Learning Traffic Signal Control with Advice

Patrick Mannion
Discipline of Information Technology
National University of Ireland Galway
p.mannion3@nuigalway.ie

Jim Duggan
Discipline of Information Technology
National University of Ireland Galway
jim.duggan@nuigalway.ie

Enda Howley
Discipline of Information Technology
National University of Ireland Galway
enda.howley@nuigalway.ie

ABSTRACT

The development of Adaptive Traffic Signal Control strategies for efficient urban traffic management is a major challenge faced by traffic engineers today. Reinforcement Learning (RL) has been shown to be a promising approach when applied to traffic signal control (TSC) problems. When using RL agents for TSC, difficulties may arise with learning speed and performance due to the high dimensionality of the state action space.

Potential-Based Advice is an emerging technique in RL literature, where learners are advised using knowledge specific to the problem environment. Previous works have shown this to be a promising approach, which can increase learning speed and improve an agent's performance. Up to now, Potential-Based Advice has mainly been tested on abstract problem domains. In this work, we apply Potential-Based Advice to a complex, real-world problem domain.

We extend previous work on RL for TSC by incorporating Potential-Based Advice based on heuristic knowledge relevant to the problem domain. We prove experimentally that the proposed method speeds up learning, and reduces delay times and queue lengths compared to a standard RL approach without advice.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Performance, Experimentation

Keywords

Reinforcement Learning, Potential-Based Advice, Multi Agent Systems, Intelligent Transportation Systems, Adaptive Traffic Signal Control, Smart Cities

1. INTRODUCTION

Traffic congestion is one of the major issues currently faced by modern cities. The many negative environmental, social and economic consequences of urban traffic congestion are well documented. High vehicle usage rates, along with the lack of space and public funds available to construct new transport infrastructure add to the significant challenges currently faced by traffic engineers. Against this backdrop, it is now necessary to develop intelligent and economical solutions to improve the quality of service for road users.

One relatively inexpensive way to alleviate the problem is to ensure optimal use of the existing road network, e.g. using Adaptive Traffic Signal Control (ATSC). Continued improvements in ATSC will have an important part to play in the future development of Smart Cities, especially in light of the current EU-wide emphasis on the theme of Smart, Green and Integrated Transport in Horizon 2020 [1]. Developing ATSC strategies for efficient urban traffic management is a challenging problem, and one which is not easily solved.

In recent years, several Artificial Intelligence (AI) methods such as Fuzzy Logic, Neural Networks, Genetic Algorithms and Reinforcement Learning have all been applied successfully to traffic control problems. These developments coincide with an increasing interest among researchers in the broader field of Intelligent Transportation Systems (ITS). The approach that we present in this paper is based on Reinforcement Learning (RL), a field that has many potential applications in the ITS area. RL algorithms have also been applied to other complex control problems besides ATSC, including air traffic control (see e.g. [21]).

In Reinforcement Learning for Traffic Signal Control (RL-TSC), each intersection is typically controlled by a single autonomous agent. Each agent has the responsibility of determining the light switching sequence at its assigned intersection, and learns a control policy by a process of continuous interaction with its environment. A network of traffic signal control agents may be considered as a Multi Agent System. This opens up possibilities in relation to agent coordination strategies to reach a global rather than local optimum.

RL-TSC approaches offer many benefits; RL agents have the capability to learn online to continuously improve their performance and thus adapt readily to changes in traffic demand patterns, and are capable of dealing with the incomplete information and stochastic nature inherent in this problem domain. Traffic control problems are a very attractive testbed for emerging RL approaches [5], because they present a number of interesting challenges such as developing strategies for coordination and information sharing between individual agents. Work by numerous authors has demonstrated that Reinforcement Learning is a promising approach for urban traffic signal control applications (e.g. [3, 7, 11, 16, 20]). The complexity and uncertainty of traffic signal control problems make them an extremely interesting application area for AI researchers to investigate.

The number of possible state action combinations for complex intersections with many phases present a significant challenge when applying RL to traffic signal control. RL literature refers to this problem as the Curse of Dimension-

ality. As RL agents are presented with increasingly complex problems, convergence times and the quality of the policy learned tend to degrade. When dealing with very large state action spaces, it may not be possible for the agent to visit each state action pair sufficiently often to learn a good policy within a reasonable timeframe. In general, as the problem complexity is increased an agent will require more experience in order to learn a good policy, which necessitates more training time. Potential-Based Advice is an emergent paradigm within RL that has been developed in recent years, which has the potential to help mitigate against the problems described above, by improving both learning speed and performance. Potential-Based Advice allows the designer to impart domain specific knowledge to the agent, in the form of a potential function. This knowledge can then guide the agent's actions in the environment.

