LaNPro: Non Stop Driving Thru Low Traffic Intersections

Cristiano M. Silva[1,2], Andre L. L. Aquino[3], Wagner Meira Jr.[1]

[1] Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

[2]Universidade Federal de São João Del-Rei (UFSJ)
Ouro Branco – MG – Brasil

[3]Universidade Federal de Alagoas (UFAL)
Maceió – AL – Brasil

cristiano@ufsj.edu.br, alla@ic.ufal.br, meira@dcc.ufmg.br

Abstract. *LaNPro is a vehicular client-server application that avoids the stop of vehicles at traffic lights in low traffic conditions. The server-side of the application is deployed as a module of a smart traffic light. It senses the presence of vehicles along the road through radars, cameras, road sensors and Wi-Fi communication to assign the right of way. The client-side of the application runs inside the vehicle's on-board unit. Results (via simulation) show that the application can ensure the non-stop drive thru of intersections that have an expected traffic volume equal or less than* $\lambda = 0.10$ *vehicles per second, assuming intersections of 2, 3 and 4 lanes. This works presents strategies to deal with the allocation of vehicles in low traffic intersections and presents the scenario where this technique can be used.*

1. Introduction

The traffic light is the device responsible for ensuring the right of way, ordering the flow of vehicles and avoiding accidents.While the social geography of urban centers changes over time, it imposes new restrictions on the flow of vehicles, which often are not reflected in the signal programs, which may cause congestions in urban centers[Koonce et al. 2008, Silva et al. 2014]. It is estimated that only the update of the traffic signals programming in the US is able to reduce travel time by almost 40%, saving over 640 billion gallons of fuel per year[Harris 2005].

The scientific community has been searching for solutions to improve the displacement of citizens in several areas of traffic engineering, including the use of **smart traffic lights**. These devices are capable of auto configuration and monitor the flow along the roads segments, seeking to organize the system [Helbing et al. 2005] and minimize the global wait time of drivers [Liu 2007, Balan and Luke 2006, Houli et al. 2010]. Most of the research in smart traffic lights targets to avoid/minimize congestions. Meanwhile, in this work we present a different approach: we address the flow in **very low traffic conditions** usually found in small cities and during the early hours of morning (dawn).

The main motivations for our proposal, called LaNPro, are: (i) avoid situations where the driver is stopped on a red light alone, with no vehicle competing for the intersection; (ii) reduce the risk of robberies: while the driver concentrates on the light, he/she becomes an easy victim for robberies.

LaNPro[1] is a traffic signal module that schedules vehicles to drive thru low traffic intersections. The complete design of LaNPro demands the smart traffic light to communicate with other traffic lights, and wireless communication with vehicles, road sensors, cameras and radars. Vehicles must be equipped with a GPS[2] and onboard units capable to communicate with the traffic lights.

In this article we propose experiments that investigate intersections composed of two road segments, each road segment made of one or two lanes. We study the behavior of the vehicles when the intersections are: (i) unsigned (no traffic lights); (ii) signed with static traffic lights; (iii) signed with LaNPro. Our goal is to define the context where LaNPro becomes an interesting solution and the pros and cons of our approach. We vary the level of traffic and measure the expected number of collisions, size of vehicle queues and maximum level of traffic that LaNPro routes without vehicles stops. Our results show that LaNPro is able to eliminate vehicle stops in intersections composed of road segments with an expected arrival of vehicles $E(x) = \lambda$, where $\lambda \leq 0.10$ vehicles per sec., supposing segments of 200 m long, intersections 10 m wide, vehicles 5 meters long where vehicles travel in an average speed of $\mu = 40$ km/h with a std dev $\sigma = 4$ km/h.

The main contributions of this work are: (i) demonstrate how to handle low traffic intersections in a non-stop approach; (ii) propose a strategy to reduce the risks of assaults/robberies of drivers; (iii) present two models to switch the traffic lights in this dynamic scenario. Along this text we present LaNPro and the set of experiments that has supported our conclusions. The modules of LaNPro are intentionally described with variable details, taking into account the article needs and patent request limitations.

