3

Self-Organizing Traffic Lights: A Realistic Simulation

Seung-Bae Cools, Carlos Gershenson, and Bart D’Hooghe

3.1 Introduction: Catch the Green Wave? Better Make Your Own!

Everybody in populated areas suffers from traffic congestion problems. To deal with them, various methods have been developed to mediate between road users as well as possible. Traffic lights are not the only pieces in this puzzle, but they are an important one. As such, different approaches have been used in attempts to reduce user waiting times and prevent traffic jams. The most common involves finding the appropriate phases and periods of traffic lights to quantitatively optimize traffic flow. This results in “green waves” that flow through the main avenues of a city, ideally enabling cars to drive through them without facing a red light, as the speed of the green wave matches the desired cruising speed for the avenue. However, this approach does not consider the current state of the traffic. If there is high traffic density, cars entering a green wave will be stopped by cars ahead of them or cars that turned into the avenue, and once a car misses the green wave, it has to wait the whole duration of the red light to get into the next green wave. On the other hand, for very low densities, cars might arrive at the next intersection too quickly, and then to stop at each crossing. This method is certainly better than having no synchronization at all; however, it can be greatly improved.

Traffic modeling has greatly enhanced our understanding of this complex phenomenon, especially during the last decade (Prigogine and Herman 1971; Wolf et al. 1996; Schreckenberg and Wolf 1998; Helbing 1997; Helbing and Huberman 1998; Helbing et al. 2000), suggesting various improvements in traffic infrastructure. One of these consists of adapting the traffic lights to the current traffic conditions. Indeed, modern “intelligent” advanced traffic management systems (ATMS) use learning methods to adapt phases of traffic lights, normally employing a central computer (Federal Highway Administration 1998; Hunt et al. 1981). The self-organizing approach we present here does not need a central computer, as the global synchronization is adaptively achieved by local interactions between cars and traffic lights, generating flexible green waves on demand.

We have previously shown in an abstract simulation (Gershenson 2005) that self-organizing traffic lights can greatly improve traffic flow for any density. In this chapter, we extend these results to a realistic setting, implementing self-organizing traffic lights in an advanced traffic simulator using real data from a Brussels avenue. In the next section, we give a brief introduction to the concept of self-organization. The SOTL control method is then presented, followed by the moreVTS simulator. In Section 3.5, results from our simulations are shown, followed by discussion, future work, and conclusions.

3.2 Self-Organization

The term *self-organization* has been used in different areas with different meanings, as is cybernetics (von Foerster 1960; Ashby 1962), thermodynamics (Nicolis and Prigogine 1977), biology (Camazine et al. 2003), mathematics (Lendaris 1964), computing (Heylighen and Gershenson 2003), information theory (Shalizi 2001), synergetics (Haken 1981), and others (Skår and Coveney 2003). (For a general overview, see [(Heylighen 2003).] However, the use of the term is subtle, since any dynamical system can be said to be self-organizing or not, depending partly on the observer (Ashby 1962; Gershenson and Heylighen 2003): If we decide to call a "preferred" state or set of states (i.e., attractor) of a system "organized," then the dynamics will lead to a self-organization of the system.

It is not necessary to enter into a philosophical debate on the theoretical aspects of self-organization to work with it, so a practical notion will suffice (Gershenson 2006):

> A system *described* as self-organizing is one in which elements *interact* in order to achieve *dynamically* a global function or behavior.

This function or behavior is not imposed by one single or several elements, nor is it determined hierarchically. It is achieved *autonomously* as the elements interact with one another. These interactions produce feedbacks that regulate the system. If we want the system to solve a problem, it is useful to describe a complex system as self-organizing when the "solution" is not known beforehand and/or is changing constantly. Then, the solution is dynamically sought by the elements of the system. In this way, systems can adapt quickly to unforeseen changes as elements interact locally. In theory, a centralized approach could also solve the problem, but in practice such an approach would require too much time to compute the solution and would not be able to keep pace with the changes in the system and its environment.

In engineering, a self-organizing system would be one in which elements are designed to *dynamically* and *autonomously* solve a problem or perform a function at the system level. Our traffic lights are self-organizing because each one makes a decision based only on local information concerning its own state. Still, they manage to achieve robust and adaptive global coordination.