The contributions of this research paper are as follows: 1) we extend previous works on RL-TSC by designing a Traffic Signal Control agent that learns guided by heuristic Look-Ahead Advice; 2) we prove experimentally that the proposed approach improves learning speeds on complex intersections and improves the quality of the policy learned compared to an agent without advice; 3) we identify and discuss specific issues that need to be taken into account when applying Potential-Based Advice to this difficult problem domain; 4) we discuss the future direction of this research topic and the wider implications for both ATSC and RL researchers.

The remainder of this paper is structured as follows: the second section discusses related research, while the third section describes our proposed approach. The following section details the design of our experimental set up, after which we present our experimental results. Finally, we conclude by discussing our findings and our plans for future work.

2. RELATED RESEARCH

2.1 Reinforcement Learning

Reinforcement Learning is an area within Machine Learning which has received considerable attention from AI researchers. RL agents are deployed into an environment, about which they generally have no prior knowledge. Instead, an RL agent must learn how to behave by a process of continuous interaction with its environment. The agent receives a scalar reward signal r based on the outcomes of previously selected actions, and by examining stored estimates for r for each state action pair the agent can decide which action to select when in a particular state. This reward signal can be either negative or positive, and a properly designed reward function will allow the agent to iteratively learn an optimal or near optimal control policy. The estimates for r are referred to as Q values, which are generally stored in a matrix. A balance must be struck between exploiting known good actions and exploring the consequences of new actions, and the ultimate goal of the agent is to maximise the reward received during its lifetime.

Markov Decision Processes (MDPs) are considered the de facto standard when formalising problems involving learning sequential decision making [23], and thus RL problems are generally modelled as an MDP. An MDP consists of a reward function R, set of states S, set of actions A, and a transition function T [17], i.e. a tuple $< S, A, T, R >$. Selecting an action $a \in A$ when in a state s $\in$ S, will result in the environment transitioning into a new state $s' \in S$ with probability $T(s, a, s') \in$ (0,1), and give a reward $r = R(s, a, s')$.

Two broad categories of RL algorithms exist: they are either model-based (e.g. Dyna, Prioritised Sweeping), or model-free (e.g. Q-Learning, SARSA). In the case of model-based approaches, it is necessary to know the transition function T for successful implementation [23]. This may be problematic, considering that T may be difficult or even impossible to determine in complex problem domains. Model-free approaches do not have this requirement, and instead they rely on sampling the underlying MDP in order to gain knowledge about the unknown model. This means that exploration is necessary for a model-free learner in order to gain the required knowledge about its environment, and the exploration vs exploitation dilemma discussed above must be balanced appropriately. The ϵ-greedy action selection strategy is an example of an approach commonly used to obtain the required balance.

Two model-free RL algorithms which are commonly used are Q-Learning, and SARSA. Q-Learning is an off-policy, model-free learning algorithm that has been frequently used in RL-TSC literature, e.g. [3, 9, 10, 11]. Q-learning has been proven to converge to the optimum action-values with probability 1 so long as all actions are repeatedly sampled in all states and the action-values are represented discretely [22]. In Q-Learning, Q values are updated according to the equation below:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r_t + \gamma Q_{\max_a}(s_{t+1}, a) - Q_t(s_t, a_t)) \quad (1)$$

Here the learning rate $\alpha \in [0, 1]$ determines by how much Q values are updated at each time step t. The discount factor $\gamma \in [0, 1]$ controls how the agent regards future rewards. A low value of γ results in an agent which is myopic, while higher values of γ make the agent more forward looking.

2.2 Incorporating Domain Knowledge

In Knowledge-Based RL, domain knowledge is incorporated to guide the agent in its action selection choices. Designers often have some knowledge about the problem domain, which can be used to speed up learning. Typically, RL agents begin learning with their Q-values initialised to zeroes, random numbers, or pessimistic/optimistic values. One of the simplest methods to incorporate domain knowledge is by Value Function Initialisation, where the Q-values are initialised in a meaningful way, rather than arbitrarily.

Reward Shaping is a promising and well studied approach which has been used successfully to speed up learning in both single and multi agent learning tasks. To use Reward Shaping, the designer adds an additional reward to the reward received from the environment. Reward Shaping generally takes the form below, where R is the environment reward, F is the shaping reward, and R' is the combined reward signal.

$$R' = R + F \quad (2)$$

While this technique is easily implemented and has been used successfully, care must be taken when designing the shaping reward to avoid undesired behaviour. A classic example of undesired effects due to reward shaping is shown by Randløv and Alstrøm [18], where an RL agent learns to ride a bicycle towards a goal. An additional shaping term was added to the reward function to encourage the agent to stay balanced. However, when poorly designed this shaping function caused the agent to converge to a policy where it

cycled in circles continuously to keep collecting the shaping reward, and never actually reached the goal state.

