 1 Introduction

Vehicular traffic is among the main plagues of modern cities. The ever increasing number of vehicles, both private and public, sets new challenges in the road network related to traffic optimization. This does not only mean to improve the traffic flow but also aims at reducing pollution and costs of the monitoring infrastructures [1].

Currently, the most common way to sense vehicular traffic is through the use of road deployed sensors. The most common sensor used is the inductive loop, whose installation and maintenance is expensive. To lower installation costs, approaches for traffic management based on V2X vehicular communication technology have started to be investigated. Unfortunately the V2X approaches mainly rely on message exchange between vehicles (V2V), requiring a high penetration rate of equipped vehicles (meaning a high number of V2X enabled vehicles over the total vehicle population) to achieve the desired goals.

One of the objectives of COLOMBO is the design and the development of an innovative, robust<sup>1</sup> and low cost<sup>2</sup> traffic light control system inspired by swarm intelligence methodologies (see [2]). Swarm intelligence is a discipline that studies natural and artificial systems composed of a large number of (typically identical or very similar) individuals called agents, which coordinate with decentralized

---

<sup>1</sup> COLOMBO system can work with very low penetration rates (see Section 5).

<sup>2</sup> Based on V2X technology.

control and self-organization. Recent research results [3][4] have shown that the principles underlying many natural swarm intelligence systems can be exploited to engineer artificial swarm intelligence systems that show many desirable properties and leads to effective solutions.

Following the principles of Complex Adaptive Systems (CAS), we conjecture that a smart and planned global traffic flow could emerge as the result of local decisions, automatically made by local controllers, executing simple policies in an emergent fashion. Emergent systems are very common in nature and colonies of social insects are one of the most interesting examples for our purposes. Every traffic light controller is a simple agent that controls one or more intersections and operates independently of all other controllers. It relies only on local information coming from the lanes that form the controlled intersection, which are distinguished between incoming and outgoing lanes. This principle is taken from autonomous agents theory, where each agent relies only on local information since there is no central coordination, either by choice or by force.

Following these principles, our system offers unlimited scalability, adaptability to traffic conditions and maximizes road network capabilities, while totally removing at the same time the costs associated to the control center and to the communication infrastructure required in conventional systems.

Our system is based on [5] which is inspired by two academic works: [6] presents local policies able to reach global traffic control through emergence. This is not enough for our purposes, since every policy is thought for a specific traffic density; [7] discusses a mechanism able to choose among different local policies with respect to traffic density, but executes non-reactive policies. Taking into account this work, we developed a control system composed of two levels, with different policies to handle multiple traffic situations in real-time and a high-level policy that selects between them. The system has been extensively tested and compared to a traditional static approach (called *static*) and to a dynamic approach based on inductive loops detections (called *actuated*). Results show that our approach is viable, even in case of low penetration rates.

The rest of the paper is structured as follows: Section 2 presents the related works. Section 3 illustrates the key ideas of the proposed system, which is detailed in Section 4. Then Section 5 presents a wide evaluation of our system. Section 6 concludes pointing out some interesting future development.

## 2 Related Works

Far from being exhaustive, given the large amount of literature in the field, in the following we present a rapid selection of works which we consider more similar to ours; for a comprehensive survey of existing efforts in the field we refer interested readers to [8].

Urban traffic control systems can be roughly divided into three major categories: centralized, decentralized and fully distributed systems.

- Centralized systems present a unified control center that collects data from the sensors scattered through the city. They have complete knowledge of the

controlled network, which is used to create the traffic plan. These systems can also be overridden by traffic experts, if it is necessary. Different solutions basically differ in the evaluation of the control strategy and inherit in some way from the TRANSYT off-line optimization model [9]; for example, SCOOT [10] is a largely deployed centralized solution.

