# Coordinated Intelligent Traffic Lights using Uppaal Stratego

Thamilselvam B, Subrahmanyam Kalyanasundaram, and M V Panduranga Rao
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
India 502285.
cs17resch11005@iith.ac.in, subruk@iith.ac.in, mvp@iith.ac.in

***Abstract*—Automatic decision making in traffic signal controllers, semi-automated assistance to drivers, accident detection and response, anti-collision measures in autonomous driving etc., are relatively new applications in Intelligent Transport Systems (ITS). Recent developments in radar and sensor technology coupled with algorithmic and software advances brings ITS closer to realization. In this paper, we extend the work of *Ericksen et. al., "Uppaal Stratego for Intelligent Traffic Lights", in Proc. of the 12th Int. Conf. on ITS European Congress, 2017, France* where they use a tool called UPPAAL STRATEGO to synthesize traffic light timing strategies through statistical model checking and machine learning.**

**While Ericksen et.al. consider a single traffic light controller at an isolated intersection, we consider coordination between the controllers at two traffic intersections by providing a "green wave" in the heavily congested direction which reduces the overall waiting time of cars and queue length. Our experimental results show a significant improvement over uncoordinated isolated traffic light controllers in terms of the waiting time of cars and providing a new functionality of the controller such as giving a green wave.**

## I. Introduction

As the usage of vehicles keeps growing, intelligent traffic light controllers are expected to play an important role in reducing traffic congestion in urban settings. Intelligent Transportation System (ITS) combines Information and Communications Technology and Transportation Systems for ensuring smooth traffic flow through little or no manual involvement. A poor traffic light control system increases the waiting time of road users due to its inefficiency in handling signal phases. Traffic light management systems that use sophisticated sensors, radar and cameras, along with optimization techniques, have been suggested in the past for handling the highly congested traffic flow without a significant increase in infrastructural requirements.

In a novel and promising approach, Ericksen et. al. [1] used UPPAAL STRATEGO, a tool that uses model checking and machine learning to synthesize traffic signal timing strategies. They reported significant improvements in throughput as against static time controller and loop controller in the context of a single traffic junction.

We explore extension of their work multiple junctions. In this paper, we report significant improvements in two coordinated junctions with this approach. While this demonstrates the effectiveness of the approach for multiple junctions as a proof-of-concept, we intend to extend it and evaluate it for general topologies in the future.

This paper is structured as follows. We discuss several existing approaches of coordinated traffic controllers in the next section. In Section III, we briefly discuss UPPAAL STRATEGO, how it can be used in a cooperative traffic light control algorithm and demonstrate the approach with different traffic scenarios. Section IV reports the experimental setup and results. We conclude the paper in Section V and discusses some future directions.

## II. Related Work and Methodology

It is a complex problem to optimize control strategies for traffic signals with varying traffic conditions and dynamically fluctuating demands. Several dynamic control strategies have been proposed for this. In offline optimization, all the relevant data is required to be available before to train the model. In contrast, online optimization does not require complete data prior to train the model. One of the easiest approaches is online plan selection which considers predetermined timing patterns based on current traffic flows, but not considering stochastic behaviour and future traffic flow while generating plans. To fill the gap of offline and online strategies, several types of algorithms have been proposed including those based on fuzzy logic [4], evolutionary algorithms [7], and reinforcement learning (RL) [6].

Huang and Li [7] discuss the advantage of cooperative traffic light control using genetic algorithms show that this approach yields a better performance than the traditional adaptive traffic light control algorithm. Similarly, Zhu et. al., [8] design a semi-real time algorithm towards an intelligent cooperative traffic light control system. Information like waiting times from incoming lanes is used by the controller to adjust the signal dynamically, based on prediction of the future traffic flow using coordination among the intersections.

Liu et. al., [3] present the usage of a recent algorithm called distributed multi-agent Q learning in order to reduce the traffic congestion in an urban city by collecting the neighbours' information at the intersections and coordinating among them. Katwijk et. al., [9] illustrate the advantage of making an isolated intelligent controller and sharing their actions among other controllers. This enables not only the progression to the next traffic light but also the adaptation to different traffic scenarios. They describe a procedure for coordination among the traffic light controllers called multi-agent coordination by which an isolated intelligent agent can be communicated and coordinated in order to achieve global target.

