Related works and 2 critiques for: 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*— I list 18 different works related to the topic of my research: Optimizing traffic flow using learning and the shout-ahead agent architecture. Additionally, I critique two of the works.**

Keywords—traffic flow, cooperative systems, reinforcement learning, evolutionary learning, shout ahead, related works

I. CRITIQUE 1

A. Overview

[1] introduces a hybrid cooperative behavior learning method for a rule-based shout-ahead architecture, which allows for the use of communicated intentions of other agents to create new agents which can cooperate with various other agents in fulfilling a predetermined task. The main objective of the paper is to describe the shout-ahead agent architecture and the hybrid learning method for cooperative behavior for agents using this architecture, and display its effectiveness when implemented into a video game, Battle for Wesnoth.

The Shout-ahead agent architecture described in [1] 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 agent, aiming to improve its 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. Within the Shout-ahead architecture, an agent holds two rule sets, each containing several 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.

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. A fitness measure for an individual is 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. These two learning phases each play unique roles in improving the performance of the 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.

[1] concludes that, in the context of learning cooperative behavior for units in a turn-based strategy game, the availability of shout-ahead intentions improved the performance of learned agents substantially. It is also noted however, that the selection of what predicates should be used in addition to communicated intentions can have a significant effect of the performance of learned behaviors.

B. Strengths

The work presented in [1] is extremely relevant, given the ever-growing relevance of artificial intelligence and

machine learning in society. The new theory presented in [1] provides a different approach to multi-agent systems, with agents benefiting from the communicated intentions of one another when making a decision for themselves. The theory presented in the paper, the shout-ahead architecture, is shown to hold merit through various rounds of experimental evaluation. The application of this approach increased the quality of learned behaviors for agents in the computer game Battle of Wesnoth that used only communicated intentions in the second rule set compared to agents not using shout-ahead. [1] also explored the impacts that different conditions in the second rule set had on the success of agents with the shout-ahead architecture.

Another strength of this paper lies in the depth of the descriptions provided for the shout-ahead architecture and the hybrid behavior learning method for it. [1] doesn't dive straight into the architecture description itself, but rather readies its readers with foundational information on basic concepts and definitions, such as the definition of an agent and the methods of evolutionary and reinforcement learning that will be utilized. The description of the actual rule-based shout-ahead agent architecture is then well detailed, presented in a step-by-step format, with each component of it robustly explained. The same can be said for the hybrid learning method for cooperative behavior, which is accompanied by a diagram for better understanding its process. Though the Battle of Wesnoth is used to test the effectiveness of the new architecture, the detailed presentation of its components makes for easier extendibility, with everything one would need to apply this to a new field provided within [1]. This is a major strength of the paper, as it decreases the barrier of entry for future work on this topic. While this paper may not immediately impact the way the world works, or even the world of Artificial Intelligence and Machine Learning, it does provide a useful tool to the field of collaborative learning, which may prove relevant in the future, and a new avenue of approaching multi-agent learning.

The experiments are chosen carefully, with game context and situational diversity clearly in mind. They vary in complexity, from simple shout-ahead vs. not, to ones investigating the usefulness of different types of information. The findings from these experiments not only supported the legitimacy of the architecture itself, they could also prove beneficial for the extendibility prospects of the theory, with future work taking into consideration what may improve the quality of the architecture, and what may not. A description of the Battle of Wesnoth beforehand also affords a preliminary understanding of what the tests may entail, a useful context for understanding both the tests and the results.

C. *Weaknesses*

A weakness of [1] can be identified as the, at times, convoluted and complicated descriptions of various components of the agent architecture and hybrid learning method. The paper introduces a great number of abbreviations and acronyms, all of which are important to understanding the inner workings of the architecture and learning method, but make for a difficult read at times. Interrupting reading to revisit what an abbreviation or acronym pertains to slowed down the process of digesting what is already a greatly complex paper. Understanding equations that are primarily made up of abbreviations and acronyms is particularly challenging. This results in multiple readings, or note taking alongside reading, to fully understand the concepts and descriptions put forth by [1]. The paper could be improved with a dedicated section describing all relevant acronyms, perhaps arranged by the components they pertain to, for easy cross-referencing during reading.