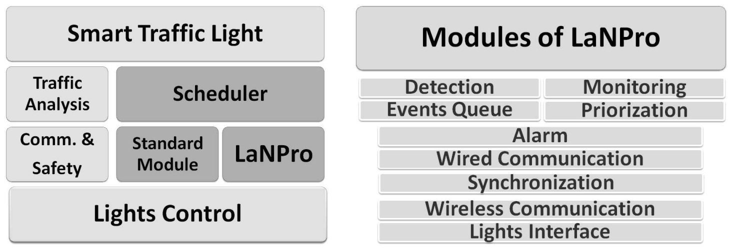
The article is organized as follows: Section 2 overviews traffic lights and related work. Section 3 introduces LaNpro. Section 4 presents the main functionalities of LaNPro. Section 5 present the experiments, and Section 6 concludes the document.

2. Related Work

Traffic lights can be classified into **static** or **dynamic** (smart). The main characteristic of **static traffic lights** is the pre-programming. That means that its operation and the duration of each phase is defined in advance[Liu 2007]. In order to improve its operation, static traffic lights may also employ several static programmings that are scheduled along the day. However, since the control of traffic is non-linear and non-deterministic[Liu 2007], static methods may not achieve high performance. This has motivated the development of **dynamic traffic lights** that continuously adapt to the traffic demands using fuzzy logic[Chiu and Chand 1993], artificial intelligence, neural networks[Shihuang 1992], genetic algorithms[Foy et al. 1992] or reinforced learning[Kaelbling et al. 1996].

The V2I communication applied to smart traffic lights is covered in, at least, two works involving the industry: **BMW Green Wave Project**[3] demonstrates that smart traffic lights reduce up to 20% fuel consumption. **Audi Travolution**[4] proposes the use of smart traffic lights combined with vehicular communication to implement speed advisory

[1]Patent: BR 10 2013 027283 3
[2]Global Positioning System.
[3]http://www.themotorreport.com.au/46137/
[4]http://www.audi.com/.



(a) Traffic signal. (b) LaNPro.

Figure 1. Architecture: (a) traffic light: scheduler selects between standard module and LaNPro according to the level of traffic. (b) Modules of LaNPro.

that minimizes stops at traffic lights. The results show the economy of 0.02 liters of fuel per traffic light. **LaNPro** differs from all these proposals in the sense that it focuses on the management of intersections under low traffic conditions.

3. Overview of LaNPro

LaNPro (**La**te **N**ight **Pro**gramming) implements an algorithm that routes vehicles over low speed traffic intersections, managing these vehicles to safely drive thru the intersection without stopping. Because it is a very specific operation, LaNPro must be combined with a **standard module** that will control the lights when in high traffic. Thus, the traffic light needs a scheduler that selects which module should be active: **standard module** or LaNPro, according to Fig. 1(a).

The traffic light starts being controlled by the **standard module**. When the traffic light detects low traffic (idleness) in the intersection, LaNPro takes over the control of the traffic light and turns the lights to blinking yellow (flash). Let's suppose an intersection of roads r_1 and r_2. LaNPro starts monitoring both segments of roads (r_1 and r_2) through a set of road sensors, cameras, and radars. When it detects any vehicle traveling on one of the roads (let's say r_1) heading to the intersection, LaNPro communicates to other smart lights, gives a green light to r_1 and a red light to r_2. As soon as the vehicle (v_1) traverses the intersection, LaNPro returns the lights to blinking yellow if there are no cars willing to traverse the intersection.

After LaNPro detects (v_1), a second vehicle (v_2) is also detected, but now on road r_2. In this case, LaNPro will communicate to its traffic light pairs, define the priorities of the vehicles to drive thru the intersection, reach an agreement with all traffic light involved and communicate the decision to the vehicles (v_1 and v_2), defining the speed that each one needs to keep so both can drive thru the intersection without stopping. The width of segments and the speed of vehicles are important parameters because they define the safety margin that LaNPro has to route the vehicles.