3.3 Self-Organizing Traffic Lights: The Control Method

In the SOTL method [originally named *SOTL-platoon* in Gershenson (2005)], each traffic light, i.e., intersection, keeps a counter κ_i that is set to zero when the light turns red and then incremented at each time step by the number of cars approaching *only* the red light (i.e., the next one a car will reach) independently of the status or speed of the cars (i.e., moving or stopped). When κ_i (representing the integral of cars over time) reaches a threshold θ, the green light at the same intersection turns yellow, and at the following time step it turns red with $\kappa_i = 0$, while the red light that counted turns green. In this way, if there are more cars approaching or waiting behind a red light, it will turn to green faster than if there are only few cars. This simple mechanism achieves self-organization in the following way: if there are only a few cars, these will be stopped behind red lights for more time, giving other cars time to join them. As more cars join the group, cars will wait less time behind red lights. With a sufficient number of cars, the red lights will turn green even before they reach the intersection, generating "green corridors." Having "platoons" or "convoys" of cars moving together improves traffic flow, compared to a homogeneous distribution of cars, since there are large empty areas between platoons, which can be used by crossing platoons with little interference.

The following constraint prevents traffic lights from switching too fast when there are high densities: a traffic light will not change if the number of time steps is less than a minimum phase, i.e., $\varphi_i < \varphi_{\min}$ (φ_i is the time since the light turned green).

Two further conditions are taken into account to regulate the size of the platoons. Before changing a red light to green, the controller checks if a platoon is crossing through, in order not to break it. More precisely, a red light is not changed to green if on the crossing street there is at least one car approaching within a distance ω from the intersection. This keeps crossing platoons together. For high densities, this condition alone would cause havoc, since large platoons would block the traffic flow of intersecting streets. To avoid this, we introduce a second condition: condition one is not taken into account if there are more than μ cars approaching the green light. Thus, long platoons can be broken, and the restriction only comes into play if a platoon will soon be through an intersection.

The SOTL method is formally summarized in Algorithm 3.1.

This method has no phase or internal clock. If there are no cars approaching a red light, the complementary light can stay green. We say that this method is self-organizing because the global performance is given by the local rules followed

Algorithm 3.1: Self-organizing traffic lights (SOTL) controller.

```
foreach ($time\ step$) do
    $\kappa_i$ += $cars_{\text{approachingRed}}$ in $\rho$
    if ($\varphi_i \geq \varphi_{\min}$) then
        if not ($0 < cars_{\text{approachingGreen}}$ in $\omega < \mu$) then
            if ($\kappa_i \geq \theta$) then
                $\text{switchlight}_{\text{i}}()$
                $\kappa_i = 0$
            end
        end
    end
end
```

by each traffic light: they are “unaware” of the state of other intersections and still manage to achieve global coordination.

The method employs a similar idea to the one used by Porche and Lafortune (1998), but with a much simpler implementation. There is no costly prediction of arrivals at intersections, no need to establish communication between traffic lights to achieve coordination, and not fixed cycles.

3.4 A Realistic Traffic Simulator: moreVTS

Our simulator (moreVTS 2006), (a more realistic Vehicle Traffic Simulator) is the third in a series of open source projects building on the previous one, developed in Java. Green Light District (GLD 2001) was developed by the Intelligent Systems Group at the University of Utrecht (Wiering et al. 2004). It was then improved upon by students in Argentina within the iAtracos project, which we used as a starting point for our simulator, which introduces realistic physics into the simulation. Among other things, acceleration was introduced and the scale was modified so that one pixel represents 1 m and one cycle represents 1 s.

The simulator allows the modeling of complex traffic configurations, enabling the user to create maps and then run simulations varying the densities and types of road users. Multiple-lane streets and intersections can be arranged, as well as spawn and destination frequencies of cars. For implementation details of moreVTS, the reader is referred to Cools (2006).

The self-organizing traffic light controller described in the previous section was implemented in moreVTS. Using data provided by the Brussels Capital Region, we were able to build a detailed simulation of the Rue de la Loi/Wetstraat, a four-lane one-way westward avenue in Brussels that gathers heavy traffic toward the center of the city. We used the measured average traffic densities per hour on working days for 2004 (shown in Table 3.1) and the current “green wave” method, which has a period of 90 s, with 65 s for the green phase on the Wetstraat, 19 for the green phase on side streets, and 6 for transitions. This enabled us to compare our self-organizing controller with a standard one in a realistic setting. Figure 3.1 shows the simulation view of the Wetstraat and its surrounding streets.