[1] is also limited by the premise chosen to test the architecture itself. While the results are positive when shout-ahead was implemented into the Battle of Wesnoth, these results cannot be generalized. It is possible the shout-ahead architecture and hybrid behavior learning method for it are only effective in the context of the Battle of Wesnoth, and cannot be extended to other games or applications. Subsequent implementation of the architecture to another game or application, with similar results observed, would go a long way in supporting the general usefulness and effectiveness of the architecture presented in [1].

II. CRITIQUE 2

A. *Overview*

[2] introduces a new algorithm which aims to apply multi-agent reinforcement learning to increase the effectiveness of traffic light signal controllers, given the growing concern regarding traffic congestion. Specifically, it aims to formulate the "traffic signal control (TSC) problem" as a discounted cost Markov decision process (MDP), with multi-agent reinforcement learning (MARL) algorithms to generate policies for governing traffic signals, with the goal of decreasing vehicle wait times more effectively than currently implemented TSC algorithms, Fixed Signal Timing (FST) and Saturation Balancing (SAT). Agents update their Q-factors using Q-learning with either $\varepsilon -$ greedy or UCB based exploration strategies, meaning two algorithms will be provided. Additionally, they will make use of a feedback cost signal obtained from neighboring agents, which allow an agent to determine the cost of their action on neighboring junctions. Based on these factors, the agent then chooses how long a green phase should last for. For this purpose, it views each traffic intersection as an agent. Due to the exponential growth in size and complexity of the state and action space as the number of intersections in a road network grows, approximation methods are detailed to be needed for solving the MDP.

The TSC problem is described as a controlled Markov process, for which actions are chosen in each state such that the certain long-term cost is minimized. A state for a given junction (or intersection) is described as a vector of dimension L + 1, with L denoting the number of incoming lanes into said junction. A state vector is defined, with i^{th} component in the vector representing the queue-length in the i^{th} lane. The last component in the vector gives the index of the phase that must

be set green in the round-robin schedule. Abstractions are also provided as a way of dealing with the massive possible state space depending on the size of any one junction. Traffic volume is abstracted into three portions: low, medium and high. In a similar vein to the handling of state spaces (and their potential for exponential growths in size), action spaces are considered individually per junction. Action space too is discretized into low, medium and high designations, each pertaining to a certain phase length. A policy is described as a sequence of maps from the state space to the action space, such that when the state is at a time *t*, the policy specifies the time duration for the current phase. [2] only considers stationary deterministic policies that do not change with time, with the end goal being to acquire individual policies for every junction, that minimize the delay of road users. Interwoven through the problem description are notes of how the different definitions were instantiated during experimental evaluation, as they're presented. Finally, a definition for a cost function applied for any action is provided. These costs are used for different purposes; to evaluate the effect of an action an agent may take, and the effect said action would have on neighboring junctions.

As mentioned, the learning algorithm presented in [2] is based on Q-learning. This algorithm both updates Q-factors, and obtains the actual TSC policies. The Q-factors for any one policy indicate the quality of an action in a given state if the action is applied in said state, and then follows a given policy. The algorithm tries different actions in a given state, calculating the cost of said action, searching for the action providing minimal cost. An update rule for the Q-function, based on learning, is provided, considering current queue sizes and then setting the next green phase length based either on ε −greedy exploration, or the UCB exploration strategy. Given the MDP framework, a random action may also be selected based on some probability, to balance the exploration versus exploitation trade off.

