# Traffic Control Through Traffic Lights Management: A Comparison Study

IDs: 1 and 13

Dec 22, 2011

## 1 Introduction

Traffic jams are a severe problem affecting cities around the world, and they are only getting worse as the population and number of vehicles continues to increase significantly while the area available for building roads does not. Annually, many citizens waste countless hours stuck in traffic, and the amount of time spent is expected to increase. One study calculated that in 75 different city regions in 1982 the average number of hours wasted per year per traveler during peak hours was 16. By 2000, this had increased to 62 hours [1]. However, in spite of this, many roads are rarely congested. As such, one approach to dealing with this is solving the Urban Traffic Control Signal problem. Simply put, the traffic lights of the city could be used to redirect traffic such that vehicles are moved away from heavily congested roads and onto less congested ones, thereby reducing the amount of time wasted stuck in traffic.

The problem then is figuring out how to generate appropriate signal timing patterns, as most simple patterns are not sufficient to optimally redirect traffic and reduce congestion. Thus many researchers have actively employed different approaches to deal with the problem. Out of these approaches, there have been three main categories: Network Flow approaches, Reinforcement Learning approaches, and Game Theory approaches.

## 2 Network Flow Approaches

So far, there are two different concepts for modeling vehicular traffic. In the "coarse-grained" fluiddynamical description, traffic is viewed as a compressible fluid formed by vehicles that do not appear explicitly in the theory. In contrast, in the "microscopic" models, traffic is treated as a system of interacting particles where attention is explicitly focused on individual vehicles and the interactions among them. These models are therefore much better suited for the investigation of urban traffic. Most of the "microscopic" models developed in recent years are usually formulated using the language of cellular automata CA [2]. But, "macroscopic" models is useful when we want to shape the traffic

on different streets without going into the details of microscopic behaviour of cars which may not be that important when we care more about increasing the social welfare.

The main functions of flow control in a network (originally from data communication networks) are:

1. prevention of aggregated utility degradation and loss of efficiency due to overload,
2. deadlock avoidance,
3. fair allocation of resources among competing users,
4. speed matching between different network segments and the cars.

Aggregated utility degradation and deadlocks occur because the traffic that has already been accepted into the city network (i.e., traffic that has already occupied road segments) exceeds the nominal capacity of the roads. To prevent over allocation of resources, the flow control procedure includes a set of constraints (on traffic lights cycles, on green and red phases, on the number of cars waiting behind red light, etc.) which can effectively limit the access of traffic into the roads or, more precisely, to selected sections of the roads. These constraints may be fixed, or may be dynamically adjusted based on traffic conditions [3].

Note that maximizing the social welfare is not always equivalent to fairness among users. For example, consider Lions Gate in Vancouver. Since there is a huge traffic coming to Downtown from north shore in the morning and going back in the afternoon, it is most efficient to open all lanes for that traffic. But, by doing so the chance of taking the shortest path to West Vancouver is taken from drivers going there in the morning.

Part of the network flow control from communication networks which is applicable to this problem is consisted of hop level and entry-to-exit level flow control. The hop level flow control attempts to maintain a smooth flow of traffic between neighbour intersections, while the latter one tries to deal with the source and destination intersections for each path.

# 3 Reinforcement Learning Approaches

The basic concept of Reinforcement Learning is that agents are provided a set of states and a set of actions, and the agent starts learning by associating rewards and state transitions with the actions it makes during each state. Ideally, the agent will learn what action is best to perform for each state it is presented with after iterative trial of every state-action pair. There are a large variety of ways to implement this learning ability, since agents are largely free to choose what order of actions to attempt and how to associate the utilities gained with previously selected actions or states.

One of the most common ways is the Q-learning algorithm. In this algorithm, agents associate $Q(s, a)$ values with every state-action pair $s$ and $a$ that indicate

the expected future utility of performing the action $a$ when in state $s$. In the simplest form, the algorithm has a two parameters $\alpha$ and $\gamma$. $\alpha$ determines how quickly agents learn from the latest events, but in general this is set to a small value to prevent the agent from rapidly jumping to conclusions, so agents tend to require a large amount of timesteps to learn the optimal policy. $\gamma$ determines how much emphasis agents place on attaining future utility.

Each time the agent performs an action $a$ in a state $s$, it examines its increase in utility $r$ and the new state it arrives in $s'$. It then assigns $(1-\alpha)Q(s,a) + \alpha(r + \gamma \max_b Q(s',b)$ to $Q(s,a)$ [4]. Intuitively, the agent is updating its beliefs on the value of performing action $a$ in state $s$ by first remembering what its old belief was in the $(1-\alpha)Q(s,a)$ term, and then combining it with the reward $r$ it received and the expected future utility changes from arriving in the new state $s'$.