The data from Table 3.1 is for the cars entering the Wetstraat on the east, so the spawn rates for the two nodes in the simulation representing this were set according to these data. For the other nodes, the spawn and destination frequencies were set based on a field study we performed in May 2006, comparing the percentage of cars

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
476	255	145	120	175	598	2933	5270	4141	4028	3543	3353	3118	3829	3828	3334	3318	3519	3581	3734	2387	1690	1419	1083

Table 3.1. Average vehicle count per hour at the beginning of the Wetstraat. Data kindly provided by the Brussels Capital Region.

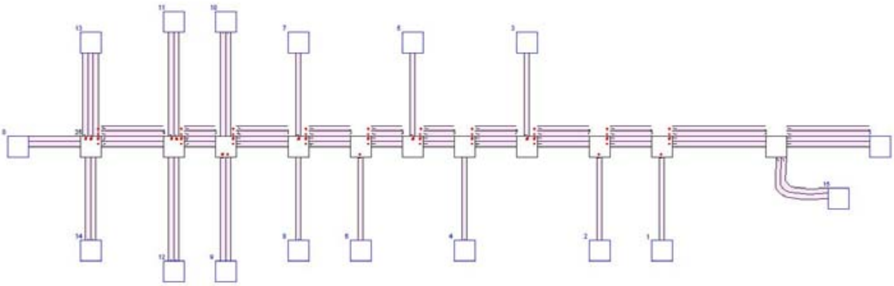

Fig. 3.1. Simulation of the Wetstraat and intersecting streets. Cars flow westward on the Wetstraat. Dots represent traffic lights for each incoming lane at intersections.

that flow through the Wetstraat and those that flow through side streets, entering or leaving the Wetstraat. These percentages were kept constant, so that when the density of cars entering the Wetstraat changed, all the other spawn rates changed in the same proportion. On average, for every five cars flowing through a side street, 100 flow through the Wetstraat. This is not the case of the Kuststraat, a two-way avenue at the west of the Wetstraat (second and third crossing streets from left to right in Fig. 3.1), where for 100 cars driving through the Wetstraat, about 40 turn right, 40 turn left, and only 20 go straight, while 20 more drive through the Kuststraat (about 10 in each direction). The precise spawn rates and destination frequencies are given in Cools (2006, pp. 55–57).

3.5 Results

To measure the performance of the current green wave method and our self-organizing controller, we used the average trip waiting times (ATWT). The trip waiting time for one car is the travel time minus the minimum possible travel time (i.e., travel distance divided by the maximum allowed speed, which for the Wetstraat simulation is about 60 s).

Several simulation runs were performed to find the best parameters for the SOTL method. For each parameter and traffic density, five simulation runs representing 1 h, i.e., 3600 cycles, were averaged. The results were robust and consistent, with SOTL performing better than the green wave method for a wide range of parameters θ and $\varphi_{\min}$ (Cools 2006). Only the best results are shown in Fig. 3.2, together with the results for the green wave method. The cruise speed used was 14 m/s, $\omega = 25$ and $\mu = 3$. Since some densities from Table 3.1 are very similar, we averaged and considered the same densities for 2:00, 3:00, and 4:00; 8:00 and 9:00; 10:00, 17:00, and 18:00; 11:00, 15:00, and 16:00; 13:00, 14:00 and 19:00; and 21:00 and 22:00.

As Fig. 3.2 shows, there is considerable reduction in ATWT using SOTL instead of the current green wave method. The ATWT for the densities at different hours using SOTL were from 34 to 64% of the ATWT for the green wave method, and on average

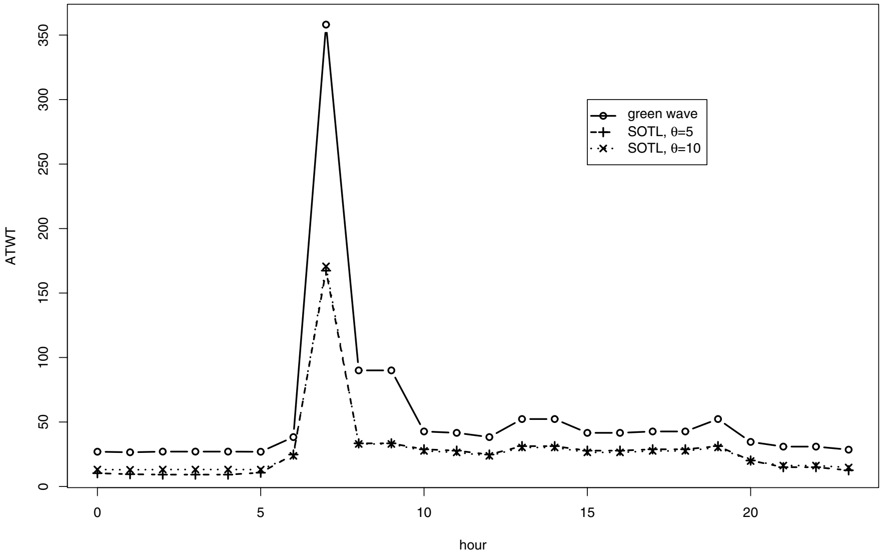

Fig. 3.2. Average trip waiting times (ATWT) at different hours of the day with green wave and SOTL controllers with $\varphi_{\min} = 5$ and $\theta = 5$ and 10.

