*Optimizing traffic flow using learning and the shout-ahead agent architecture*

Christian Roatis   
*Department of Computer Science*  
*University of Calgary*Calgary, Canada  
christian.roatis@ucalgary.ca

*Abstract*— We present a proposal for a research project in which we will attempt to improve traffic throughput in the traffic simulator SUMO by implementing the shout-ahead agent architecture and a hybrid behavior learning method for it into traffic signal controller agents. This architecture is based on two rule sets, one making decisions without communicated intentions of other agents, and one with these intentions. Reinforcement learning is used to determine which rule in a real set will be selected in a given situation. Evolutionary learning is used to learn these rules. We wish to apply the shout-ahead architecture to the traffic signal controller agents in order to improve their performance, evaluated via traffic throughput.

Keywords—traffic throughput, cooperative systems, reinforcement learning, evolutionary learning, shout ahead

# Introduction

For modern suburb dwellers in large metropolitan cities across the globe, the drive to and from work, often featuring long commute times, has become a part of life [1]. A 2016 Statistics Canada report found that 7% of all car commuters in Canada spend at least 60 minutes travelling to work, with that number jumping to 10% for car commuters in metropolitan areas [2]. Long commute times are generally a result of traffic jams caused by bottlenecks, rather than long distances, and traffic signal timing at intersections are a major factor in creating said bottlenecks [3]. Traffic signal timing can also hinder traffic flow at times without congestion, such as at night, where for instance vehicles may be held up at an intersection that is giving way to a direction with no oncoming vehicles.

Problems such as these string from traffic signal algorithms that do not effectively take into account traffic volume and flow, and lack the flexibility to adapt to the ever-changing traffic state. Specifically, traffic signal algorithms rely on pre-set timer cycles that can be adjusted based on demand, but do so in predetermined ways that do not optimally accommodate for the current state of an intersection [4]. These pre-set timing cycles are also why vehicles may be held up at an empty intersection in the middle of the night. Adjusting traffic signal behavior based on current traffic states could theoretically improve traffic flow, by jettisoning through heavier flows for longer during rush hour, and responding to demand immediately at empty intersections, to name a just a few examples. Theoretically, traffic flows could be optimized further if traffic signal algorithms were able to account for expected traffic flows in the immediate future and adjust their behaviors accordingly. follow.

# Proposed Solution

Every set of individual traffic signals in an intersection is controlled by an agent, which is responsible for changing their states. As discussed previously, the current algorithms for these agents are relatively static and inefficient, running predetermined cycles with little room for adjustment based on traffic volumes and flows [4]. This project aims to optimize traffic signal routines using Artificial Intelligence and Machine Learning principles, namely by applying the Shout-ahead agent architecture and a hybrid behavior learning method for it, to intersection controller agents [5]. The Shout-ahead agent architecture allows the use of communicated intentions of other agents, which in concert with information about an agent’s current state and environment, allows the agent the opportunity to take a more informed action at any given state than it would otherwise. The hybrid behavior learning method for the Shout-ahead architecture then works to refine the behavior of the intersection controller agent, aiming to improve the agent’s decision making such that the action being taken at any one time step, is the optimal action at said time step, or very near to it. The application of this approach was shown to be effective when applied to units in a computer game, thus the motivation for extending it to intersection controllers [5].

Within the Shout-ahead architecture, an agent holds two rule sets, each containing a number of rules with corresponding actions that the agent can apply at a given time-step. Each rule has a weight, which factors into both the rule selection process later on, and the hybrid behavior learning method. One rule set makes decisions without any communicated intentions of other agents, and the other makes them using mainly these intentions. This way, the agent can incorporate both information about itself, its environment and other agents into its overall decision-making process. This makes the agent’s behavior more dynamic and responsive to its environment at a given state. The agent will first choose a rule from the first set using rule weights, as well as some randomness for exploration, to determine its intended action, and communicates it to the other agents. It then chooses a rule from the second set, containing communicated intentions of other agents, using the same process involving rule weights and randomness. Finally, based on a mixture of looking at rule weights and using probabilities, the agent will select one of the rules and apply its corresponding action. We do however plan to make one adjustment to this process. An additional rule set will be added to each of the two previously described rule sets with user defined rules. Rules within this set will have maximum priority, such that if a rule within this set is applicable at a time step, it will always be chosen. If there are multiple applicable rules within this set at a given time, the one with the highest weight will be chosen. If no rules within this set are applicable, then the selection process will continue as described previously. This allows for prioritization of rare, but critical circumstances such as giving way to an emergency vehicle.

To apply the Shout-ahead architecture to our problem case, we must develop a set of predicates, which together describe the agent’s environment. Combinations of these predicates will make up rules, and dictate which rule may be applicable by an intersection controller agent at a given time. The design of these predicates will greatly affect how the agent behaves, and they will likely be adjusted as the implementation process proceeds to account for circumstances unforeseen at this point in time. Candidates for predicates include: (1) ones that consider how many vehicles are waiting to proceed through a given traffic light, (2) ones that consider the length of time the longest-waiting vehicle has waited to proceed through the intersection, and (3) ones that consider the length of time a particular phase (green or red) has lasted for, to name a few. Predicate candidates for the user-defined rule set could include ones considering whether there is an emergency vehicle waiting to proceed through the intersection, or how long a traffic light has been in the yellow phase.