As this concept has had large successes in various artificial intelligence problems, there has been many attempts to apply this to the traffic signal problem as well. The differences in these attempts are mainly how they define the agents, states, and actions of the problem.

## 3.1 State Definitions

Firstly, in contrast with game theory or network flow approaches which seem well-suited for generating static traffic signal plans for periodic traffic patterns, reinforcement learning approaches seem more compatible with generating dynamic traffic signal plans which react to the environment in real time. This can be seen from how the states can be defined. For instance, assume a certain city has a certain traffic pattern on weekdays and another traffic pattern on weekends. This allows us to simply define 2 states for the agent differentiating between weekdays and weekends, and the set of actions is simply the set of possible traffic signal patterns for the day. However, that means the agent only gets to try out one state-action pair per day, which tends to be impractical without an accurate simulation model as the city would have to suffer a significantly large number of days of suboptimal traffic signal control before the agent learns the optimal actions to take.

Instead, most Reinforcement Learning approaches define the states as the actual real-time state of the traffic network, such as the number of cars on each road, or even dividing the city roads into vehicle-sized cells and describing whether each cell contains a car or not [5, 6, 7]. This allows agents to react to environments in real time and allows for somewhat more efficient learning since agents can attempt new state-action pairs extremely often and simulation models tend to be more accurate. Note that this also allows the set of actions to be significantly reduced in complexity, since actions can now be defined simply without time components. For instance, a single traffic light can have simple actions like "green"' or "red"' as opposed to "'repeat for the day: green for 30 seconds then red for 30 seconds"'.

However, this also runs into issues with exponential increases in the number of possible states as the problem size increases. As a result, many approaches

attempt to define states in manners so as to reduce the total number of states.

For instance, one approach uses neurofuzzy traffic signal controls, which allows them to define states using terms like "'a few"', "long"', or "medium"' instead of using numbers [8]. Thus while other approaches may have $X$ different states for a road that may contain 0 to $X$ vehicles, this approach will only have a small finite number of states for that road. Through simulation experiments, the researchers concluded that Reinforcement Learning could still be applied on these fuzzy traffic signal controls to decrease vehicular delay.

## 3.2 Agent Definitions

Secondly, defining the agents for this problem also presents a number of different alternatives. A somewhat naive approach would be to define the entire network of traffic lights as a single agent. This agent can then examine the state of the network and assign traffic signal patterns accordingly. However, although this seems ideal in that reinforcement learning would eventually learn the optimal actions for each state, this would actually take exponential time in practice due to the number of possible actions [6]. For instance, even in the simple case where every intersection only has a single traffic light with only two signals, that still leaves the agent with $2^N$ possible actions per state where $N$ is the number of intersections.

At the other extreme, many approaches define each traffic light as a single agent. Thus the problem becomes a multiagent system where each traffic light only has a small constant number of actions. However, these agents tend to find local optimal solutions as opposed to global optimal solutions [6, 7]. This is problematic because it may often be the case that a globally optimal solution is made up of suboptimal local solutions. For instance, it may be better for a distant traffic light to stay red and refuse to allow vehicles to proceed on to a congested area. For these approaches, the key to their successes is in coming up with methods to make each traffic light agent cooperate with one another.

Bazzan et. al. [7] used additional supervisor agents to ensure cooperation. In their approach, the traffic light agents are split into small groups, each with one supervisor agent. The traffic light agents continue to find their local optimal state-action pairs, while the supervisor observes the joint effects of actions taken by the group of traffic lights and recommends joint actions to take. Using simulation models, the researchers determined that the supervisor agents did indeed improve traffic conditions, with the best conditions achieved when traffic light agents listen to the supervisor recommendations when the expected utility for the recommended action for the agent is almost as good as the expected utility for the locally optimal action.

Kuyer et. al. [6] take a different approach and use coordination graphs and approximation algorithms to handle the agents. The approximation algorithm they used has been proven to converge to the optimal action in tree-structured graphs, but not cyclical graphs like urban road networks. Nevertheless, it scales well with the problem size and it successfully outperformed non-coordinated

methods in various simulation scenarios, including those with large road networks.

In addition to these agent definitions, there is also the possibility of including drivers as agents in the system. In this approach, the drivers learn what roads to take at the same time as the traffic light agents learn what signal patterns to use in a form of co-learning [5]. Wiering [5] considered the scenario where drivers and traffic light agents share the same utility function and cooperate to improve global traffic conditions. In this scenario, he found that drivers actually helped to improve traffic conditions significantly, such as by avoiding congested roads and spreading out. However, this seems infeasible to implement in practice due to real life drivers having selfish utility functions instead of sharing a global utility function.