- Decentralized systems present more than one decision-making entity and a master entity that coordinates them. This is the approach adopted by these two systems currently in production: SCATS [11] and UTOPIA [12].
- Fully distributed systems do not have any centralized controller that coordinates or generates traffic plans: every single intersection controller is an independent *agent* that takes its own decision and it is influenced only by its neighborhood. This is an innovative approach without evidence of large scale deployment. Many proposals about distributed solutions exist: they may be agent-based, as in [13], where an agent is in charge of handling the traffic lights in the controlled intersection and performs actions with regard to the local traffic status only, while in other cases the information coming from surrounding agents is also taken into account [14].

Static optimization is not able to adapt to changing needs in traffic. Instead, centralized solutions are able to reach good performances, but are really expensive: they need a unified control center that needs to be connected to every sensor in the network and to every traffic light controller. A system like this one requires high initial installation and maintenance costs. Also, centralized systems do not scale well when used to control big road networks. A partial solution to the scaling problem is given by decentralized system, since they do not require a central controller. Distributed approaches would be simpler and would scale better, but problems may arise for the communication part of these systems. [15] and [16] present this problem in relation to MARL (Multi Agent Reinforcement Learning), which has communication needs that grow exponentially with the number of agents. Communication requires also mutual knowledge, thus reconfiguration of neighboring agents is needed when the topology of the network changes: MARL has no centralized controller but still has high costs due to the communication part.

As previously stated our system is based on [5]. Our controller has enhanced [5] in several key aspects, such as:

- robustness: our system is robust to incomplete traffic information (i.e. different penetration rate), while [5] requires full knowledge.
- dispersion: our system takes into account the non homogeneity of the traffic flows over different lanes of the intersection.
- reliability: the representation of the actual traffic condition is based on the average speed instead of the number of V2X equipped vehicles. In fact, we see that the speed is more robust w.r.t. to changes in the penetration rate.

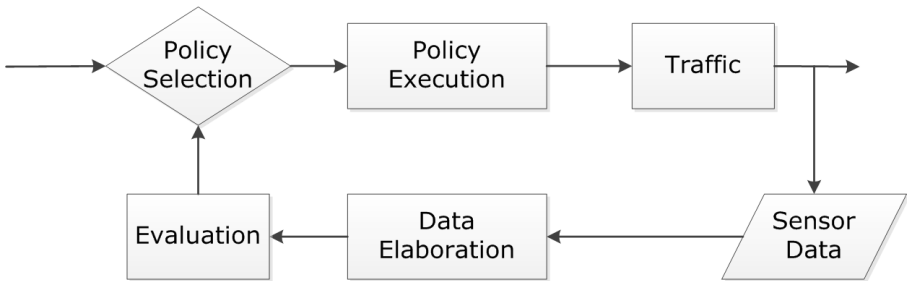
### 3 The Concept

Swarm intelligence systems make use of natural metaphors and share a common principle: they present a multitude of simple agents that may be unaware of the system they are part of. Typically, the interactions between the agents are based on alterations of the surrounding environment: this is a form of indirect coordination called *stigmergy* [2].

Our self-organizing system is made up of different independent traffic light controllers. Every controller works in a continuous loop like the one represented in Figure 1. Data about the status of the traffic is acquired by the sensors, translated into *pheromone values* and used as input for the *stimulus functions*. These stimuli are particular functions used to probabilistically determine which policy is the most appropriate to handle the current sensed traffic conditions. The policies are simple rules specifically defined to cope with different traffic conditions. The system also receives feedback from the traffic itself, rewarding or penalizing the choices it takes.

Our proposal abandons the traditional static approach: as implied before, the system decides when it is time to switch to the next phase on the basis of the sensed traffic conditions and not necessarily according to a clock. This makes our system able to react to changes in the traffic density both on the input and on the output lanes of the controlled junction. Communication with the neighboring traffic light controllers is done indirectly through stigmergy exploiting the natural metaphor of the pheromone and without explicit knowledge of the existence of other controllers.

The reason why we chose to forgo a centralized control is that it would be computationally too expensive and difficult to optimize. A centralized system would also need to predict the traffic behavior, which is known to be a hard task since the traffic is a complex system. Moreover, a decentralized system capable of self-organization is simpler to implement and is more reactive to rapidly varying conditions.



