Journal of the Nigerian Association of Mathematical Physics
Volume 52 (*July & Sept., 2019 Issue*), pp165 – 170

A DYNAMIC ROAD TRAFFIC ROUTING MODEL USING Q LEARNING AND MARKOV DECISION PROCESS

O. A. Sanya and O.A. Bello

Department of Mathematical and Physical Sciences, College of Sciences, Afe Babalola University, Ado Ekiti, Ekiti State, Nigeria

Abstract

Road traffic delay has become a menacing challenge in cities around the world. This research work gives an overview of a dynamic road traffic routing system that can help in determining optimized path from one point to another in a given road network in terms of path and the associated delay. This is achieved by simulating real-world traffic situations on selected road paths in the University of Lagos. Q Learning, a reinforcement learning algorithm was used to facilitate an efficient and dynamic handling of parameters. Q Learning is even more appealing because it inherently has routing capability and it easily allows us to train an agent that represents a road user to make optimal decision. The result indicated that the system will always select a route that is optimal in relation to prescribed constraints. Markov Decision Process (MDP) was adopted in modeling the research

Keywords: Routing, Q – Learning, Reinforcement Learning, Markov Decision Process**.**

1. Introduction

Time places a strict constraint on all of human endeavors such that there is a deadline which must be met if one is to subsist. Hence, path selection and optimization is as important as human existence itself. It is central to all of mankind's action, hence, success or failure. Making a certain decision or a set of decisions among a collection of several possible decisions requires some sort of means or mechanism to determine that the decision taken is the best possible alternative at that particular instance of time and that the forgone alternatives would not yield a better outcome at that particular instance [1]. Road users are faced with this challenge of decision making every now and then. Another factor for consideration is environmental traffic complexity that requires that a traffic agent be independent, reliable, adaptive and correct in its decision making while navigating through an environment [2].

This research work models selected road paths together with the associated path lengths and path weights in the road network of the University of Lagos. It then represents random scenarios of traffic density (number of vehicles per road length) on fixed-length paths and presents perceived optimal path based on prescribed conditions. Our interest is to combine a sequence of optimized points in getting to a destination with minimal delay.

Our approach is the use of a learning agent to learn how best to navigate along the points in the model environment under dynamic conditions.

2. Problem Statement

Path planning and optimization problem is a classical combinatorial problem explored and applied in many areas of human endeavor. The major challenge is that commuters are not able to determine the desirability and optimality in terms of delay on any road path at any given time.

Usually, an agent may know the shortest route to a destination but may not always choose an alternative route to the same destination which may now be more favourable due to delays on the usually optimized route.

Corresponding Author**:** Sanya O.A., Email: sanyaluwafemi@abuad.edu.ng, Tel: +2348034402700, +2348166360838 (OAB)

3. Related Literature

Routing problem lies at the heart of distribution management vis-à-vis free movement of people, goods, services, labour and capital. It is faced each day by thousands of companies, organizations and individuals engaged in the delivery and collection of goods or people faced with the challenge of quickly moving from point A to point B. Because conditions vary from one setting to another, the objectives and constraints encountered in practice are highly variable. Most algorithmic research and software development in this area focus on a limited number of prototype problems. This project is a problem in the class of time dependent dynamic vehicle routing. It models scenarios of changing traffic delay along different routes and makes a learning agent take a more computationally appealing route to the expected destination.

The observation that the flow (and velocity) of traffic was dependent on the traffic density was made in early traffic studies [3]. Since then, the concept has been used as a basis for traffic simulation formulations.

Some studies in literature make use of a heuristic approach when a road map is characterized by tightly interconnected network of nodes as the complexity of finding the shortest path could only be estimated in real-time [4].

This research work is perhaps most similar to that of [5-6]. It directly extends these methods to perform self-aware traffic routing given that the actual road conditions like road length and traffic density can be known for any given travel time. [7] proposed a method to optimally route cars given uncertain information about travel times within a road network. [8] provides an optimization to the procedure that allows fewer paths to be explored, while optimizing for a specific arrival deadline.

Another area of similar work is the study of Dynamic Traffic Assignment (DTA) done primarily in Civil Engineering. This problem involves flows of traffic from known origins to destinations (OD flows). The solution approaches attempt to optimally route all the flows in order to maximize aggregate or individual statistics.

The most related of these approaches are the simulation methods. In these approaches, cars are iteratively routed and simulated. The simulation provides the estimate of the network state that is used for the next iteration of routing. Over a number of iterations, the routes settle into equilibrium. By building enough flexibility in optimization systems, one can adapt these to various practical contexts.