3.1. Motivations

LaNPro follows a different approach from typical smart traffic lights, which are usually concerned in managing the flow in high traffic scenarios. LaNPro **is designed for low traffic** management, routing vehicles across the intersection in a non-stop way. LaNPro is designed to solve situations where a driver is stuck in a red light, alone, with no car in sight, and he/she has to wait for a green light. Unfortunately, nowadays traffic lights are not smart enough to detect this situation and just let the driver go on. LaNPro is not a futuristic mechanism. Instead, we envision LaNPro to help the society right now, as soon as possible. Probably, in the future we will not even need traffic lights: cars can communicate and define the right of way by themselves. But, it is still a far reality, so we believe that LaNPro has much to contribute for the next ten years.

Another point is: any smart traffic light can address the low traffic. But they are no specialized devices to do this. Every time we think about improving the traffic, we think about rush hours and traffic jams. The main differential of our work is to improve low demand traffic, and, as we will explain along this paper, this involves several challenges. We have searched for similar solutions in every bank of patent in the world, and we have not found any solution like this one.

4. Main Functionalities of LanPro

The organization of LaNPro is shown in Fig. 1(b). LaNPro has a **module to detect new arriving vehicles**. When a vehicle is detected, the **monitoring module** is activated to follow the vehicle until it traverses the intersection. The **lights control module** is responsible for controlling the lights. It has a **sub-module responsible for scheduling the vehicles** along the intersection. In its simplest form, the first vehicle to reach the intersection is prioritized[5]. After defining the priority of vehicles, all traffic lights communicate using the **wired communication module**, the lights are set and the vehicles are informed via the **vehicular module** of the speed and the slot of time (detailed in Section 4.2) to traverse the intersection. The monitoring module continues to monitor the position and speed of each vehicle. If any vehicle tries to cheat (vehicle doesn't respect the traffic light commands and speeds up to seize/force a green light), all traffic lights communicate between each other and give red light to all roads. Then, standard module is scheduled to manage the intersection.

4.1. Evaluation of the Collision Risk

The evaluation of the collision risk assuming that only one vehicle is present in each road segment in a intersection of two roads is (almost) simple. We must make sure that both vehicles will not reach the intersection at the same time, considering a **safety margin**. The safety margin indicates the difference in time (or distance) between two consecutive passages of competing cars through the intersection. This margin is important because it mitigates small speed variations suffered by vehicles along the displacement, as well as measurement errors of velocity and position of each vehicle. Keeping the same speed is not trivial because of steepness, holes and unpredictable events.

4.2. Allocation of the Intersection

Although LaNPro deals with situations of low traffic, the assumption that only one vehicle is present in each road segment is unreasonable. There are some alternatives that could be implemented to allow the system to handle multiple vehicles, designed by our team: (i) slow and (ii) fast switching.

slow switching: the traffic light tries to traverse as much vehicles as it can, until the first competitor vehicle arrives at the intersection. Then it reverses the phases. The

[5] Advanced versions of LaNPro can prioritize emergency vehicles.

competitor vehicles are advised to travel at the minimum speed allowed on the road[6]. This strategy has the advantage of being closer to current traffic signal standards.

fast switching: the traffic light schedules vehicles individually to traverse the intersection through the attribution of time slots. Whenever a vehicle is detected, it receives a slot considering the expected time to reach the intersection. In our experiments, we have adopted fast switching because that is a more challenging strategy. In the next section we show a high level version of the LaNPro algorithm.

4.3. LaNPro Algorithm

In this section we present LaNPro's pseudo-algorithm. We assume road segments having the same length and that cars update their speed instantaneously. We use 'pairs' to indicate the set of traffic lights responsible for an intersection. Although LaNPro receives several parameters as input, such as the distance of road sensors, number of road sensors, speed of the road, quality of the floor, special conditions (blind crossing), in the text we concentrate on its two most important parameters. LaNPro receives as input:

λ: threshold of low traffic;
η: maximum size of the vehicle queue.