**Fig. 1.** Traffic lights execution loop.

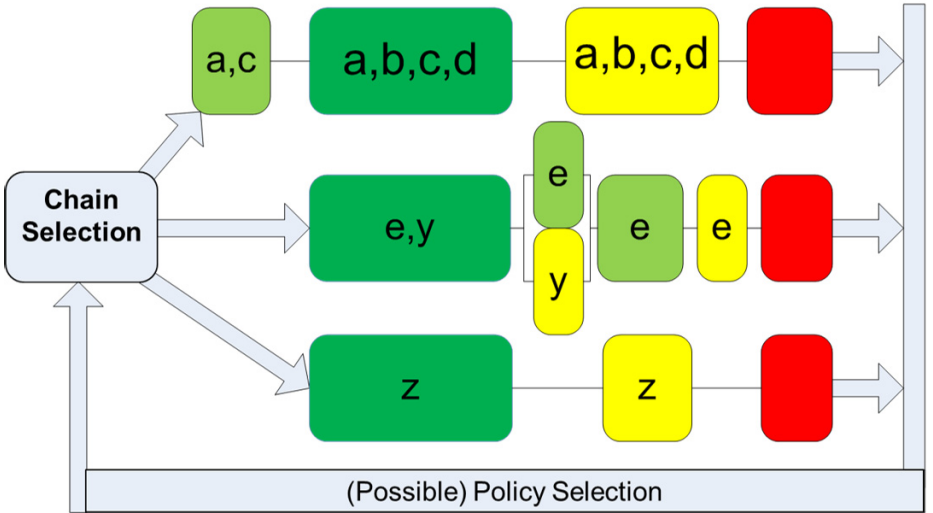
## 4 Description of the System

Our system is structured in two levels called *microscopic* and *macroscopic*. The former takes short-term decisions like which lanes should receive green and for how long the green light should be kept while the latter takes more higher-level decisions, like what is the criteria that should be used at low-level. Both judgments are done on the basis of the current traffic conditions.

### 4.1 Chains

The traditional static execution of a traffic light plan is a continuous loop between phases giving green to a particular set of directions, followed by yellow lights, red lights to all directions and then another green phase to allow traffic transit from a different set of directions. This is done in a static way, which means that green could be kept to lanes even after all waiting vehicles have left or given to lanes with no cars waiting for it. Our idea is that this decision should be taken on-line on the basis of the current traffic conditions. The best moment when this decision can be taken is during the so-called *all-red phase*, which is the phase needed when red is given to all the directions because of safety constraints.

The whole signal sequence can be split in different sub-sequences called *chains*, as shown in Figure 2. The first phase of a chain is called *target phase* and it gives green to a set of lanes identified as *target lanes*. The last phase of a chain, which gives red to all the lanes, is called *commit phase*. When the commit phase is reached, we probabilistically decide if the *microscopic level policy*



**Fig. 2.** Chain selection and execution. The letters denote different lanes.

should be changed and, in that case, we proceed by probabilistically selecting it on the basis of the current traffic status. This concludes the selection of the chains. The criteria we use for the selection depends both on the average speed of cars in the incoming lanes in the intersection and on how long these vehicles have been waiting.

Phases are also distinguished between *decisional* and *transient*. The former can have their duration varied between a minimum and a maximum duration time while the latter must be executed for a predetermined amount of time. Transient phases are needed because in some conditions it is not possible to extend a phase. This happens when we have a yellow light phase, whose duration is decided by regulations. The duration of a *decisional* phase is determined by the logic of the currently selected microscopic level policy.

## 4.2 Macroscopic Level “Swarm” Policy

The goal of the macroscopic level policy, named *Swarm*, is to decide the most appropriate microscopic level policy according to the actual sensed traffic conditions. Since there is some uncertainty in measuring the traffic, e.g. which situations should be interpreted as “high” traffic and which ones should be “low”, we rely on the natural metaphor of the *pheromone*.