Potential-Based Reward Shaping (PBRS) was proposed by Ng et al. [15], in an attempt to overcome the problems with standard reward shaping approaches. When using PBRS, each possible system state has a certain potential, which allows the designer to express a preference for the agent to reach certain system states. The additional shaping reward F is defined as follows in Potential-Based Reward Shaping:

$$F(s, s') = \gamma\Phi(s') - \Phi(s) \tag{3}$$

where Φ is the potential function which maps states to potentials, and γ is the same discount factor used in the agent's Q update rule. Potential-Based Reward Shaping as defined by Ng et al. [15] has been proven not to alter the optimal policy of a single agent. Interestingly, Wiewieora [24] has proven that an agent learning with Potential-Based Reward Shaping and Q-values initialised to zero behaves the same as an agent learning without shaping whose Q-values have been initialised with Φ. PBRS is ideal for problems where an agent must reach a specific goal state (e.g. Gridworld, Mountain Car, etc.), where relatively simple potential functions can give a large increase in learning speed.

However, PBRS can only express a designer's preference for an agent to be in a certain state, and therefore cannot make use of domain knowledge that recommends actions. Wiewieora et al. [25] propose an extension to PBRS called Potential Based Advice (PBA), that includes actions as well as states in the potential function. The authors propose two methods of PBA: Look-Ahead Advice and Look-Back Advice. The former method defines the additional reward received F as follows:

$$F(s, a, s', a') = \gamma\Phi(s', a') - \Phi(s, a) \tag{4}$$

Wiewieora et al. [25] have provided a proof of policy invariance for Look-Ahead Advice for single agent learning scenarios. No corresponding proof has been provided for Look-Back Advice, although empirical results suggest that this method also does not alter the optimal policy in single agent learning scenarios. When using Look-Back Advice, Wiewiora et al. recommend the use of an on-policy algorithm such as SARSA. To maintain policy invariance when using Look-Ahead Advice, the agent must choose the action that has the maximum sum of both Q-value and potential:

$$\pi(s) = argmax_a(Q(s, a) + \Phi(s, a)) \tag{5}$$

Here $\pi(s)$ is the agent's policy in state s (the action that will be chosen by the agent in state s).

Recent work by Devlin et al. [19] examined the use of Potential-Based Reward Shaping and Potential-Based Advice in a Multi-Agent Reinforcement Learning (MARL) scenario, based on RoboCup Soccer. They found that by incorporating heuristic knowledge agents could learn a joint policy in less time and with equal or better performance than agents learning without this additional knowledge. However, the authors also found that the addition of shaping rewards could modify the joint policy learned. While in this paper we will consider only single agent learning problems, this work on MARL problems with advice has implications for any future works involving multiple traffic signal control agents learning with advice.

2.3 Reinforcement Learning for Traffic Signal Control

Numerous authors have studied the application of Reinforcement Learning to Traffic Signal Control problems in the last two decades, coinciding with an increased interest in ITS among researchers. Thorpe [20] presents some of the earliest work on RL-TSC, in which an approach based on SARSA is presented. This algorithm was benchmarked against a fixed time control scheme, and was found to offer significant performance increases compared to the latter approach.

Brys et al. [7] observed in their experiments that the objectives throughput and delay are correlated. They implemented a Multi-Objective RL-TSC algorithm, where the single objective reward signal is replaced with a scalarised signal, which was a weighted sum of the reward due to both objectives. They report that the proposed multi-objective approach exhibits a reduced convergence time, as well as decreasing the average delay in the network when compared to a single objective approach.

Abdoos et al. [3] developed an RL-TSC approach based on Q-Learning using Tile Coding as a method of Function Approximation. Their approach consists of an agent controlling each intersection, and these agents are grouped together under the control of superior agents. This hierarchical approach was tested against a standard Q-Learning approach on a 3 x 3 grid of intersections, and was found to reduce delay times in the network.

Pham et al. [16] present an RL-TSC system based on SARSA that also uses Tile Coding as a method of Function Approximation. In contrast to the approach above, in this system each SARSA agent is completely independent, and Tile Coding is used as a method of approximating the value function for the agent's local states.

Dresner and Stone [8] suggest the use of RL in combination with their Autonomous Intersection Management architecture. Here the intersection is treated as a marketplace where vehicles pay for passage or pay a premium for priority, and the RL agent's goal is to maximise the revenue collected. Thus in future, revenue collected could be used in reward functions for RL-TSC.

El-Tantway et al. [11] present a coordinated Multi Agent RL-TSC architecture called MARLIN-ATSC. This is a model-free architecture based on Q-Learning, where the state definition is based on queue length, and the reward definition is based on Total Cumulative Delay. The system is tested on a simulated network of 59 intersections in Downtown Toronto, and outperformed the currently implemented real world control scheme, resulting in a reduction in average delay, average stop time, travel times, queue lengths and emissions.