Ericksen et. al. [1] used model based Stochastic Timed-Game Automata to control the traffic signals in one intersection of the road network. To this end, they used a newly developed tool from the Uppaal family of software tools called UPPAAL STRATEGO.

In this paper, we extend and examine the concept of an isolated traffic signal light controller to coordinated traffic light controllers. We simulate results of this controller and observe the performance improvement of the coordinated traffic controllers to the uncoordinated isolated traffic light controller.

## III. Modeling and Implementation

### A. Tools for Modeling

*1) Uppaal Stratego:* Model checking is a technique to algorithmically decide whether or not a formally defined system satisfies a formally specified property. The initial UPPAAL family of tools (UPPAAL, UPPAAL SMC) concerned with different approaches for this purpose, e.g. symbolic and statistical model checking. However, subsequent advances in theory led to the development of UPPAAL TIGA, that allowed *synthesis* of strategies or controllers that ensured satisfaction of desired properties. Finally, UPPAAL STRATEGO [13] facilitates further analysis of these strategies for optimality against some performance metrics, by searching the strategy space. Of particular note is the tool's ability to learn from multiple (non-deterministic) strategies, an optimal deterministic one. Indeed, this tool has been successfully used for several case studies. For example, it has been used to learn a controller for adaptive cruise control [10]. There were two cars, one being controlled by the environment and the other car controlled by the strategies which are learned by UPPAAL STRATEGO so as to avoid collision. In another case study UPPAAL STRATEGO was used to learn strategies for controlling the floor heating in a real house [12]. In our work, we use UPPAAL STRATEGO to generate traffic controllers by finding the optimal phase duration in a system of traffic junctions.

*2) SUMO: Simulation of Urban MObility:* Simulation of Urban MObility is an open source, microscopic traffic simulator [2]. It provides external interfacing tools, such as Traci, through which other external programs can communicate with objects of SUMO. It supports generation of different traffic scenarios by generating vehicles from an external program, routes of the cars which run on simulation for visualization of traffic flow, relevant road maps, traffic signal lights which controls and directs the flow of cars in intersections. Waiting time, queue length of waiting cars and jammed cars in the lanes are supportable in the simulator to get a better analysis of different controllers.

The current version of SUMO provides vehicle to vehicle communication to ensure the coordination between vehicles. In our case we use some basic functions of SUMO using Python packages that facilitate communication between SUMO and other external programs. It supports modeling the relevant road network from the map, roads lanes and intersections, different vehicle types for modeling traffic demands, traffic lights for modeling a signalized intersection, induction loops which indicate if a car is on the given detector, area detectors which indicate the number of cars moving or jammed in an area. One can use induction loop and area detector information to improve the controller of a given traffic light. Traci is a software interface that gives access to objects in the running simulation and manipulates their behaviour to implement on-line controllers.

### B. Types of Traffic Light Controllers

The main objective of a traffic light controller is to manage the flow of traffic in intersections by making use of existing infrastructure effectively. Figure 1 shows the two intersections of traffic signal road model. The intersection of a traffic signal has two phases named green and yellow. Further green phase is called GreenA (GreenB) providing green signal in A11-A12 (B11-B12) and A21-A22 (B21-B22) at intersection-1 and intersection-2 respectively. YellowA and YellowB are

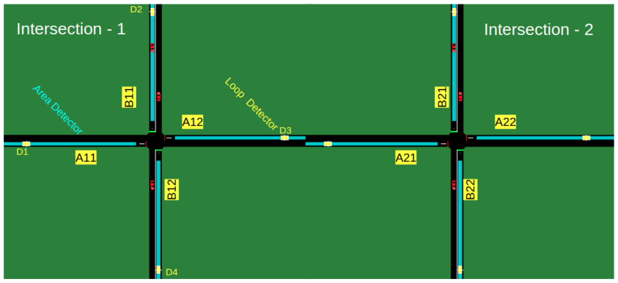


Figure 1. Two Intersection Road Traffic Model

| Controller | Traffic Load | Direction | | Yellow | Cycle Length |
|---|---|---|---|---|---|
| | | A | B | | |
| Static | MAX | 52 | 36 | $2 \times 8$ | 104 |
| | MID | 31 | 17 | $2 \times 8$ | 64 |
| | LOW | 24 | 12 | $2 \times 8$ | 52 |
| Loop | MAX | max. 64 | max. 40 | $2 \times 8$ | 104 |
| | MID | max. 54 | max. 26 | $2 \times 8$ | 64 |
| | LOW | max. 36 | max. 20 | $2 \times 8$ | 52 |