**Pheromone Levels.** Measuring the traffic is a hard task since there are numerous variables that must be considered, e.g. the number of lanes, the geometry of the intersection and the vehicles’ paths. A good measure should also be insensitive to sudden short spikes and be able to react rapidly to more persistent changes. Our idea is to use the natural metaphor of the pheromone to abstract the traffic.

Natural pheromone is an olfactory trail left by some animals. For example, ants leave a pheromone trail from their nest to a source of food along the shortest path between these two points. Pheromone has two interesting characteristics. The first one is that it is additive, so the more ants walk along the same path, the higher the value of the pheromone is. The second characteristic is that pheromone also evaporates over time, this is why ants are able to determine which is the shortest path from their nest to the food: if there are two paths, the shortest one will obviously be covered in less time so less pheromone will evaporate than on the other path.

In our implementation, pheromone is proportional to the difference between the maximum allowed speed on a lane and the average speed of the sensed vehicles, limited to 0. This way, the faster the cars, the smaller the values of pheromone, and vice-versa. This approach ignores the number of cars and it is supported by the speed-density relation of the fundamental diagram of traffic flow [17], which simply says that the number of cars does not count as long as the traffic is flowing freely.

We calculate an average value of the pheromone on the *incoming lanes* (i.e.  $\varphi_{in}$ ), the ones that enter the controlled intersection, and another average value

of the pheromone on the *outgoing lanes* (i.e.  $\varphi_{out}$ ), which are the ones that come from the controlled junction.

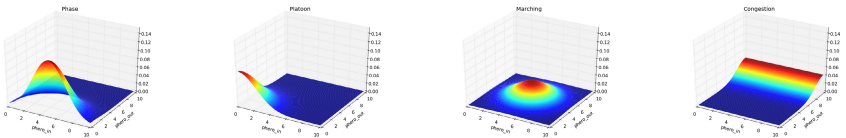
Note that, averaging the pheromone value, the system cannot be aware of the non homogeneity of traffic density. To avoid this potential drawback we introduce a new pheromone level (i.e.  $\varphi_{dsp\ in}$ ) proportional to the dispersion of the information of the traffic over the incoming lanes. This allows the selection of microscopic policies working better either for homogeneous or non homogeneous traffic conditions. These three pheromone values serve as input for the *stimulus functions*.

**Stimulus Function.** The stimulus functions, or *stimuli*, are functions associated with the microscopic level policies used to probabilistically determine which policy is more appropriate according to the current traffic conditions. They map their associated policy in a  $\varphi_{in} \times \varphi_{out} \times \varphi_{dsp\ in}$  space. This mapping is determined experimentally or via automatic parameter tuning and is used by the Swarm policy to probabilistically select the proper policy given the pheromone on the incoming lanes ( $\varphi_{in}$ ), the pheromone on the outgoing lanes ( $\varphi_{out}$ ) and the pheromone dispersion over the incoming lanes ( $\varphi_{dsp\ in}$ ).

The idea behind the stimulus function is that the more desirable the associated policy the higher the stimulus should be, given the current pheromone values.

The shape used for the stimulus function is obtained by considering the maximum value of a family of Gaussians. The use of more than one function allows the definition of multiple areas in the pheromone space (i.e. different traffic conditions) where the policy performs best, centering each Gaussian on a different area. As stated before, the parameters defining the shape of the Gaussians are computed off-line via ad-hoc experiments or automatic parameter tuning. The same policy can perform well in different traffic conditions depending on the characteristics of the controlled intersection, so the associated stimulus function may have different parameters for different agents.

The stimulus function must be normalized over its domain. This formulation makes the stimulus function similar to a probability density function, which is important since it expresses the level of specialization of a policy: we want high stimuli in the neighborhood of specialization and a rapid decrease outside it. Figure 3 plots the Gaussians of the stimulus functions for each policy considering only a single function to improve the readability (see Section 4.2 for details).