### 3.3 Utility Versus Regret

One of the major issues in multi-agent reinforcement learning is that agents' strategies often converge to suboptimal equilibria or worse, fail to converge at all. This is true even for simple games that do not have a pure strategy Nash Equilibrium [9]. In order to fix this problem, a new type of learning called No-Regret Learning was developed.

In No-Regret Learning, the agent does not associate utility values with states and actions, instead it associates regret values. Intuitively, regret is the utility difference between playing the strategies suggested by the No-Regret Learning algorithm instead of playing some other pure strategy. In a paper by Jafari et. al. [9], the researchers found that using No-Regret Learning algorithms allowed agents to learn Nash Equilibria in constant-sum games and $2 \times 2$ general-sum games. However, they also found counterexamples for larger general-sum games [9], suggesting more work has to be done before Reinforcement Learning approaches can truly find the optimal Nash equilibria in complicated problems such as this urban traffic signal problem.

## 4 Game Theory Approaches

Game Theory approaches consider drivers as selfish agents aiming to maximize their own personal utility, which is often a more accurate depiction of the real world. In addition, there is also an agent representing the traffic system with a utility function based on the efficiency of the traffic network, which may be measured in many different ways. The goal of game theory approaches is to find the Nash Equilibrium for which the traffic system agent's utility is maximized. Upon finding this equilibrium, drivers can then be instructed to conform to the strategies in the equilibrium based on the fact that if everyone else conformed to the strategies in equilibrium, each driver's best response to maximize personal gain would be to conform as well. However, finding Nash Equilibriums is computationally difficult and game theory models in general suffer from exponential blow up in complexity as the size of the problem increases. Thus approaches

attempt to construct game theory models which exploit the structure of the traffic signal control problem in order to be more compact.

## 4.1 Congestion Games

Suppose $n$ players are simultaneously competing for a shared resource or a set of resources $R$ (here, roads), that minimizes their cost $C$ or maximize their benefit (e.g. travel speed).This is the classical formulation for Congestion Games. Since the resources are limited and the cost of each resource, say the delay of traversing a road segment, is a function of players who want that resource, there is usually a congestion $n_r$ on resource $r$ among agents with overlapping strategies. Agents are self-interested and try to minimize their choice cost (summation of costs of different road segments on the path of that agent) by choosing the best strategy $S$.

It is shown that agents in a given state $S = (S_1, S_2, ..., S_n)$, by doing a sequence of improvement steps, which is changing the strategy in which case the cost is decreased, not only do not run into cycles but also converge to Nash equilibrium after a finite number of steps [10]. Another important result is shown by Fabrikant et al. [11], which says computing a Nash equilibrium in symmetric congestion games is Polynomial Local Search (PLS)-complete.

## 4.2 Temporal Action Graph Games

In 2009, a new compact model called Temporal Action-Graph Games (TAGG) was introduced [12]. As the name implies, TAGG model builds upon Action-Graph Games (AGG), a compact representation for games with no time component and certain structural features. In particular, AGGs allow for very compact representations when the utility functions depend only on the number of other agents selecting certain actions, as opposed to the identities of these agents [13]. This is certainly the case for the urban traffic signal control problem, since drivers in general don't particularly care about who the other drivers are, instead only caring about the number of cars that are in their way.

The standard game theory model defines a game using a set of agents $N$, a set of actions $A$, and a set of utility functions $U$. AGGs build on this by adding a directed graph $G$ that indicates interactions between different actions and restricting the utility functions to be based solely on the action chosen $a$ and the number of agents who choose actions that have interactions with $a$. In so doing, the space complexity of the game only increases polynomially in the number of agents. This also allowed for various computations to be sped up significantly, including the search for Nash Equilibrium [13].

TAGGs build upon AGGs by simply adding a discrete time component. In a sense, TAGGs are like imperfect-information extensive form games while AGGs are like normal form games. This is useful for the urban traffic signal control problem, which has different cars starting from their origins at different times. In particular, the researchers investigated a small example of an urban traffic problem where individual cars have to choose which lane to drive on [12]. In

this problem, cars enter the starting point at different times, and each car can see which lanes the previous drivers have chosen, but not see the lanes chosen by drivers entering at the same time. Like the urban traffic signal control problem, the utility functions of the drivers depend on the number of cars in the way, or in other words, the number of drivers who choose the same lane before or at the same time. For this small problem, the TAGG model has a compact representation which only increases polynomially in terms of the number of drivers, the number of lanes, and the number of discrete time steps.