Algorithm 1 LaNPro Algorithm.

```
Input: λ, η;
 1: turn lights to blinking yellow
 2: hand-shake with pairs to start operation
 3: while (level ≤ λ and queue < η and not alarm) do
 4:     start thread T1 to detect new vehicles
 5:     start thread T2 to synchronize the change of lights according to the vehicle queue
 6:     if (vehicle detected) then
 7:         create a ticket containing the expected arrival time
 8:         start thread T3 to monitor the position and speed of the vehicle
 9:         forward the ticket to pairs
10:         receive acknowledgements from pairs
11:         event ← true
12:     end if
13:     if (received ticket from pair) then
14:         acknowledge the ticket
15:         store the expected arrival time and vehicle id
16:         event ← true
17:     end if
18:     if (event) then
19:         forward event to thread T2
20:     end if
21:     if (thread T1 detects high traffic) then
22:         communicate scheduler and pairs requesting unscheduling
23:         receive acknowledgement from scheduler
24:     end if
25:     if (thread T3 detects cheating) then
26:         forward alarm to thread T2 requesting red light
27:         communicate scheduler and pairs requesting unscheduling
28:         receive acknowledgement from scheduler
29:     end if
30: end while
31: handshake with pairs to finish operation;
```

Once LaNPro is scheduled, it turns the lights to blinking yellow (line 1) and handshakes with its pairs to negotiate the start of the operation (line2). LaNPro is always monitoring the level of traffic, so it is active while the level of traffic is below some λ and

[6] Defined by the transit authority.

the queue of vehicles is below η, both input parameters, or it has not detected any alarm (line 3). LaNPro starts a thread T_1 to detect new vehicles (line 4) and another thread to operate the lights T_2. T_2 works as a consumer process. Requests of vehicles are inserted into a queue and T_2 assigns slots of time to each vehicle, and makes an agreement with its pairs, so every traffic light is always synchronized. When there is no request in the queue, T_2 returns the traffic lights to blinking yellow. If T_1 detects a vehicle (line 6), it creates a ticket containing the expected arrival time at the intersection (line 7), starts thread T_3 to monitor the speed of the vehicle until it crosses over the intersection (line 8). The ticket is broadcast to all traffic lights (line 9) and acknowledgements are received (line 10). The algorithm records the pending event (line 11).

If LaNPro receives a ticket from another traffic light (line 13), it acknowledges the ticket (line 14), stores it (line 15) and records a pending event (line 16). If there is a pending event (Line 18), LaNPro forwards it to thread T_2 (line 19), the thread responsible for negotiating the ordering of vehicles and setting up the lights. If thread T_1 (line 21) detects queue $> \eta$ or level $> \lambda$, it requests the finalization of LaNPro (line 22). After the acknowledgement of the scheduler (line 23), the traffic lights hand-shake the end of operation (line 31). If thread T_3 (line 25) detects that some vehicle is not following the agreement, it forwards an alarm to T_2 requesting an immediate red light (line 26), communicates its pairs and scheduler (line 27), receives the acknowledgement (line 28) and finalizes LaNPro (line 31). Due to space limitations, we will not show the algorithms of the scheduler and threads T_1, T_2 and T_3.

5. Experiments and Results

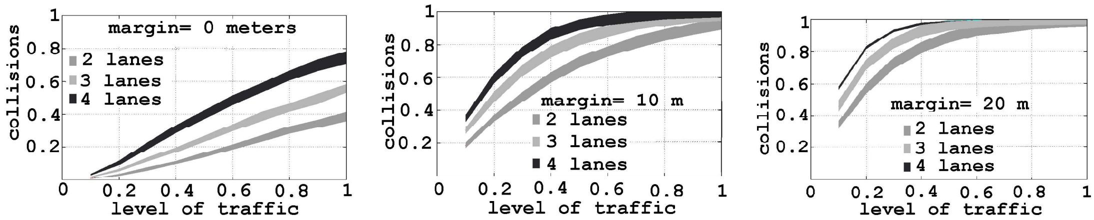
This section presents the models generated and the results obtained through simulations in order to validate the potential contribution of LaNPro.