**Fig. 3.** Stimulus functions of the microscopic level policies. From left to right: Phase, Platoon, Marching and Congestion.

**Policy Selection.** The choice of the policy occurs after the evaluation of the stimulus functions in a probabilistic selection fashion. The probability  $P(i, j)$  of the  $i$ -th agent to choose the  $j$ -th policy for its intersection is determined by the following equations:

$$P(i, j) = \frac{T_{\theta,j}(s_{i,j})}{\sum_j T_{\theta,j}(s_{i,j})} \quad (1)$$

$$T_{\theta,j}(s_{i,j}) = \frac{s_{i,j}^2}{s_{i,j}^2 + \theta_{i,j}} \quad (2)$$

Note that the stimulus functions  $s_{i,j}$  are not taken into account directly in the calculation of the probability  $P(i, j)$ .  $\theta_{i,j}$  is the sensitivity threshold above which the  $i$ -th agent adopts the  $j$ -th policy. The sensitivity threshold  $\theta_{i,j}$  represents the level of sensitivity of the agent to the adoption of that policy. This threshold is variable in time, decreasing if a policy is selected in a stable way, and increasing in time as a policy is not selected. This is called *reinforcement* because as a policy is selected and found to be working well, it is learned and its probability to be selected again (stable policy) increases. At the same time, even a policy that has a nearly-zero stimulus function will always have a non-zero probability to be selected, because the threshold cannot go above a maximum  $\theta_{max}$  value, as well as a policy that is stably selected for a long time will not be reinforced too much, as the threshold cannot go below a minimum  $\theta_{min}$  value. These two values are also bounded:  $0 \leq \theta_{min} < \theta_{max} \leq 1$ .

The new policy is selected in a probabilistic way: the higher the stimulus function for a policy, the higher the probability that policy will be selected. According to the model given by [18], at each step in simulation, the pheromone levels of the controlled incoming and outgoing lanes are updated and at the end of a commit phase the Swarm selects a new microscopic level policy for execution. The event of selecting another policy is also probabilistic, with probability  $0 < p_{change} \leq 1$ . It is clear that if  $p_{change} = 1$  the process of policy selection occurs every time a commit phase is reached, making the agent highly susceptible to traffic density variations. This probability should be sufficiently low to mitigate the indecision of the agent and to guarantee its reactivity at the same time.

### 4.3 Microscopic Level Policies

Microscopic level policies take short-term decisions. Usually, they operate on a base sequence of stages and apply variations to this base sequence. The most common decisions taken by an adaptive policy concern the duration of the green phase for different lanes, or including/excluding parts of the whole base sequence.

However, most traffic-dependent policies will always cycle the signal sequence in a pre-fixed order. We have already presented the concept of *chains* in Section 2 and we mentioned the existence of some particular phases called *decisional*. The duration of those phases changes according to the logic of the currently selected microscopic level policy. We are now going to present the implemented



microscopic level policies<sup>3</sup>. Most of them make use of a so-called *traffic threshold* as one of the conditions used to end a decisional phase. This is a simple threshold applied to the same indicator we use to determine which chain should be selected.

**Phase Policy.** This policy will terminate the current chain as soon as another one has reached the traffic threshold, respecting the minimum duration constraint of the decisional phases. In case no other chain wants to be activated, i.e. there is no traffic opposing the current green directions, the current decisional phase is kept indefinitely, regardless of its maximum phase duration.

```
bool canRelease(int elapsed, bool thresholdPassed,
    const MPhaseDefinition* stage, int vehicleCount){
    if(elapsed >= stage->minDuration)
        return thresholdPassed;
    return false; }
```

This policy is adequate in medium-low traffic situations, where this early termination will not make the traffic lights switch too often.