We must also define a set of actions to be applied when a rule is selected by an intersection controller agent. Traffic lights in our study will adhere to the same signal phase changing conventions Canadian traffic lights do, and so possible actions are bounded by said conventions, too. As such, candidates for actions when a given rule is selected to be applied by an agent may include: (1) transitioning from a red phase to a green phase, (2) transitioning from a green phase to yellow phase, (3) transitioning from a yellow phase to a red phase, or (4) do-nothing, to name a few. More actions will be defined to account for the various traffic light configurations possible, namely phases for left-turn lanes, which can be controlled in concert with straight-ahead lanes of the same direction, or can have their own dedicated green phase. It is critical we provide the intersection controller agent with all possible actions such to not limit its ability to reach optimal behavior.

The hybrid behavior learning method for the Shout-ahead architecture is then responsible for optimizing the agent’s behavior. Two learning algorithms are employed by this method. The main one is an evolutionary algorithm, with each individual consisting of the two rule sets for each agent. New individuals are created using crossover and mutation operators, with respect paid to rule weights in selecting individuals to be used in these operators, and some additional random factors. Naturally, we need a fitness measure for an individual, which would be determined by doing simulation (training) runs. The agents perform a SARSA variant of Reinforcement Learning on their rule sets, to refine the weights of their existing rules, giving a higher priority to rules that have yielded positive results in the past and lower priority to the opposite [7]. This variant was shown to be effective with the Shout-ahead architecture when applied on unit agents in a video game [5]. These two learning phases each play crucial roles in improving the performance of the intersection controller agents, with the evolutionary algorithm providing new individuals that could potentially perform better than existing ones, and the reinforcement algorithm painting a better picture of what rules could be used to create better individuals. Altogether, if the objective is to increase traffic throughput, over time it is expected that individuals will be created that operate more efficiently than current intersection controller algorithms.

The Shout-ahead architecture and hybrid behavior learning method will be implemented within the traffic simulator SUMO (Simulation of Urban MObility) [6]. SUMO provides a robust environment for both the implementation of the agent architecture for multiple intersection types, as well as for simulating a comprehensive set of test cases and training simulations to determine the effectiveness of the Shout-ahead architecture in this context. The extendibility of the SUMO platform and the fact it was designed for research of this very nature make it an ideal tool for the purpose of this project.

Within SUMO, we will instantiate three types of intersection types, and thus three different classes of intersection controller agents.

### Standard 4-arm Intersection: The Standard 4-arm Intersection, Fig. 1 for example, is an obvious choice to be included, given its prevalence in road networks globally.

1. Example of a Standard 4-arm Intersection in SUMO.

### T- Intersection: The T-Intersection, Fig. 2 for example, is another prevalent intersection in road networks. Much like the Standard 4-arm Intersection, it is simple in design but presents many interesting test cases given the forced turning scenario presented by one of its straight phases ending.

1. Example of a T- Intersection in SUMO.

### 4-arm Incoming Layout Intersection: The third and final intersection type we plan to use is the 4-arm Incoming Layout Intersection, Fig. 3 for example. Unlike the other two, this intersection type is more complex, and potentially presents more bottleneck, or inefficient signal timing, opportunities.

1. Example of a 4-arm Incoming Layout Intersection in SUMO.

The design of this intersection is more irregular than the previous two selected, with a two-way road having two separate one-way roads converging in on it orthogonally, requiring at a minimum three separate green phases. This is unlike the Standard 4-arm and T-Intersections, whom theoretically only require a minimum of two green phases, should they choose to exclude dedicated left-turning phases. In real world applications, this intersection type is far less prevalent than the other two intersections, so its inclusion in our study allows us to see the effectiveness of the Shout-ahead architecture when applied to potentially more complex, irregular scenarios.

# Summary

Ineffective traffic signal timing algorithms can exacerbate poor traffic conditions, creating bottlenecks leading to traffic jams. Traffic signals within individual intersections are managed by a controller agent, which is responsible for changing signal states based on an algorithm that relies on a predetermined signal timing cycle. Though the cycle can be altered based on heavier traffic flow, applicable adjustments are also predetermined and thus don’t take into consideration the full state of the intersection, information that could lead to a different action being taken. The aim of this project is to incorporate the Shout-ahead agent architecture and a hybrid behavior learning method for it, into the intersection controller agent in hopes of improving traffic throughput.

##### References

1. M. Bennardo, “StatsCan study shows Canadian commute times are getting longer — and it's costing us,” *CBC*, March 2019. [Online]. Available: <https://www.cbc.ca/news/business/statistics-canada-commute-times-study-1.5038796>. [Accessed: Sept. 21, 2019].
2. T. Yaropud, J. Gilmore, S. LaRochelle-Côté, “Results from the 2016 Census: Long commutes to work by car,” *Statistics Canada*, February 2019. [Online]. Availble: <https://www150.statcan.gc.ca/n1/en/pub/75-006-x/2019001/article/00002-eng.pdf?st=wRNvAal2>. [Accessed: Sept. 21, 2019].
3. Y. Shaoxin, X. Zhao, Y. An, “Identification and optimization of traffic bottleneck with signal timing,” Journal of Traffic and Transportation Engineering (English Edition), vol. 2, pp. 353-361, October 2014.
4. National Academies of Sciences, Engineering, and Medicine, Signal Timing Manual – Second Edition, Washington, DC: The National Academies Press, 2015.
5. S. Paskaradevan, J. Denzinger, D. Wehr, “Learning cooperative behaviour for the shout-ahead architecture,” Web Intelligence and Agent Systems: An Internation Journal, vol. 12, pp. 309-324, October 2014.
6. M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO–simulation of urban mobility: an overview,” in Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011), 2011.
7. R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, The MIT Press, 1998.