Mannion et al. [14] proposed a Parallel Reinforcement Learning algorithm for Traffic Signal Control. This framework allows multiple agents to learn in parallel on separate instances of the same TSC problem while sharing experience, with the goal of improving learning speed and exploration. The algorithm was tested on three intersections of varying complexity, and was found to offer statistically significant reductions in delay times and queue lengths as well as increasing learning and exploration rates when compared to a standard single agent approach.

For a more comprehensive review of the usage of learning agents in Traffic Signal Control, we refer the interested

reader to review papers published by Mannion et al. [13] and Bazzan and Klugl [6].

3. LEARNING TRAFFIC SIGNAL CONTROL WITH ADVICE

Traffic Signal Control presents a number of significant challenges when compared with the abstract problem domains (e.g. Gridworld) traditionally studied by RL researchers, due to the high degree of complexity and stochastic behaviour exhibited. In the simplest 2 Phase Traffic Signal Control scenario we consider, the number of possible discrete system states is of the order of 1.8×10^4, rising to approx. 8×10^5 states for the 3 Phase case. By contrast, a typical 50×50 Gridworld experiment has only 2.5×10^3 states. Traffic Signal Control is a continuous optimisation problem with no terminal goal state, whereas many traditional abstract problem domains have a goal state that the agent must reach.

The scale of transportation networks and the number of independent entities involved mean that an RL agent cannot possibly keep track of every detail about the environment state; therefore these problems are classified as Partially Observable Markov Decision Processes. Actions in Traffic Signal Control problems are not deterministic - i.e. the agent's action choice in a specific state is not the only factor that determines the next system state. Every additional variable considered in the representation of the environmental state increases the number of possible state action combinations, and transportation optimisation problems are thus challenging application domains for RL agents. The use of model-based RL algorithms in a highly stochastic problem domain like traffic control has been found to add unnecessary extra complexity when compared with model-free techniques [11].

For these reasons we have based our approach on a model-free RL method, namely Q-Learning. Here we extend previous works on RL-TSC by developing a method that incorporates Potential-Based Advice in order to speed up learning and improve agent performance in this complex problem domain. One of the more complex empirical studies using Potential-Based Advice is based on a problem domain with full observability [19]; however, our proposed application of Potential-Based Advice is based on a problem domain with partial observability. To the best of our knowledge, this is the most complex problem domain that Potential-Based Advice has been evaluated in thus far.

We develop two different agents that are identical in all respects, except that one of the agents receives Potential-Based Advice, and the other does not. The state, action and reward function definitions for our agents are similar to those used in other published works in RL-TSC (e.g. [4, 9, 10, 11]). Thus, our approach with advice could be considered to be an extension of these works. RL-TSC agents using these state, action and reward definitions have already been proven to offer considerable performance improvements compared to real world traffic control systems based on fixed-time control, semiactuated control, and SCOOT control [11]. In our experimental work we evaluate our agent with advice against this existing and proven approach, as other authors have already dealt with the efficacy of RL versus other techniques for Traffic Signal Control in a comprehensive manner.

Traffic engineers use the term Phase to refer to a specific traffic movement through an intersection, and determining an appropriate phasing sequence (order and duration in which phases move through the junction) is the main objective in TSC. The environmental state is defined as a vector of dimension $2 + P$, shown formally in Equation 6 below, where P is the number of phases at the junction. The first two components in the state definition are the index of the current phase (P_c) and the elapsed time in the current phase (PTE), while the remaining P components represent the queue lengths (QL_i) for each phase at the junction.

The state vector is thus constructed as follows for a given state s:

$$s = \left[\; P_c, \quad PTE, \quad QL_1, \quad ..., \quad QL_n \right] \tag{6}$$

By making use of a mixed radix conversion we represent the state vector for each possible state as a single number, which is used when setting and retrieving values in the Q values matrix.

The maximum number of queueing vehicles considered is limited to 20, and the maximum phase elapsed time considered is limited to 30 seconds. By imposing these limits, we reduce the number of possible environmental states considered by an agent. Even with these limits, over 18,500 discrete states are possible for a two phase junction. A vehicle is considered to be queueing at a junction if its approach speed is less than 10 km/hr. Limiting the number of queueing vehicles about which an agent knows to 20 adds further realism to our experiments, as in practice it would be prohibitively complex and expensive to detect queueing vehicles along the entire length of the approach lane.

At each time step t, the actions available to the agents are: to keep the currently displayed green and red signals, or to set a green light for a different phase. To eliminate unreasonably low durations from consideration, phases are subject to a minimum length. In the case of the 2 Phase test junction, the minimum phase length is 10 seconds, while the minimum phase length for the 3 and 4 Phase intersections is 5 seconds.