Table I
GREEN TIMES FOR THE STATIC AND THE LOOP CONTROLLERS.

also defined similarly. The phase GreenA is encoded as RGRG signalling red in B11-B12(B21-B22) direction, green in A11-A12(A21-A22) direction at intersection-1(intersection-2). The phase GreenB is encoded as GRGR signalling green in B11-B12(B21-B22) direction, red in A11-A12(A21-A22) direction at intersection-1(intersection-2). Phases YellowA and YellowB are encoded in the same way mentioned above, for example YellowA is coded as RYRY. Whenever the controller indicates green or yellow signal in one direction, the perpendicular direction is always red. While switching between the phases GreenA and GreenB, the signal remains in the corresponding yellow phase for 8 seconds. We consider three controllers in the intersection, namely static time controller, induction loop based controller, and a controller using UPPAAL STRATEGO. We consider three types of scenarios, called MAX, MID and LOW, based on the number of cars arriving to the intersection. These scenarios are generated by Poisson distribution defining the maximum, medium and low traffic load respectively.

*1) Static Time Controller:* In static time controller, a fixed-time signal control policy is used for traffic control due to its simplicity in implementation. This controller has different fixed cycle length and different fixed phase duration between two phases for varies traffic scenarios for any given direction based on optimized computations in offline. The phase duration for the different traffic load scenarios are given in Table I. In figure 1, A11(A21), A12(A22) lanes are called 'A' direction and B11(B21), B12(B22) lanes are called 'B' direction at intersection-1(intersection-2).

All load scenarios of static controller are encoded in an XML file. For every scenario, there is one XML file for this controller. Coordination between static controllers is implemented by adjusting its green and yellow phase time based on the distance between two traffic lights and speed of the vehicles.

*2) Loop Controller:* We have implemented the Loop controller in SUMO using Traci and Python. Loop detectors (D1, D2, D3, D4) are placed 320 meters away from the intersection on all lanes as shown in figure 1 which are used to count the crossed cars before they reach the intersection from all directions. Area detectors are placed on lanes A11(A21), A12(A22), B11(B21) and B12(B22) at intersection-1(intersection-2) for waiting time calculation. A green phase has 8 seconds of minimal duration. The maximum duration for the different traffic load scenarios are given in Table I. In Loop Controller, if there is no indication from B side, then controller directs green signal in A direction forever. Whenever the car crosses the loop detector in either direction, then the relevant green timer is extended until MAX time is reached as in table I. If there are any indications from B side, then we extend the green timer in A side until MAX time is reached as in table I.
Note: Extension of time should be the time period such that either a car crosses the intersection or reaches the crossing line.

*3) Uppaal Stratego Controller:* This controller integrates SUMO and UPPAAL STRATEGO using Traci. The controller will read the status of the traffic lights and data from the areal detectors in SUMO, in every 5 to 8 seconds. The sensed data is updated in the Uppaal model to get the next phase information. Optimal strategies for traffic light are generated by simulating the model for predicting the near future traffic flow (120 seconds) using current sensor values. The predicted strategies are used for next 5 seconds to control the traffic flow in SUMO simulation. Then the next cycle of execution starts with the new sensed data. While updating the model the shared data such as traffic light status of other controller is also updated along with sensed data. This shared information is used by the controller to decide the next phase. The coordination is done based on fuzzy contextual rules. If the load of one controller crosses some threshold values in particular direction, the other controller extends the green time in relevant direction. If load of the controller is in normal, then the controller acts as an isolated optimized one.