5.1. Methodology

The flows of vehicles are modeled via a **Poisson distribution** supported by [Gerlough and Schuhl 1955, McNeil 1968], which expresses the probability of a series of events occuring in a certain period of time, since these events are independent of when the last event occurred. Because we are dealing with low traffic, Poisson seems quite reasonable. The vehicle speed was modeled based on a **Gaussian distribution** with parameters μ=40 km/h and σ=4 km/h. Segments of roads are 200 meters long, intersections are 10 meters wide. The results presented are based on the results of 100 simulations, which run for 5000 seconds. In order to validate LaNPro we build a simulator containing one single intersection consisting of 2, 3 and 4 lanes. The simulation is composed of 3 categories of experiments:

- **(i) intersection without traffic light:** we measure the number of collisions in blind traverse of vehicles with/without safety margins for different levels of traffic;
- **(ii) intersection with a fixed programming traffic light:** we measure the number of vehicles stops in red lights for different levels of traffic;
- **(iii) intersection with LaNPro:** we measure the number of LaNPro interruptions and the queue size of vehicles, also for different levels of traffic.

Below we detail each experiment.



(a) Collisions (no safety margin). (b) Collisions (margin=10 m). (c) Collisions (margin=20 m).

Figure 2. Probability of collisions for intersections composed of 2, 3 and 4 lanes as we change the safety margin (0 meters, 10 meters and 20 meters).

5.2. Intersection without traffic light: collisions in blind traversal with no safety margin

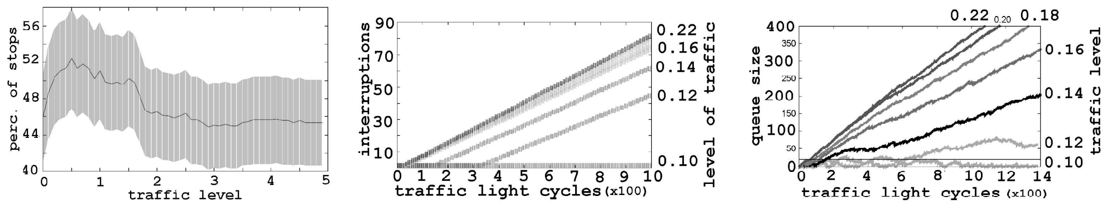
In this experiment we evaluate the probability of collisions of vehicles traversing intersections in a non-stop and blind way. Basically, the drivers maintain their route independently of any other variable (Russian roulette). We present results for an intersection composed of 2 road segments, each road segment having 1 or 2 lanes, as we change the level of traffic. Fig. 2(a) indicates the probability of collision considering the same level of traffic on the roads under study. The x-axis indicates the level of traffic from $0 \leq \lambda \leq 1$ while the y-axis indicates the probability of collision. Curve 'A' indicates an intersection between 2 single lane roads. Curve 'B' indicates intersection between a single lane and a double lane road. Curve 'C' indicates intersection between 2 double lane roads. The width of each curve indicates the standard deviation.

We consider as a time unit the average time required for a vehicle to traverse the entire intersection. We can notice that when the traffic level is $\lambda = 1.0$ we achieve A=40%, B=60% and C=80% of collisions. In practical terms, this means that considering an intersection 10 meters wide, density of vehicles less than 1 vehicle per 10 meters, just 40% of the vehicles collide if we consider 2 single-lane roads; 60% collide if we consider 3 lanes; 80% collide in 4 lanes. Notice that 1 vehicle per 10 meters is not exactly what we mean by low traffic. Nevertheless, this result demonstrates that if we introduce some kind of ordering, we can manage the non-stop traversal of intersections efficiently (in Section 5.5 we demonstrate that LaNPro can manage the traversal of intersections up to $\lambda = 0.1$ vehicles per unit of time, but we still have more to discuss until we get there).

However, the cars are operated by humans and we cannot allow vehicles to cross over the intersection without a minimum distance/delay between them. We call this distance between two vehicles as **safety margin**. In the next experiment we measure the number of collisions considering variable safety margins.

5.3. Intersection without traffic light: collisions in blind traverse with variable safety margin

Basically, this is the same simulation as the previous one. The difference is that we add safety margins. If any vehicle enters the intersection in a distance less than some threshold, we consider the event as a collision. We arbitrarily defined the safety margins as 10 meters and 20 meters. In Fig. 2 the x-axis represents the level of traffic from $0 \leq \lambda \leq 1$. The y-axis indicates the collisions probability. The graph presents the same



(a) Stops in static traffic lights. (b) LaNPro interruptions. (c) Queue size with LaNPro.