Agents are free to extend the current phase or switch to the next phase as they see fit, and there is no fixed cycle length. If an agent decides to switch phases, an amber signal is displayed for 3 seconds, followed by an all red period of 2 seconds, followed by a green signal to the next phase. This adds greater realism as it accounts for lost time due to phase switching, along with reducing the chances of vehicle collisions occurring.

Each agent selects actions using the ϵ-greedy strategy, where a random action is chosen with probability ϵ, or the action with the best expected reward is chosen with the remaining probability $1 - \epsilon$. The value of ϵ is set to 0.05 for all agents in these experiments. This value of ϵ promotes exploitation of the knowledge the agent has gained, while still allowing for sufficient exploration.

The reward function used by all agents is shown in Equation 7 below. When an agent selects an action a in a given state s and transitions to a resultant state s', the reward received is defined as the difference between the current and previous cumulative waiting times (CWT) of vehicles queueing at the junction. Therefore, actions that decrease the cumulative waiting time receive a positive reward, while actions that increase the cumulative waiting time incur a negative reward (or penalty).

$$R(s, a, s') = CWT_s - CWT_{s'} \tag{7}$$

We have decided to use Look-Ahead Advice to incorporate domain knowledge into our approach for two reasons. Firstly, a proof has been published that guarantees policy invariance when using LAA in a single agent learning scenario, while no corresponding proof has been published for Look-Back Advice. As the three problems we consider here are all single agent learning scenarios, the theoretical guarantees of this proof hold true. Secondly, the approaches that we wish to extend are based on Q-Learning, and Look-Back Advice requires the use of an on-policy algorithm such as SARSA. By using a different Q update rule, our approach would no longer be comparable to those published previously that use these state action and reward definitions, as all of these approaches are based on Q-Learning (e.g. [4, 9, 10, 11]). It is important that we use the same Q update rule in the basic and advised agents, in order to conduct a fair evaluation of the efficacy of Potential-Based Advice for Traffic Signal Control applications.

The potential function is defined as follows:

$$\Phi(s, a) = \frac{QL_a}{\Sigma QL} \tag{8}$$

Here QL_a is the queue length corresponding to the phase a, while ΣQL is the sum of the queue lengths for all of the phases at the junction. This approach could be considered to be similar to a longest queue first rule, as the potential of a state action pair is higher when the proportion of total queueing vehicles for that phase is higher. Therefore, the agent will be encouraged to give more green time to phases that have a higher number of vehicles waiting than the other phases.

4. EXPERIMENTAL DESIGN

We have based our experimental setup around the microscopic traffic simulation package SUMO (Simulation of Urban MObility) [12], and agent logic is defined in our external framework, which is implemented in Java. The simulation timestep length is set to 1 second for all experiments. We use the TraaS library [2] to make simulation parameters available to the agents, and also to send signal switching instructions from the agents back to SUMO.

All agents begin each experiment with their Q values for each state action pair initialised to zero. The values used for the learning rate α and the discount factor γ are 0.08 and 0.8 respectively. All learning agents in our experiments use these values of α and γ.

The RL agents with and without advice are evaluated experimentally using three different scenarios, which are based on the intersections shown in Figs. 1 to 3. The number of phases, lane configuration, and traffic demand levels differ for each test intersection. As work by other researchers has already proven the efficacy of RL-TSC approaches in test scenarios with multiple intersections (see e.g.[3, 7, 11, 16]), we have decided to focus on using single junction test cases to clearly illustrate the benefit of our RL approach using advice without adding unnecessary additional complexity.

The traffic demand D (measured in vehicles per hour) at each junction is generated using a step function. This demand step function is comprised of a episode length e, base flow b (baseline flow of vehicles through the junction), step demand increase h (the additional demand introduced at each step in the function), and a step interval i (duration in seconds between demand steps). Thus the demand at any time t into a particular episode can be calculated according to Equation 9 below:

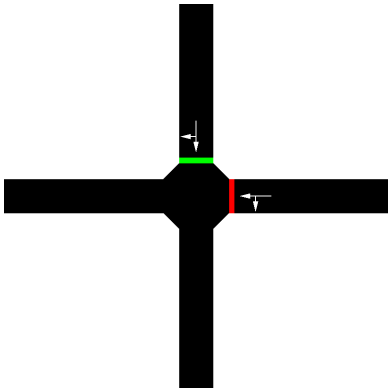

Figure 1: 2 Phase Junction

$$D(t) = \begin{cases} b + \dfrac{h \times t}{i} & t < \dfrac{e}{2} \\ b + h \times \left(\dfrac{e}{2 \times i} - 1 - \dfrac{t - 0.5 \times e}{i} \right) & \text{otherwise} \end{cases} \tag{9}$$

The increase in demand due to this function is computed at intervals equal to i. This time-varying traffic demand presents a more challenging flow pattern for the agents to control, compared to a constant hourly demand definition. These step functions aim to emulate peaks in demand similar to those during a morning or evening rush hour. Agents are trained on a junction for a number of successive episodes, and the same demand step function is repeated during each episode. An agent then builds up experience gradually over the training duration. For all intersections we use an episode length of 2 hours. Agents are trained for a period of 75 episodes on each intersection. There are a fixed number of possible routes through each intersection, and vehicles are randomly assigned to one of the possible routes upon insertion into the network.