However, the TAGG model alone does not seem sufficient to fully model the urban traffic signal problem, due to the stochastic nature of the problem and the fact that drivers' action spaces may change at every time step, which would require complicated, possibly superpolynomial, transition definitions in the representation.

### 4.3 Stochastic Games

The formal setting of Stochastic Games includes a set on $n$ agents, $N$, a set of discrete state action space, $S$, a discrete set of joint actions, $A$, a reward (payoff) function, $R$, and finally the set of transition probabilities over the state space $S$, $T$.

Most of works on stochastic games is based on a single state game with common payoff and therefore the game is no longer dynamic. Claus and Boutilier [14] studied influential parameters on dynamics of a Q-learning process in a coordination game. There are two settings: in the first one, each agent act independently; in the second one, agents have beliefs about other players' strategies. They showed by experiment that independent learners converge quickly but not necessarily to the same equilibrium and the performance of the second setting is not much better.

The special case of zero-sum games in stochastic games is discussed by Littman in [15]. Each agent follows his maxmin strategy in the 2-player case, and tries to maximize his minimum achievable utility. For 2-player general sum game, because the minimax Q-learning cannot be used, both players take actions in state $s$ and then, follow their Nash equilibrium strategies ( [16]). It is worth mentioning that we need perfect information for this to work.

Now the question is whether we should use stochastic games for traffic lights control or not? There are two main issues discussed by Shoham et al. [17], to consider for this scenario. The first is the focus on convergence to equilibrium regarding the stage game: "If the process [of playing a game] does not converge to equilibrium play, should we be disturbed?" The major part of the research so far has concentrated on the actions and strategies in which the game converges, not on the players' utilities in equilibrium. The second issue is that "In a multi-agent setting, one cannot separate learning from teaching" because agent $i$'s action selections both arise from information about agent $j$'s past behavior, as well as impact $j$'s future actions' selections.

Unless $i$ and $j$ are completely unaware of the presence of each other, both can teach and learn how to play in mutual benefit. So, a more neutral term

would be multi-agent adaptation (rather than learning). This is very important since it is consistent with a view that some operational control related issues are more an attempt for adaptation rather than for optimization. As optimization is hard to achieve within a short time frame, it is often the case that this cannot be done in real-time. One more point in favor of adaptation is that many works on Multi-Agent Reinforcement Learning (MARL) have been assuming static environments. In this kind of environment it may make sense to evaluate MARL algorithms by convergence to a stationary policy, and convergence to a best response if the opponent converges to a stationary policy. It certainly makes little sense to evaluate a learning or adaptation algorithm by such criteria when the environment is itself dynamic, as it is the case of the traffic scenario discussed here.

From the above discussion, we can understand that although both cooperative and non-cooperative games can be applied to model this problem, but because of the highly variable environment and the fact that the information is far from complete, as the scale of the problem grows, it is not easy to use game theoretic formulation. In fact, we can use game theoretic modeling when there is a level of abstraction necessary for that modeling. The other main issue that this kind of modeling (and any other modeling) has to deal with is the problem of global versus local optimum. Assume we are to decide whether change the traffic light of an intersection to green or keep it red. What is the best strategy here? Suppose there is a huge backlog at a neighbor intersection which is on the route for some of the cars in our example intersection. Clearly, for those cars it may be better to stay a little longer here to let the other intersection clear its traffic. In other words, it is in their best interest if we do not change the traffic light indication. However, for other cars that their path does not include the mentioned backlogged intersection, it is best to get the green indication as soon as possible.

# 5 Conclusion

In this paper, we investigated the three main approaches to the urban traffic signal control problem. Each approach had their own strengths and weaknesses, so none of them could be said to have solved the problem perfectly.

In general, network flow approaches are more computationally efficient and scalable, but do not consider drivers as individuals and thus the solutions provided may not be as reliable or optimal. In fact network flow solutions have worked well for data networks especially in Internet traffic control, which is similar to urban traffic except that in the city, the system has less control over the actions of the drivers. Therefore, network flow approach is very scalable, and effective in shaping the traffic for different road sections. On the other hand, the main weakness is that internet traffic has obedient "drivers" while the real world often has selfish drivers and thus the solutions provided may not be very precise.

In contrast, Game Theory approaches consider every driver as an individual

agent and attempt to discover the optimal Nash Equilibrium. The discovery of the strategies used in such a Nash Equilibrium can be proven to drivers as optimal for their own personal gain and thus they will have no reason to deviate. Thus solutions found by Game Theory approaches would be very strong and reliable. However, Game Theory approaches seem to be the most problematic computationally and in general do not scale well to larger problems. As far as the authors know there has not been a Game Theory approach completed for this problem that has a polynomial time complexity.