**Platoon Policy.** This policy will try to let all the vehicles in the current green directions pass the intersection before releasing the green light. It is similar to *Phase*: as before, a chain is executed in respect of every minimum duration of the decisional phases and until there is another one above the traffic threshold. We also have a third condition: the decisional phase must be executed for its maximum duration time or there must be no other vehicle incoming from the allowed directions.

```
bool canRelease(int elapsed, bool thresholdPassed,
    const MPhaseDefinition* stage, int vehicleCount){
    if(elapsed >= stage->minDuration)
        if (thresholdPassed)
            return ((vehicleCount == 0) ||
                (elapsed >= stage->maxDuration));
    return false; }
```

This behavior allows the creation of platoons of vehicles, which is positive since it helps to handle the traffic. In intense traffic situations, each decisional phase would be executed for the maximum allowed time, so its definition has a great impact on the performance of the system.

**Marching Policy.** This policy is adequate when the traffic looks too intense from all directions to take any on-line decision regarding only the incoming lanes. In this case, there are two possible approaches: either use a static duration for decisional stages or consider also the outgoing lanes, not allowing traffic to lanes that are too heavily loaded. This second case may suggest a different, more complex, way to select the chain to execute.

<sup>3</sup> These policies were presented in [5].

```

bool canRelease(int elapsed, bool thresholdPassed,
    const MSPhaseDefinition* stage, int vehicleCount) {
    return (elapsed >= stage->duration); }

```

**Congestion Policy.** This policy is used when the outgoing lanes are saturated and there may be vehicles waiting inside the intersection. In order to avoid gridlocks, all input lanes are inhibited, i.e. the current executing chain is terminated following the pre-defined plan to the commit phase, then no other chain is activated until the congestion has been solved. In doing this, every decisional phase is executed only for their minimum duration time.

As soon as the outgoing lanes are emptying, the pheromone levels will drop allowing Swarm to select another microscopic level policy. This will also enable again the chain selection.

**Decay Threshold.** With very low penetration rate configurations some of the microscopic level policies may reach a deadlock situation in which there is no chain change. This issue arises whenever the system can not detect any existing vehicles (due to the low penetration rate) on a lane served with red light. The unseen vehicles may wait for green light for a long period of time.

We solved this issue introducing a dynamic traffic threshold, called *decayThreshold*, working as a trigger for the chain selector. The threshold is based upon an exponential decay; it decreases at a rate proportional to its current value following the equation:

$$N(t) = N_0 \exp \frac{-t}{\tau} \quad (3)$$

where  $N(t)$  is the quantity at time  $t$  and  $N_0$  is the initial quantity, that for our setup was 1. The exponential value  $\frac{-t}{\tau}$  is a parameter and should be tuned. This threshold is used in condition that, if satisfied, triggers a chain change.

```

if(random > (1 - decayThreshold))

```

The control is probabilistic. It randomly chooses a number between  $[0, 1]$  and it checks the condition. In this way the triggering time is always different, even though it is influenced by the exponential decay value.

## 5 Evaluation

Several traffic simulators exist in the literature [19]; in COLOMBO we implemented and tested the system inside the microscopic traffic simulator SUMO [20]. The aim of this experimental section is to evaluate the performance of the proposed system, comparing the algorithm with a static and a fully-actuated approach (both implemented in SUMO). We also investigated how our method performs for different penetration rates of equipped vehicles.

The experiments are run on the penetration ratios of 100%, 50%, 25%, 10%, 5%, 2.5% and 1%. These different configurations are obtained not by modifying the number of simulated cars, which would provide unrealistic results, but by assigning to the shadow cars<sup>4</sup> a special SUMO type which allows the simulation of all vehicles but considers only the normal cars when calculating the input information used by the Swarm controller.

We developed a synthetic scenario, composed by 16 traffic lights arranged on 4 by 4 grid, built with the aim to simulate critical traffic conditions. The four central traffic lights are controlled by the Swarm policy while the others use an actuated policy. The structure of each traffic light is based on the German “Richtlinie für Lichtsignalanlagen” (Guidelines for traffic light systems), or RiLSA for short [21]. Simulation have been run using different penetration ratios and measuring how the average waiting time of the vehicles varies (lower values mean better results).