The first test case is a simplified junction with 2 phases: North and East (see Fig. 1). Here two intersecting one-way streets are controlled by a set of traffic lights, with the number of phases $P = 2$. The base flow for the step function is 1000 vehicles per hour (veh/hr). The demand level rises by 250 veh/hr every 15 minutes, reaching a peak of 1750 veh/hr, before stepping down again to a value of 1000 veh/hr. There are four possible routes through the intersection.

The second intersection is a T junction, with three intersecting two-way streets (shown in Fig. 2). There are 6 possible routes through the junction, and three phases: North, East and South. The base flow is set to 1000 veh/hr, rising by 100 veh/hr every 15 minutes to a peak of 1300 veh/hr, before returning to 1000 veh/hr.

The final junction joins four streets with two-way traffic (see Fig. 3). This is divided into four phases: North, East, South and West. Here there are 12 possible routes through

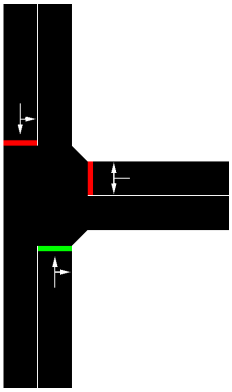

Figure 2: 3 Phase Junction

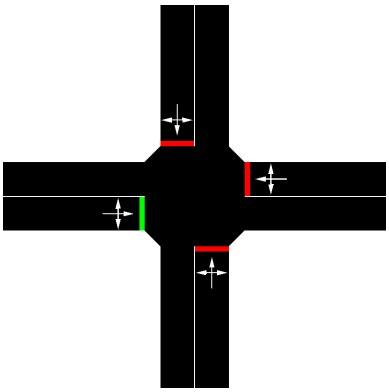

Figure 3: 4 Phase Junction

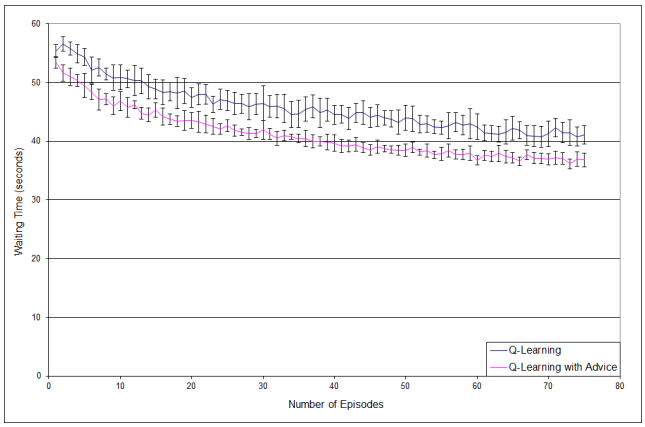

Figure 4: Waiting Times, 2 Phase Junction

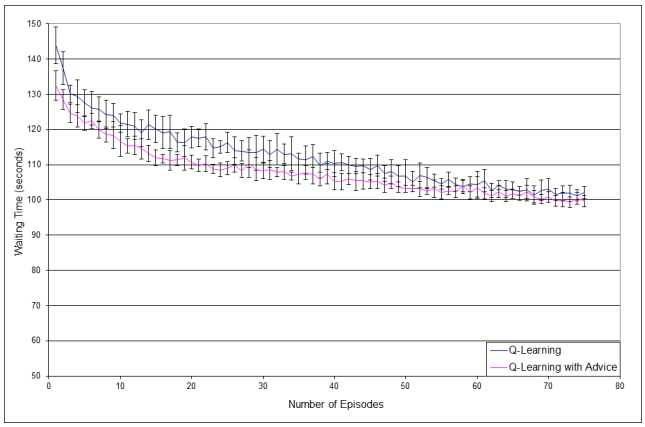

Figure 5: Waiting Times, 3 Phase Junction

the intersection. The base flow is set to 1000 veh/hr, which rises by 100 veh/hr every 15 minutes, giving a peak demand of 1300 veh/hr before reducing to the original value of 1000 veh/hr.

The following parameters are measured for each experiment: Average Waiting Times and Average Queue Lengths. We measure these values from SUMO using a customised data collection framework. The values reported for these parameters are the averages for each episode.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results are presented in Figures 4 to 7. These graphs plot the Average Waiting Times (AWT) and Average Queue Lengths (AQL) for approaches tested on each of the three intersections. The results for AWT and AQL are summarised in Tables 1 and 2. We conducted 10 statistical runs of all experiments to ensure consistency and repeatability, and the plots in Figures 4 to 7 show the average values measured.