Figure 3. (a) low traffic incurs in increased traffic light stops. (b) LaNPro routes all traffic $\lambda \leq 0.10$. (c) LaNPro presents tiny queues for $\lambda = 0.10$, being able to route all traffic.

combinations of intersections. We can notice that the volume of collisions increases very fast as we increase the level of traffic and the safety margin. As mentioned before, LaNPro manages traffic levels up to $\lambda = 0.10$ considering a safety margin of 20 meters. The contribution of LaNPro is to eliminate all stops when $\lambda = 0.10$. When considering safety margin of 20 meters, 58% of the vehicles will collide if they attempt to drive through the intersection without stopping. In the next simulation we insert a fixed-programming traffic light in the intersection and observe the percentage of vehicles that receive red light.

5.4. Intersection with fixed programming traffic light: percentage of vehicles stopping in red lights

In this experiment the intersection receives a traffic light with fixed programming (same 1 minute for green and red) and we measure the percentage of vehicles that have to wait in red light. Both roads follow the same distribution of vehicles arrival. Thus, we would expect 50% of stops. Fig. 3(a) shows the results and the standard deviation. The x-axis represents the traffic level $0 \leq \lambda \leq 5$. The y-axis represents the percentage of vehicles that need to wait in red light.

As we can see, the symmetric timing shows very poor performance in low traffic situations. As the level of traffic increases, the percentage of stopped vehicles stabilizes around 45%. The standard deviation σ is indicated in the area around the main curve. The average stop time is $\mu = 28s$ with $\sigma = 8s$. That implies is safety risk for the driver, which becomes alone in the street for that period of time. That clearly represents a strong motivation for LaNPro, because in developing countries it represents a situation of real danger. In the next experiment we replace the static traffic light for LaNPro and we measure the number of times that LaNPro is able to route the vehicles in a non-stop way.

5.5. LaNPro interruptions

Now we are interested in evaluating the number of times that LaNPro is interrupted due to a high level of traffic or the impossibility to route the vehicles without stopping them at the traffic light. The simulation follows the exactly behavior of LaNPro: every time a vehicle stops, LaNPro is interrupted and the **standard module** is scheduled. LaNPro only returns when the traffic light detects idleness in the intersection. In order to reduce the simulation time, every time LaNPro is interrupted we flush the queue of vehicles and restart LaNPro. Our experiment considers the Brazilian laws of traffic: urban roads have a maximum speed of 40 km/h and a minimum speed of 20 km/h.

Fig. 3(b) shows the result of the experiment. The x-axis indicates the amount of traffic light cycles, while the y-axis shows the number of times that LaNPro was interrupted, i.e., some vehicle has to be stopped. The arrival rate of vehicles is $0,10 \leq \lambda \leq 0,22$. Rates below 0.10 have not generated any LaNPro interruption. Thus, in this scenario LaNPro is able to avoid the stops of vehicles for arrival rates until $\lambda = 0,10$.

Let's make this result more practical. We have assumed an intersection 10 meters long. Vehicles travel 40km/h (approx. 11 meters per second). Thus, the vehicle drives thru the intersection in approx. 1 s. The λ is computed considering the time duration that the vehicles need to drive thru the intersection. Considering all of this, the $\lambda = 0.10$ means that LaNPro is able to route vehicles if they arrive according to the distribution of Poisson with λ equal to 1 vehicle per 10 seconds, considering a safety margin of 20 meters. At a first sign it may seems a very easy job to route 1 vehicle every 10 seconds. However, the distribution of Poisson demonstrates that vehicles have a non-negligible probability of arriving in groups. In the last experiment we demonstrate how the queue of vehicles changes when LaNPro is managing the intersection.