It is found that the performance of [2]'s algorithm was significantly better than both the FST and SAT algorithms. Of the two types of exploration used by the algorithm, ε −greedy and UCB, it was the latter which proved more effective, given the faster exploration tendencies. Both algorithms performed better on the twenty-junction road network than on the nine-junction network. It was also observed that the policies obtained from their algorithms were able to produce self-organizing behavior of traffic lights.

B. Strengths

[2] presents both strong argumentation for the purpose of their work, and the merits of their solution based on past work in this field. Traffic congestion is a world-wide problem desperately searching for a solution that has yet remained elusive, and work done in attempting to ease the problem is extremely relevant. The improvements and changes on past solutions that the authors propose to incorporate in their algorithm are sound, as proven by the results of their experiments. Many decisions that were taken in designing the algorithm, such as treating each individual junction as an agent and modelling the problem itself as a Markov decision process (MDP), were supported as effective. The authors go into detail explaining every facet of their research, from descriptions of the problem as an MDP and the ensuing Q-learning based algorithm, to the actual instantiation of the algorithm during experimental evaluation. Enough is provided for the research to be replicated and extended, should the desire to do so exist. Additionally, the intertwined analysis and comparison of the FST and SAT algorithms to [2]'s algorithm highlights examples of improvement and optimization efforts, which are then supported during testing. Given the purpose of the research is to improve current TSC solutions, the conscious effort to occasionally contrast different facets of the new solution with those of the old ones allows for a better understanding of why [2]'s algorithm is an improvement, as opposed to simply being presented as one.

The experimental evaluation is well designed and described. Real world road networks are selected, a sensical decision given the purpose of the research is to find an improvement on real-world TSC algorithms. Instantiations of both [2]'s algorithm and the experimental evaluation are clearly and fully described. The ability to directly test the new algorithm alongside the two it is aiming to better, under identical circumstances, is a strong method of testing the superiority of the new algorithm. Additionally, analysis of the results is well presented and highlights important observations. Both the discrepancies between the Q-UCB algorithm and the Q- ε −greedy variation, as well as the performance on the larger road network versus the smaller one, could influence future work in the field. The real-world impact of [2] will fall into the hands of politicians and city-planners, as the responsibility for real-world adoption falls to them. The early returns presented by [2]'s algorithm are positive and seem to accomplish the goal of improving on what is currently in use, but are they good enough to warrant actual adoption? That question remains to be answered.

C. Weaknesses

The readability of [2] due to a confusing use of acronyms can be identified as a primary weakness. While using acronyms can make reading easier to understand, [2] uses acronyms for names that may not merit one given their relatively sparse usage in the text. Due to said usage it can be easy to forget what the acronym actually pertains to, defeating the purpose of using acronyms in the first place. This stalls the flow of the paper at multiple points, and shifts the focus away from the content on hand, to trying to remember which acronym stands for what. Some acronyms are far too similar to each other, a consequence of assigning acronyms to names that may not merit them, further convoluting things. At one point, an acronym is assigned as an extension of another acronym, without using the actual name pertaining to the acronym being extended. These shortcuts distract from the ideas being presented and, in some cases, make it harder to understand what the point being made is. Unclear, meaningless variable names used in various definitions add to the difficulty of the read. [2] would benefit greatly from detailed variable names, to allow readers to better follow along with mathematical equations and statements made in the problem description and instantiation. The authors also give superficial explanations for the use of multi-agent reinforcement learning in the solution. Simply stating

reinforcement learning is well-suited because “they are online in nature and learn good control strategies from experience” is rather shallow and insincere, given the simple implementation of reinforcement learning does not alone guarantee the learning of good control strategies.