In general, it is clear to see that our RL approach with advice outperforms a learner without advice on each of the intersections tested, with an increased rate of learning and better performance at the end of the training period.

Figs. 4 to 6 show the average vehicle waiting times (AWT) for the 2 Phase, 3 Phase, and 4 Phase junctions respectively. Each of these plots shows an improvement in AWT when advice is given to the learner. A clear increase in learning speed can be seen in these plots, with the agent receiving advice outperforming the agent without advice on all test intersections.

The Average Queue Lengths (AQL) for each experiment are plotted in Fig. 7. Similar to our findings in terms of AWT, our RL approach with advice improves performance, resulting in lower AQL values by the end of the training period in the case of the 2 and 3 Phase intersections. For the more complex 4 Phase test case, the AQL values see no noticable improvement.

To ensure the significance of our experimental results, a number of t-tests were conducted. The differences in the means were deemed to be significant if the two-tailed p-value was less than 0.05. For each experiment, we tested the mean values of waiting times and queue lengths over the final 10 episodes. Our approach using Look-Ahead Advice was found to have statistically better performance in terms of waiting times for all junctions tested at the end of the 75 episode training period. The reductions in AQL for both the 2 and 3 Phase intersections were also deemed to be statistically significant. In the case of the 4 Phase junction, there was no significant difference between the mean queue lengths at

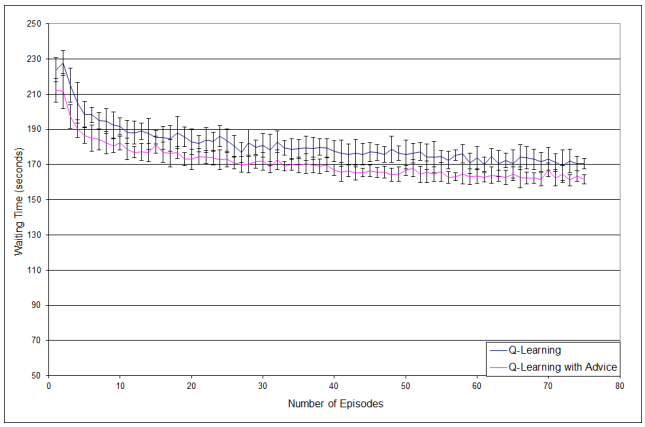

Figure 6: Waiting Times, 4 Phase Junction

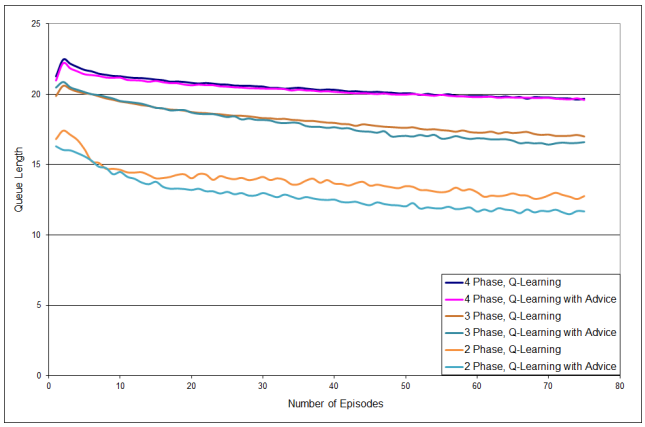

Figure 7: Queue Lengths, All Junctions

the end of the training period for both approaches tested.

For all test intersections our RL approach with Look-Ahead Advice was found to increase learning speed and to have reached a better policy by the end of the 75 episode training period when compared to a learner without advice. Reductions of up to 10% in waiting times were observed for the 2 Phase test case at the end of the training period, with encouraging results for the 3 and 4 Phase intersections also. Further refinement of our potential function is required to achieve the same level of performance in the more complex 3 and 4 Phase scenarios as seen in the 2 Phase case.

6. CONCLUSIONS AND FUTURE WORK

Here we have presented an application of Potential-Based Advice to three different traffic signal control problems. Our approach extends previously published works on Reinforcement Learning for Traffic Signal Control using Look-Ahead Advice with a domain specific heuristic as a potential function. We have proven experimentally that the proposed approach with Look-Ahead Advice outperforms a previously published Q-Learning approach, with improvements in learning speed, waiting times and queue lengths. This was achieved by testing our algorithm with advice on progressively more complex signalised junctions with time-varying traffic flow

Table 1: Summary of Experimental Results (Average Waiting Times over Final 10 Episodes)