The flow function in the model handles the shared variables such that to increase the density of flow to give green wave in highly congested direction. For example, in Yellow Phase model, we start with initializing the global variables required. Once the phase is chosen randomly in location ChoosePhase it may either go to location GreenA or GreenB. The Yellow signal is indicated when green signal was over in relevant direction by YellowA or YellowB location. The flow function defines the flow of traffic, in this case number of cars crossed, arrived and curStep value are updated based on delchoice(delay choice) and simulation comes to the ChoosePhase location. If the curStep is less than Horizon value, then the simulation continues. Once it reaches the 'Done' location one run is over and waiting time is calculated internally. For the second run it starts with initial global variables and simulates. After many simulation the waiting time is converged and optimized, then the uppaal stratego simulation stops and gives its output to SUMO simulator. The high level algorithm is shown in Algorithm 1:

---

**Algorithm 1 High level algorithm for the Coordinated Uppaal Stratego controller**

---

**Input:** Read areal data from SUMO
**Output:** Next signal phase from Uppaal Stratego
  **if** Status of Traffic Light is in green phase **then**
    Run Uppaal Stratego Figure 2 – to learn whether extend green phase or go to yellow
    Share the Status and data to neighbour controller
  **else if** Traffic Light in yellow phase **then**
    Run Uppaal Stratego Figure 3 – to learn which direction should have the next green phase
    Share the Status and data to neighbour controller
  **end if**

---

## IV. Experiment and Simulation Evaluation

We use SUMO micro traffic simulator for evaluating and simulating the road traffic network. A combination of UPPAAL STRATEGO models of Green phase and Yellow phase acts as a controller and integration of UPPAAL STRATEGO and SUMO is done through Traci in Python. The area detectors placed on lanes give details of arriving and jamming cars in Figure 1.

### A. Experimental Setup

We evaluate our algorithm for waiting times and queue lengths. These are defined as follows:

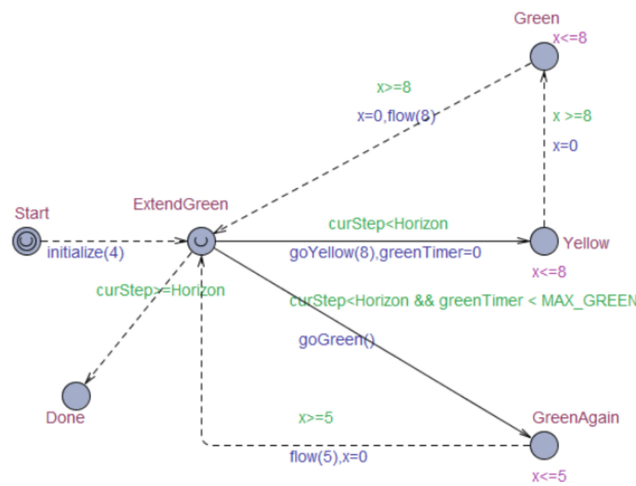


Figure 2. ExtendGreen Phase

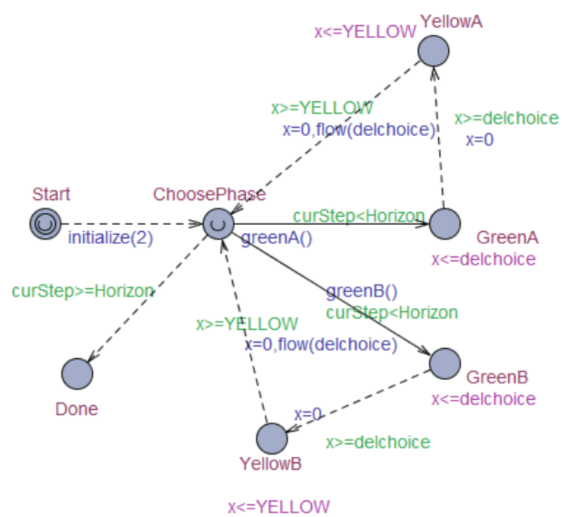


Figure 3. Yellow Phase

| LOW | Coordinated | | | | Uncoordinated | | | |
|---|---|---|---|---|---|---|---|---|
| | Waiting Time (s) | | Cumulative Queue Length (m) | | Waiting Time (s) | | Cumulative Queue Length (m) | |
| | T1 | T2 | T1 | T2 | T1 | T2 | T1 | T2 |
| **Static** | 17597 | 2153 | 56654 | 8403 | 12849 | 2852 | **39605** | 9796 |
| **Loop** | 22736 | 1596 | 77858 | 5967 | 22736 | 1596 | 77858 | 5967 |
| **Stratego** | **14810** | **138** | **48214** | **620** | **11916** | **225** | 43979 | **937** |