The experimental evaluation section of [2] is shallow, with tests run on just two different road networks. Though the progression of results as learning progresses is provided in this section, the overall test suite utilized in [2] lacks depth and variance. At minimum, three test networks should’ve been utilized, but five or more are realistically required for a robust evaluation. Additionally, [2] does not account for different junction types, rather treating all of them as equal. The study would benefit from analysis on the effectiveness of their algorithm on different junctions, and if the nature of the junction affects performance in any way, and if-so, an analysis of such. Furthermore, only one vehicle flow was observed. Set values of cars entering the networks per hour were provided, and all the testing was based off this. This further diminishes the robustness of the evaluation. It raises questions such as “how does different vehicle flow impact the effectiveness of the algorithms?”. [2] provides a good instance in which their algorithm performs better than FST and SAT algorithms but fails to inspire confidence that this is true in most cases. To better test the validity of their traffic signal algorithm, multiple flows should’ve been observed, including one or more edge cases, on more, different road networks. [2] also fails to detail possibilities for future work in a satisfactory manner. The success of the experimental evaluation, though limited in scope, would suggest runway for future work to extend and build upon [2], but the authors provide no direction in this regard.

RELATED WORK

[1] S. Paskaradevan and J. Denzinger, “A Hybrid Cooperative Behavior Learning Method for a Rule-Based Shout-Ahead Architecture,” *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2012.

[2] K. J. Prabuchandran, A. N. Hemanth Kumar and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 2529-2534.

[3] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, “Multiagent Reinforcement Learning for Urban Traffic Control Using Coordination Graphs,” *Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science*, pp. 656–671, 2008.

[4] M. Abdoos, N. Mozayani and A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning," *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, 2011, pp. 1580-1585.

[5] P. Gregoire, C. Desjardins, J. Laumonier and B. Chaib-draa, "Urban Traffic Control Based on Learning Agents," *2007 IEEE Intelligent Transportation Systems Conference*, Seattle, WA, 2007, pp. 916-921.

[6] M. Wiering, “Multi-Agent Reinforcement Learning for Traffic Light Control,” *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, Stanford, CA, 2000.

[7] K. Dresner and P. Stone, “Multiagent Traffic Management: Opportunities for Multiagent Learning,” *Learning and Adaption in Multi-Agent Systems Lecture Notes in Computer Science*, pp. 129–138, 2006.

[8] D. de Oliveira Boschetti, A. Bazzan, B. da Silva, E. Basso and L. Nunes, “Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: A Case Study in a Microscopic Simulator,” *Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06*, Lisbon, Portugal, 2006.

[9] E. Camponogara and W. Kraus, “Distributed Learning Agents in Urban Traffic Control,” *Progress in Artificial Intelligence Lecture Notes in Computer Science*, pp. 324–335, 2003.

[10] M. A. Khamis and W. Gomaa, "Enhanced multiagent multi-objective reinforcement learning for urban traffic light control," *2012 11th International Conference on Machine Learning and Applications*, Boca Raton, FL, 2012, pp. 586-591.

[11] I. Arel, C. Liu, T. Urbanik and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128-135, 2010.

[12] E. Van der Pol and F.A. Oliehoek, “Coordinated Deep Reinforcement Learners for Traffic Light Control,” *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*, December 2016.

[13] D. Houli, L. Zhiheng, and Z. Yi, “Multiobjective Reinforcement Learning for Traffic Signal Control Using Vehicular Ad Hoc Network,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, 2010.

[14] J. Jin and X. Ma, “Hierarchical multi-agent control of traffic lights based on collective learning,” *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 236–248, 2018.

[15] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu and M. Wu, "Machine Learning for Vehicular Networks: Recent Advances and Application Examples," in *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 94-101, June 2018.

[16] S. El-Tantawy and B. Abdulhai, “Multi-Agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC),” *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012.

[17] B. Bakker, S. Whiteson, L. Kester, and F. C. A. Groen, “Traffic Light Control by Multiagent Reinforcement Learning Systems,” *Interactive Collaborative Information Systems Studies in Computational Intelligence*, pp. 475–510, 2010.

[18] D. D. Oliveira and A. L. Bazzan, “Multiagent Learning on Traffic Lights Control: Effects of Using Shared Information,” *Multi-Agent Systems for Traffic and Transportation*, pp. 307–321, 2009.