50%. Since the minimum travel time for the Wetstraat is about 1 min, whereas the overall ATWT for the green wave method is also about 1 min and for SOTL about half of that, the improvement in the average total travel times would be of about 25%, i.e., cars under a green wave method would take 33% more time to reach their destination than those under SOTL. This shows with a realistic simulation that SOTL greatly improves traffic flow compared to the current green wave method.

3.6 Discussion

The green wave method works well for regulating traffic on the Wetstraat, since most of the traffic flows through it. Still, having no concern as to the actual state of the traffic has several drawbacks. It can give a green light to a side street even if there are no cars on it or when a group of cars is about to cross in the other direction. Also, if the traffic density is high, the speed of the cars will be slower than that of the green wave. Furthermore, when a car misses a green wave, it has to wait a full cycle to get into the next one.

Having actual information about the traffic state enables SOTL to adapt to the current situation: it only gives green lights on demand, so time is not wasted for streets without cars, whereas streets with more cars, which thus have more demand, have more green lights. Cars do have to wait behind red lights, but since while doing so they are demanding to cross, it is very unlikely that a car will have to wait more than

$\phi_{\min}$. Moreover, when a car is stopped, a platoon is likely to be formed, accelerating the switching of green lights.

Another advantage of platoons is that they reduce entropy in the city, defined via the probability of finding a car in any part of the city. If there is maximal entropy, there is the same probability of finding a car anywhere in the city. This increases the probability of interference, i.e., that two cars will meet at an intersection, thus requiring one to stop. The opposite extreme is less desirable: if we have a certainty of the position of every car, it is because they are stopped, i.e., in a traffic jam. However, platoons offer a useful balance: there is a high probability that a car will be close to another car, i.e., in a group. Thus, there are many free spaces left between platoons, which other platoons can exploit to cross without interference. There will be interferences, but these will be minimal.

3.7 Future Work

The following list summarizes future work.

- A method similar to SOTL has been used successfully in the United Kingdom for some time, but only for isolated intersections (Vincent and Young 1986). Indeed, it is not obvious to expect that traffic lights without direct communication would be able to coordinate robustly. In any case, the technology to implement it is already available, so a pilot study could be quickly deployed in a real city. Since the traffic lights are adaptive, only a few intersections would have to be changed, to adapt to the control method used in the rest of the city. This also would make it easy to incrementally introduce them in large cities.
- We have observed that there is a monotonic relationship between the best θ and the traffic density (Cools 2006). Exploring this relation better could allow us to set a variable θ depending on the current traffic density measured by the traffic lights. However, since SOTL performs very well for a broad range of parameters, it does not require the calculation of precise parameters. In other words, SOTL is not sensitive to small changes in parameters, making it a robust method.
- The SOTL method could also be used to give preference to certain users, e.g., public transport or emergency vehicles. Simply, a weight would be given to each vehicle in the count κ_i, so that vehicles with preference would be able to trigger green lights by themselves. They would be equivalent to a platoon of cars, thus being seamlessly integrated into the system. This might be a considerable improvement compared to current methods, where some vehicles (e.g., buses in London, trams in Brussels) have preference and the rest of the users are neglected, in some cases even when there are no preferred vehicles nearby.
- The "optimal" sizes of platoons, depending on different features of a city, is an interesting topic to research. The parameters of SOTL can be regulated to promote platoons of a certain size, so knowing what size should be aimed at would facilitate the parameter search.

- It would be interesting to compare SOTL with the Dresden method (Helbing et al. 2005; Lämmer et al. 2006), which couples oscillators using self-organization, whereas SOTL has no internal phases or clocks.

3.8 Conclusions

In this chapter we presented results showing that a self-organizing traffic light control method considerably improves the traffic flow compared to the current green wave method, namely reducing average waiting times by half. These results are encouraging enough to continue refining and exploring similar traffic light controllers and to implement them in real cities, starting with pilot studies. However, we would not like to further motivate the use of cars with efficient traffic control, since this would increase traffic densities and pollution even more. Any city aiming at improving its traffic flow should promote in parallel alternative modes of transportation, such as cycling, walking, car pooling, and public transport.