Experiment	AWT	% Reduction	σ
2 Phase, Q-Learner	41.28	-	0.82
2 Phase, Q-Learner LAA	36.96	10.46 %	0.42
3 Phase, Q-Learner	102.11	-	0.42
3 Phase, Q-Learner LAA	100.46	1.62 %	0.66
4 Phase, Q-Learner	171.99	-	1.75
4 Phase, Q-Learner LAA	162.92	5.27 %	2.03

Table 2: Summary of Experimental Results (Average Queue Lengths over Final 10 Episodes)

Experiment	AQL	% Reduction	σ
2 Phase, Q-Learner	12.75	-	0.21
2 Phase, Q-Learner LAA	11.67	8.52 %	0.14
3 Phase, Q-Learner	17.13	-	0.09
3 Phase, Q-Learner LAA	16.53	3.49 %	0.08
4 Phase, Q-Learner	19.73	-	0.05
4 Phase, Q-Learner LAA	19.70	0.15 %	0.04

distributions. In contrast to previous empirical evaluations of Potential-Based Reward Shaping and Potential-Based Advice, our results are derived from an application to a complex real world problem domain.

While we have tested using only single isolated junctions, the benefits in terms of learning speed and performance are already clear compared to an RL approach without advice. In future we plan to test this method more extensively on traffic networks with multiple signalised junctions. Our experimental work in this paper only considered single agent learning scenarios, and a proof of policy invariance for Look-Ahead Advice exists for the single agent case. In future experimental work with multiple signalised junctions, we will have to consider the implications of the lack of a proof of policy invariance for LAA in these Multi Agent learning scenarios. The eventual goal is to test this method using a simulated real world traffic network to prove its merit when compared to commonly deployed traffic control strategies. We also wish to explore other alternatives to the potential function presented here that may offer further performance benefits.

7. ACKNOWLEDGMENTS

The primary author would like to acknowledge the financial support provided to him by the Irish Research Council, through the Government of Ireland Postgraduate Scholarship Scheme.

8. REFERENCES

[1] Horizon 2020, http://ec.europa.eu/programmes/horizon2020/en.

[2] Traas: Traci as a service, http://traas.sourceforge.net/cms/.

[3] M. Abdoos, N. Mozayani, and A. Bazzan. Hierarchical control of traffic signals using q-learning with tile coding. *Applied Intelligence*, 40(2):201–213, 2014.

[4] B. Abdulhai, R. Pringle, and G. Karakoulas. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3):278–285, 2003.

[5] A. L. C. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3):342–375, 2009.

[6] A. L. C. Bazzan and F. Klugl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29:375–403, 6 2014.

[7] T. Brys, T. T. Pham, and M. E. Taylor. Distributed learning and multi-objectivity in traffic light control. *Connection Science*, 26(1):65–83, 2014.

[8] K. Dresner and P. Stone. Multiagent traffic management: Opportunities for multiagent learning. In K. Tuyls, P. Hoen, K. Verbeeck, and S. Sen, editors, *Learning and Adaption in Multi-Agent Systems*, volume 3898 of *Lecture Notes in Computer Science*, pages 129–138. Springer Berlin Heidelberg, 2006.

[9] S. El-Tantawy and B. Abdulhai. An agent-based learning towards decentralized and coordinated traffic signal control. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 665–670, 2010.

[10] S. El-Tantawy and B. Abdulhai. Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc). In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 319–326, 2012.

[11] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto. *Intelligent Transportation Systems, IEEE Transactions on*, 14(3):1140–1150, 2013.

[12] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[13] P. Mannion, J. Duggan, and E. Howley. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In A. Kotsialos, F. Kluegl, L. McCluskey, J. P. Mueller, O. Rana, and R. Schumann, editors, *Autonomic Road Transport Support Systems*, Autonomic Systems. Birkhauser/Springer, 2015 (in press).

[14] P. Mannion, J. Duggan, and E. Howley. Parallel reinforcement learning for traffic signal control. In *Proceedings of the 4th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTRANS 2015)*, June 2015 (in press).

[15] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[16] T. Pham, T. Brys, and M. E. Taylor. Learning coordinated traffic light control. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2013)*, May 2013.

[17] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[18] J. Randløv and P. Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 463–471, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[19] M. G. Sam Devlin and D. Kudenko. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems*, 14(2):251–278, 2011.

[20] T. L. Thorpe and C. W. Anderson. Traffic light control using sarsa with three state representations. Technical report, IBM Corporation, 1996.

[21] K. Tumer and A. Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 330–337, Honolulu, HI, May 2007.

[22] C. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[23] M. Wiering and M. van Otterlo, editors. *Reinforcement Learning: State-of-the-Art*. Springer, 2012.

[24] E. Wiewiora. Potential-based shaping and q-value initialization are equivalent. *J. Artif. Int. Res.*, 19(1):205–208, Sept. 2003.

[25] E. Wiewiora, G. Cottrell, and C. Elkan. Principled methods for advising reinforcement learning agents. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 792–799, 2003.