The evaluation scenario is a 4 by 4 grid composed by four horizontal streets and four vertical streets, forming sixteen intersections. The distance between two consecutive neighboring crossing is about 500m. Every street is defined as a single lane, which splits into two lanes near each intersection. The outer traffic lights are used to regulate the traffic flow for the Swarm controlled junctions. This structure gives a good approximation of a real world scenario where the controlled traffic lights are placed in an urban context. The Swarm controllers use the same configuration, however the control of each intersection is independent from the others. The configuration is obtained through off-line automatic parameter tuning executed for every penetration ratio on a simplified road network composed of a single cross-like RiLSA intersection controlled by a Swarm agent with actuated traffic lights at each lane.

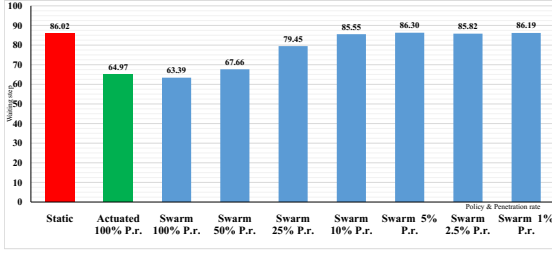
Each traffic light controller implements two chains: one that gives green to the south-north and north-south directions and a second one that gives green to the perpendicular directions. Each chain has two green decisional phases: one gives green to the straight, right and left turn directions with by a minimum duration of 10s, a maximum duration of 50s and a default duration of 31s; the other is dedicated to the left turns direction only with a minimum duration of 2s, a maximum duration of 20s and a default duration of 6s. Each chain also has a yellow transient phase of 4s and finally a red commit phase of 4s.

We adopted a traffic generator to create different simulations. Each simulation consists in traffic flows which resembles the traffic patterns as occurring in the real-world by taking into account different realistic daily load curves. The average vehicles per hour of the flows obtained is about 4500v/h.

Figure 4 depicts the performance of the different configurations. For each simulated configuration (x-axis) the figure shows the mean waiting steps (y-axis) obtained by averaging 100 different simulations. Table 1 reports the values outlined in the figure.

---

<sup>4</sup> The vehicles that do not have to be considered on a set penetration ratio. E.g. In a simulation with 20% penetration ratio the shadow vehicles will be the 80%.

**Table 1.** Average waiting steps.

Configuration	Wait steps
Static	86.02
Actuated 100% p.r.	64.97
Swarm 100% p.r.	63.39
Swarm 50% p.r.	67.66
Swarm 25% p.r.	79.45
Swarm 10% p.r.	85.55
Swarm 5% p.r.	86.30
Swarm 2.5% p.r.	85.82
Swarm 1% p.r.	86.19

**Fig. 4.** Average waiting steps for the different configurations.

It is worth mentioning that only our proposal is affected by the low penetration rate of equipped vehicles, since the actuated system relies on inductive loops and the static approach does not sense cars.

At full knowledge our system is slightly better than the actuated approach and it outperforms the static one. Note also that the Swarm algorithm is comparable with the actuated even with 50% penetration rate (which means that our system detects only half of the vehicles).

With very low penetration rates (10% or lower) our system is comparable with the static one. It is finally interesting to highlight that our system performance with very low penetration rates does not degrade with the decrease of the penetration rate.

## 6 Conclusions

This paper presented a Swarm-based Traffic Lights control system in which every intersection controller makes independent decisions to pursue common goals and is able to improve global traffic performance. This solution is low cost and widely applicable to different urban scenarios. This work is developed within the COLOMBO european project.

The promising results presented in Section 5 show that the proposed approach performance is comparable to more sophisticated systems, like the fully-actuated one (which has full knowledge using the inductive loops detectors). Moreover the experiments showed that the system performance does not degrade depending on the percentage of detectable vehicles.

As future work we plan to (1) adapt the system to include interactions with pedestrians and public transportation and (2) test it on real world scenarios.