Acknowledgments

We should like to thank the Ministerie van het Brussels Hoofdstedelijk Gewest for their support, providing the data for the Wetstraat.

References

Ashby, W. R. (1962). Principles of the self-organizing system. In Foerster, H. V., and G. W. Zopf, J., editors, *Principles of Self-Organization*, pages 255–278. Pergamon, Oxford.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ.

Cools, S. B. (2006). A realistic simulation for self-organizing traffic lights. Unpublished BSc Thesis, Vrije Universiteit Brussel.

Federal Highway Administration (1998). *Traffic Control Systems Handbook*. U.S. Department of Transportation.

Gershenson, C. (2005). Self-organizing traffic lights. *Complex Systems*, 16(1):29–53.

Gershenson, C. (2006). A general methodology for designing self-organizing systems. Technical Report 2005-05, ECCO.

Gershenson, C., and Heylighen, F. (2003). When can we call a system self-organizing? In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J. T., and Ziegler, J., editors, *Advances in Artificial Life, 7th European Conference, ECAL 2003 LNAI 2801*, pages 606–614. Springer, Berlin.

Haken, H. (1981). Synergetics and the problem of self-organization. In Roth, G. and Schwegler, H., editors, *Self-Organizing Systems: An Interdisciplinary Approach*, pages 9–13. Campus Verlag, New York.

Helbing, D. (1997). *Verkehrsdynamik*. Springer, Berlin.

Helbing, D., Herrmann, H. J., Schreckenberg, M., and Wolf, D. E., editors (2000). *Traffic and Granular Flow '99: Social, Traffic, and Granular Dynamics*, Springer, Berlin.

Helbing, D., and Huberman, B. A. (1998). Coherent moving states in highway traffic. *Nature*, 396:738–740.

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

Heylighen, F. (2003). The science of self-organization and adaptivity. In Kiel, L. D., editor, *The Encyclopedia of Life Support Systems*. EOLSS, Oxford.

Heylighen, F., and Gershenson, C. (2003). The meaning of self-organization in computing. *IEEE Intelligent Systems*, pages 72–75.

Hunt, P. B., Robertson, D. I., Bretherton, R. D., and Winton, R. I. (1981). SCOOT-a traffic responsive method of coordinating signals. Technical report, TRRL.

Lämmer, S., Kori, H., Peters, K., and Helbing, D. (2006). Decentralised control of material or traffic flows in networks using phase-synchronisation. *Physica A*, 363(1):39–47.

Lendaris, G. G. (1964). On the definition of self-organizing systems. *Proceedings of the IEEE* 52(3):324–325.

Nicolis, G., and Prigogine, I. (1977). *Self-Organization in Non-Equilibrium Systems: From Dissipative Structures to Order Through Fluctuations*. Wiley, New York.

Porche, I., and Lafortune, S. (1998). Adaptive look-ahead optimization of traffic signals. *ITS Journal*, 4(3):209–254.

Prigogine, I., and Herman, R. (1971). *Kinetic Theory of Vehicular Traffic*. Elsevier, New York.

Schreckenberg, M., and Wolf, D. E., editors (1998). *Traffic and Granular Flow '97*, Springer, Singapore.

Shalizi, C. R. (2001). *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. PhD thesis, University of Wisconsin, Madison.

Skår, J. and Coveney, P. V., editors (2003). Self-Organization: The Quest for the Origin and Evolution of Structure. *Phil. Trans. R. Soc. Lond.* A 361(1807). *Proceedings of the 2002 Nobel Symposium on Self-Organization*.

Vincent, R. A., and Young, C. P. (1986). Self optimising traffic signal control using microprocessors - the TRRL MOVA strategy for isolated intersections. *Traffic Engineering and Control*, 27(7-8):385–387.

von Foerster, H. (1960). On self-organizing systems and their environments. In Yovitts, M. C., and Cameron, S., editors, *Self-Organizing Systems*, pages 31–50. Pergamon, New York .

Wiering, M., Vreeken, J., Veenen, J. V., and Koopman, A. (2004). Simulation and optimization of traffic in a city. In *IEEE Intelligent Vehicles Symposium (IV'04)*. IEEE pages 453–458.

Wolf, D. E., Schreckenberg, M., and Bachem, A., editors (1996). *Traffic and Granular Flow '95*, World Scientific, Singapore.