Table II
Low Traffic Scenarios

Waiting Time: The number of cars waiting for the signal in unit time. For example, if 20 cars are waiting for 10 seconds(simulation step), then the overall waiting time is 200 seconds.

| **MID** | **Coordinated** | | | | **Uncoordinated** | | | |
|---|---|---|---|---|---|---|---|---|
| | Waiting Time (s) | | Cumulative Queue Length (m) | | Waiting Time (s) | | Cumulative Queue Length (m) | |
| | T1 | T2 | T1 | T2 | T1 | T2 | T1 | T2 |
| **Static** | 68031 | 61151 | 223075 | 200526 | 31042 | 58861 | 100887 | 191803 |
| **Loop** | **15982** | 104028 | 55025 | 324793 | **15982** | 104028 | **55025** | 324793 |
| **Stratego** | 36455 | **43130** | **99413** | **84726** | 33229 | **39716** | 107545 | **86090** |

Table III
MEDIUM TRAFFIC SCENARIOS

| **MAX** | **Coordinated** | | | | **Uncoordinated** | | | |
|---|---|---|---|---|---|---|---|---|
| | Waiting Time (s) | | Cumulative Queue Length (m) | | Waiting Time (s) | | Cumulative Queue Length (m) | |
| | T1 | T2 | T1 | T2 | T1 | T2 | T1 | T2 |
| **Static** | 341505 | 96979 | 1052809 | 315013 | 358483 | 52716 | 990741 | 166502 |
| **Loop** | 366799 | 174418 | 1160835 | 544729 | 366799 | 174418 | 1160835 | 544729 |
| **Stratego** | **168385** | **11939** | **488441** | **58166** | **163184** | **10768** | **485299** | **53334** |

Table IV
MAXIMUM TRAFFIC SCENARIOS

Cumulative Queue Length: Cars waiting in a queue on the lanes for signal over the period of simulation, measured as length in meters. For example, 100 cars wait over a period of 100 simulation steps, each of them being 3 meters long, then the cumulative queue length would be approximately 30000 meters (we ignore the inter-car distance).

### B. Scenarios for simulation

The distance between two controllers is 700 meters. All vehicles' parameters are the same as in the paper [1]. We consider a simple road network. We take the density of traffic load in A direction higher than that in the B direction. We consider three different road traffic load scenarios namely MAX, MID and LOW. MAX is defined as maximum arrival rate of cars in a particular intersection from all directions based on Poisson distribution. MID and LOW are 70% and 35% of MAX rate of traffic flow respectively.

### C. Evaluation

The relevant data such as waiting time, queue length and traffic light status are extracted from SUMO simulation to check the analysis, in every 20 seconds. For all scenarios, the simulation time duration is 300 seconds. The number of cars are arriving to junction varies from LOW to MAX. From the table in medium traffic scenarios, the coordinated controller T2, has the value of 84726 meters in cumulative queue length. From the Tables II, III and IV it can be seen that coordinated stratego controller performs better than uncoordinated stratego controller by achieving green waves in the moderate congested direction, these results are from SUMO simulation. In low traffic scenarios, stratego and static controller work a little bit better than all other controllers.

## V. CONCLUSION AND FUTURE WORK

A cooperative intelligent traffic light controller has many advantages over an isolated intelligent traffic light controller because of its responsiveness to traffic demands and providing progression along arterials, making the decision not only from local information but also from the shared global information. In this paper, we studied the performance of two coordinated controllers sharing their status of current traffic lights and areal data. We used simulations to show that a coordinated intelligent controller performs better than isolated intelligent controller in terms of waiting time and queue length in moderate and low traffic scenarios.

From this, we can conclude that coordinated traffic lights has many advantages. Moreover, scaling up to several controllers is also possible since UPPAAL STRATEGO supports a combination of both symbolic model checking and statistical model checking. The symbolic model checking is used for strict constraint scenarios and the simulation-based model checking for liberal constraint scenarios.

In the future, we plan to extend this work by coordinating among several intersections in a complex road network and improve the coordination technique to handle maximum traffic load scenarios. We also plan to analyze heterogeneous traffic scenarios including pedestrian crossing. In the future, one could implement the controllers in a real time test bed, and use the real measurements instead of simulations to study this system.