## References

1. Chen, Y., Richard Yu, F., Zhou, B.: Improving throughput in highway transportation systems by entry control and virtual queue. In: Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, DIVANet 2013, pp. 9–14. ACM, New York (2013)

2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems, vol. 1. Oxford University Press (1999)
3. Ducatelle, F., Di Caro, G.A., Gambardella, L.M.: Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence* **4**(3), 173–198 (2010)
4. Ferrante, E., Sun, W., Turgut, A.E., Dorigo, M., Birattari, M., Wenseleers, T.: Self-organized flocking with conflicting goal directions. In: *Proceedings of the European Conference on Complex Systems 2012*, pp. 607–613. Springer (2013)
5. Slager, G., Milano, M.: Urban traffic control system using self-organization. In: *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 255–260 (2010)
6. Gershenson, C.: Self-organizing traffic lights (2004). arXiv preprint [nlin/0411066](https://arxiv.org/abs/nlin/0411066)
7. de Oliveira, D., Ferreira Jr, P.R., Bazzan, A.L.C., Klügl, F.: A swarm-based approach for selection of signal plans in Urban scenarios. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) *ANTS 2004. LNCS*, vol. 3172, pp. 416–417. Springer, Heidelberg (2004)
8. Seredynski, M., Arnould, G., Khadraoui, D.: The emerging applications of intelligent vehicular networks for traffic efficiency. In: *Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, DIVANet 2013*, pp. 101–108. ACM, New York (2013)
9. Robertson, D.I.: Transyt: a traffic network study tool (1969)
10. Robertson, D.I., David Bretherton, R.: Optimizing networks of traffic signals in real time: The scoot method. *IEEE Transactions on Vehicular Technology* **40**(1) (1991)
11. Sims, A.G.: The sydney coordinated adaptive traffic system. In: *Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems*, 1979, Pacific Grove, California, USA (1979)
12. Peek Traffic. Utopia/spot-technical reference manual. Peek Traffic, Amersfoort, The Netherlands, Tech. Rep (2002)
13. Priemer, C., Friedrich, B.: A decentralized adaptive traffic signal control using v2i communication data. In: *12th International IEEE Conference on Intelligent Transportation Systems, ITSC 2009*, pp. 1–6. IEEE (2009)
14. Mizuno, K., Fukui, Y., Nishihara, S.: Urban traffic signal control based on distributed constraint satisfaction. In: *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pp. 65–65. IEEE (2008)
15. Bazzan, A.L.C.: Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems* **18**(3), 342–375 (2009)
16. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **38**(2), 156–172 (2008)
17. Greenshields, B.D., Channing, W.S., Miller, H.H., et al: A study of traffic capacity. In: *Highway research board proceedings*, vol. 1935. National Research Council (USA). Highway Research Board (1935)
18. Theraulaz, G., Bonabeau, E., Deneubourg, J.N.: Response threshold reinforcement and division of labour in insect societies. *Proceedings of the Royal Society of London. Series B: Biological Sciences* **265**(1393), 327–332 (1998)

19. Shafiee, K., Lee, J.B., Leung, V.C.M., Chow, G.: Modeling and simulation of vehicular networks. In: Proceedings of the First ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, DIVANet 2011, pp. 77–86. ACM, New York (2011)
20. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements* **5**(3&4), 128–138 (2012)
21. FGSV Verlag Forschungsgesellschaft fuer Strassen-und Verkehrswesen. RiLSA - Richtlinien fur Lichtsignalanlagen - Lichtzeichenanlagen fur den Strassenverkehr. FGSV (1999)

AI\*IA 2015 Advances in Artificial Intelligence  
XIVth International Conference of the Italian Association  
for Artificial Intelligence, Ferrara, Italy, September  
23-25, 2015, Proceedings  
Gavanelli, M.; Lamma, E.; Riguzzi, F. (Eds.)  
2015, XIV, 480 p. 118 illus. in color., Softcover  
ISBN: 978-3-319-24308-5