5.6. Queue size of vehicles with LaNPro

In this experiment we evaluate the queue of vehicles when the intersection is controlled by LaNPro. Large queues make the LaNPro algorithm unfeasible, after all, when the road segment is fully occupied there is no room left to LaNPro manage the cars in order to route them across the intersection. The visualization of the queue size is also important to check LaNPro limitations. Fig. 3(c) presents a graph that demonstrates the queue of vehicles (the sum of all roads heading to the intersection) during the operation of LaNPro. The x-axis represents traffic light cycles (time), while the y-axis represents the queue size. The experiment is applied for arrival rates from $0.10 \leq \lambda \leq 0.22$. The graph indicates the queue size for each moment in time. The horizontal line indicates the queue of 100 meters long (remember that the queue represented is the sum of all queues in all road segments). We can notice that arrival rates above $\lambda = 0.10$ imply in queues, possibly leading to traffic jams.

6. Conclusion

LaNPro is a module of a smart traffic light designed to operate under low traffic conditions usually found in small cities and during the early hours of morning. The goal of LaNPro is to route the vehicles to drive thru the intersection without stopping in red light as it represents risk to drivers' safety and waste of time and vehicle resources. If the driver stops at the red light, he/she is on risk of robberies. If the driver chooses to ignore the red light and just crosses over the intersection, he/she is put on a even more dangerous situation. In this scenario, our solution is to manage the intersection, allowing vehicles to cross over it in safety through the assignement of slots of time. When getting closer to the intersection, each vehicle receives a slot (time window) to cross over the intersection, along with the speed that should be maintained.

We have searched for similar solutions in banks of patents around the world and we simply have found any solution similar to LaNPro. The technical challenge of LaNPro is to deal with variable instantaneous concentrations of vehicles that may happen, as ruled by the Poisson process. In this work we have studied the viability of LaNPro and we conclude that it can handle an arrival rate of 1 vehicle per 10 seconds supposing intersections of 10 meters, road segments 200 meters long, crossing involving 4 lanes and safety margin of 20 meters between each vehicle (slot size) considering **fast switching** (Section 4.2). While LaNPro imposes some interesting engineering challenges, it also provides improvements in the urban mobility. As a future work, we are investigating the pros and cons of low switching.

Acknowledgements

This work was partially supported by CNPq, CAPES, FAPEMIG, FAPEAL, InWeb, RNP CIA^2 and Winet.

References

Balan, G. and Luke, S. (2006). History-based traffic control. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 616–621. ACM.

Chiu, S. and Chand, S. (1993). Adaptive traffic signal control using fuzzy logic. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1371–1376. IEEE.

Foy, M. D., Benekohal, R. F., and Goldberg, D. E. (1992). Signal timing determination using genetic algorithms. *Transportation Research Record*, (1365).

Gerlough, D. L. and Schuhl, A. (1955). *Use of Poisson distribution in highway traffic*. Eno Foundation for Highway Traffic Control.

Harris, J. T. (2005). Benefits of retiming traffic signals. In *ITE 2005 Annual Meeting and Exhibit Compendium of Technical Papers*.

Helbing, D., Lämmer, S., and Lebacque, J.-P. (2005). Self-organized control of irregular or perturbed network traffic. In *Optimal Control and Dynamic Games*, pages 239–274. Springer.

Houli, D., Zhiheng, L., and Yi, Z. (2010). Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network. *EURASIP Journal on Advances in Signal Processing*, 2010:7.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *arXiv preprint cs/9605103*.

Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beaird, S., Braud, C., Bonneson, J., Tarnoff, P., and Urbanik, T. (2008). Traffic signal timing manual. Technical report.

Liu, Z. (2007). A survey of intelligence methods in urban traffic signal control. *IJCSNS International Journal of Computer Science and Network Security*, 7(7):105–112.

McNeil, D. R. (1968). A solution to the fixed-cycle traffic light problem for compound poisson arrivals. *Journal of Applied Probability*, 5(3):624–635.

Shihuang, X. D. F. J. S. (1992). A fuzzy controller of traffic systems and its neural network implementation [j]. *Information and Control*, 2:001.

Silva, C. M., Aquino, A. L. L., and Meira Jr., W. (2014). Design of roadside infrastructure for information dissemination in vehicular networks. In *NOMS (Network Operations and Management Symposium)*.