## ACKNOWLEDGMENT

We would like to thank Marco Muñiz from Aalborg University for his help with the tools. This work was supported by the project "M2Smart: Smart Cities for Emerging Countries based on Sensing, Network and Big Data Analysis of Multimodal Regional Transport System", JST/JICA SATREPS, Japan.

## REFERENCES

[1] Eriksen, A. B., Huang, C., Kildebogaard, J., Lahrmann, H. S., Larsen, K. G., Muniz, M., Taankvist, J. H., *"Uppaal Stratego for Intelligent Traffic Lights"*,in Proc. of the 2017 12th Int. Conf. on ITS European Congress,19-22 July, 2017, Strasbourg, France.

[2] Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L, *Recent development and applications of*

*SUMO - Simulation of Urban MObility*, Vol 5, Nos 3 & 4, International Journal on Advances in Systems and Measurements, December 2012.

[3] Ying Liu, Lei Liu and Wei-Peng Chen *“Intelligent traffic light control using distributed multi-agent Q learning,” in Proc. of the 2017 20th Int. Conf. on Intelligent Transportation Systems, ITSC 2017*,16-19 October 2017, Yokohama, Japan

[4] Javed Alam, M.K Pandey and Husain Ahmed *“Intelligent Traffic Light Control System for Isolated Intersection Using Fuzzy Logic,” in Proc. of the 2013 Int. Conf. on Advances in Communication and Control Systems, CAC2S 2013*,16-19 October 2017, Dehradun, India.

[5] Robbin Blokpoel and Wolfgang Niebel, *“Advantage of cooperative traffic light control algorithms”* IET Intelligent Transport Systems, vol. 11, no.2, pp. 379-386, 2017.

[6] Seyed Sajad Mousavi, Michael Schukat and Enda Howley, *“Traffic light control using deep policy-gradient and value-function-based reinforcement learning”* IET Intelligent Transport Systems, vol. 11, no.2, pp. 417-423, 2017.

[7] Zhen-Jin Huang and Chun-Gui Li*“Traffic signal control based on genetic neural network algorithm”, in Proc. of the 2010 Int. Conf. on Intelligent Computing and Integrated Systems*, 22-24 October 2010, Guilin, China.

[8] Qin Zhu, Chao Peng, Jingmin Shi, Pengfei Duan, Yu Bao, and Mengjun Xie, *“Cooperative Traffic Light Control Based on Semi-Real-Time Processing”* Journal of Automation and Control Engineering, vol. 3, no.6, pp. 463-469, 2015.

[9] R.T. van Katwijk, B. De Schutter, and J. Hellendoorn, *“Multi-agent coordination of traffic control instruments,” in Proc. of the 2008 1st Int. Conf. on International Conference on Infrastructure Systems : Building Networks for a Brighter Future 2008*, 10-12 November 2008, Rotterdam, Netherlands.

[10] Larsen K. G., Mikučionis M. and Taankvist J. H., Safe and Optimal Adaptive Cruise Control. In: Meyer R., Platzer A., Wehrheim H. (eds) Correct System Design. Lecture Notes in Computer Science, vol 9360. Springer, Cham, 2015.

[11] Behrmann, G., David, A. A. and Larsen K., *”A tutorial on Uppaal”*, SFM-RT Springer, no.3185, pp. 200-236, 2004.

[12] Kim G. Larsen, Marius Mikučionis, Marco Muñiz, Jiří Srba and Jakob Haahr Taankvist, *“Online and Compositional Learning of Controllers with Application to Floor Heating”*, pp. 244–259. Springer Berlin Heidelberg, Berlin, Heidelberg.

[13] Alexandre David, Peter Gjøl Jensen, Kim Guldstrand Larsen, Marius Mikucionis, and Jakob Haahr Taankvist, *“Uppaal Stratego”*, In Proc. TACAS 2015, LNCS 9035, pp. 206–211, 2015.

[14] Franck Cassez, Alexandre David, Emmanuel Fleury, Kim G. Larsen, and Didier Lime. *“Efficient On-the-fly Algorithms for the Analysis of Timed Games.”*, LNCS 3653. Springer-Verlag 2005, pp 66-80.