Thus far, the approaches with the most success appear to be Reinforcement Learning approaches, which allow for the traffic signal control agents to learn dynamic strategies which react to the environment in real time. They are trained on simulation models to develop optimal strategies for immediate deployment so the slow learning process will not impair traffic conditions. Thus far, there have been successful results in simulation for many Reinforcement Learning approaches. However, this requirement for learning via simulation models is precisely the weakness of Reinforcement Learning, as simulations often may not accurately reflect how drivers actually learn and behave, and thus may not actually work with real life drivers. Furthermore, in the space of strategies available, Reinforcement Learning is known to risk converging to local optimums without detecting the true global optimum strategy, and thus solutions found by it may not be as strong as those found via Game Theory approaches.

# 6 Future Directions

As Reinforcement Learning approaches have had successes in simulation models, the obvious next step for them is to deploy them in the real world and critically evaluate their successes.

In addition, more work should be done on Game Theory approaches, such as extending the Temporal Action Graph Game model to work with stochastic games where agents action spaces may change very often in very structured manners. In particular, reducing the time complexity for finding, or even approximating the optimal Nash Equilibria would significantly contribute to solving the problem.

Furthermore, Game Theory approaches can be used to work in conjunction with Reinforcement Learning approaches by using the Game Theory Nash Equilibrium strategy as the starting strategy for the Reinforcement Learning agents and allowing them to learn from there. This combination would surpass both approaches individually, as the Reinforcement Learning agents are more likely to converge to global optimums when provided with good starting strategies, and the Game Theory solutions can be improved and adjusted to perturbations in the real world such as irrational behaviour from drivers.

# References


[1] A. Downs, *Still stuck in traffic: coping with peak-hour traffic congestion*, ser. James A. Johnson metro series. Brookings Institution Press, 2004. [Online]. Available: http://books.google.ca/books?id=ckLcxEb5tM8C

[2] S. Wolfram, *Theory and Applications of Cellular Automata*. World Scientific, 1986.

[3] M. Gerla and L. Kleinrock, "Flow control: A comparative survey," *IEEE Transactions on Communications*, vol. 28, pp. 553–575, 1980.

[4] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992, 10.1007/BF00992698. [Online]. Available: http://dx.doi.org/10.1007/BF00992698

[5] M. Wiering, "Multi-agent reinforcement learning for traffic light control," 2000.

[6] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds. Springer Berlin / Heidelberg, 2008, vol. 5211, pp. 656–671.

[7] A. L. Bazzan, D. de Oliveira, and B. C. da Silva, "Learning in groups of traffic signals," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 560 – 568, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197609001699

[8] Ella and Bingham, "Reinforcement learning in neurofuzzy traffic signal control," *European Journal of Operational Research*, vol. 131, no. 2, pp. 232 – 241, 2001, artificial Intelligence on Transportation Systems and Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221700001235

[9] A. Jafari, A. Greenwald, D. Gondek, and G. Ercal, "On no-regret learning, fictitious play, and nash equilibrium," in *In Proceedings of the Eighteenth International Conference on Machine Learning*. Springer, 2001, pp. 226–233.

[10] R. W. Rosenthal, "A class of games possessing pure-strategy nash equilibria," *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, 1973. [Online]. Available: http://www.springerlink.com/index/j5t4730452755627.pdf

[11] A. Fabrikant, C. Papadimitriou, and K. Talwar, "The complexity of pure nash equilibria," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, ser. STOC '04. New York, NY, USA: ACM, 2004, pp. 604–612.

[12] A. X. Jiang, K. Leyton-Brown, and A. Pfeffer, "Temporal action-graph games: a new representation for dynamic games," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, United States: AUAI Press, 2009, pp. 268–276. [Online]. Available: http://dl.acm.org/citation.cfm?id=1795114.1795146

[13] A. X. Jiang, K. Leyton-Brown, and N. A. Bhat, "Action-graph games," *Games and Economic Behavior*, vol. 71, no. 1, pp. 141 – 173, 2011, special Issue In Honor of John Nash. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0899825610001752

[14] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, ser. AAAI '98/IAAI '98. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998, pp. 746–752. [Online]. Available: http://dl.acm.org/citation.cfm?id=295240.295800

[15] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," *Proceedings of the eleventh international conference on machine learning*, vol. 157, pp. 157–163, 1994.

[16] J. Hu and M. P. Wellman, *Multiagent reinforcement learning: Theoretical framework and an algorithm*. Citeseer, vol. 242, pp. 242–250.

[17] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: A critical survey," *Technical Report*, 2003.