4. Research Model

Markov decision process (MDP) is the underlying model upon which this research work is based. It is one of the many combinatorial design models. It is a feedback compliant system that ensures every action is rewarded and the reward helps the learning agent to make smarter decisions as the learning activity progresses [9]. MDP is defined as a 4-tuple (S,A,P,R), where:

$S = \{s1, s2, ..., sn\}$ is a finite set of states;

$A = \{a1, a2, ..., am\}$ is a finite set of actions;

P is a Markovian state transition model — $P(s, a, s\_)$ is the probability of making a transition to state $s\_$ when taking action a in state s($s \xrightarrow{-a} s\_$);

R is a reward (or cost) function — $R(s, a, s\_)$ is the reward for the transition $s \xrightarrow{-a} s\_$.

Specifically, the Q Learning algorithm was used to facilitate a more efficient and dynamic handling of parameters. The algorithm was proposed as way to optimize solutions in Markov Decision Process problems. Its distinctive feature is its ability to choose between immediate rewards and delayed rewards [10].

The proposed system is modeled as a control theory problem using MDP. It is assumed that the road points are the set of possible states S; Forward-direct movement, redirection and shut down as the possible action sets A, a road map matrix indicating the connection among the road points as the Markov transition matrix P, and a reward scheme R. These four MDP components are sufficient to describe the framework and serve as moderating factors for the learning algorithm.

The model also includes a learning matrix which captures the value function of the agent's traversals from state to state. The rows of the learning matrix represent the current state of the agent, and the columns represent the possible actions leading to the next state (the links between the nodes). The agent starts out knowing nothing, hence the matrix Q is initialized to zero. It basically represents the long-term value of taking an action in a given state. The learning agent starts out without any knowledge of the environment vis-a-vis traffic conditions on the road networks. Each time the agent traverses through a road network, the learning matrix is updated with Q-values of the actions in the various states that the agent transits through.

Hence, the description of the components of the MDP in relation to the proposed model is as follows:

Table 1: Road network at UNILAG

S/N	Road Path	Notation	Connection
1	Engineering - Botanical	S_1	S_2, S_3
2	UBA -Vc's Lodge	S_2	S_1, S_5
3	Senate -Guest House	S_3	S_1, S_4
4	Law - Mechanical works Junction	S_4	S_3, S_6
5	UBA - Mechanical works Junction	S_5	S_2, S_6
6	Mechanical works Junction – Science Junction	S_6	S_2, S_8
7	Science Junction – Science Faculty	S_7	S_6, S_8
8	Science Junction – Medical Junction	S_8	S_6, S_7, S_9
9	Medical Junction – DLI Junction	S_9	S_8

Table 2: State transition matrix

S	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	F	T	T	F	F	F	F	F	F
S_2	T	F	F	F	T	F	F	F	F
S_3	T	F	F	T	F	F	F	F	F
S_4	F	F	T	F	F	T	F	F	F
S_5	F	T	F	F	F	T	F	F	F
S_6	F	T	F	F	F	F	F	T	F
S_7	F	F	F	F	F	T	F	T	F
S_8	F	F	F	F	F	T	T	F	T
S_9	F	F	F	F	F	F	F	T	F

T indicates link between nodes and F indicates no link between nodes.

Matrix Q – The learning matrix of the agent

The learning matrix is used to capture the value function of the agent's traversals from state to state. It basically represents the long-term value of taking an action in a given state. The learning agent starts out without any knowledge of the environment vis-a-vis traffic conditions on the road networks. Each time the agent traverses through a road network, the learning matrix is updated with Q-values of the actions in the various states that the agent transits through.

Table 3: Learning Matrix

S	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	0	0	0	0	0	0	0	0	0
S2	0	0	0	0	0	0	0	0	0
S3	0	0	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0
S5	0	0	0	0	0	0	0	0	0
S6	0	0	0	0	0	0	0	0	0
S7	0	0	0	0	0	0	0	0	0
S8	0	0	0	0	0	0	0	0	0
S9	0	0	0	0	0	0	0	0	0

Routing and Learning Procedure

The agent starts out knowing nothing, hence the matrix Q is initialized to zero. The virtual agent will learn through experience, without a teacher (unsupervised learning). The agent will explore from state to state until it reaches the goal. After each exploration, the learning matrix will be updated with optimized q values.

Policy

The policy prescribes action for every state of the learning agent [11]. Also, there is a probability distribution with which one can predict the action the agent might take for every state. $\boldsymbol{\alpha}$ = Learning rate of the agent as the activity continues. This policy is any value between 0 and 1. When values closer to 0 are chosen, the agent tends to be myopic thereby considering only immediate rewards. The value of 0.5 has been chosen to balance the desirability of long-term rewards against immediate rewards. Thus, the agent receives a reward/penalty of 0.5 for every non – terminal action taken.

1. $\alpha = 0.5$ ……(1)
2. $Q(s, a) = R(s, a) + (\alpha * Max [Q(s, a)])$ ……(2)

[12] *Q(s, a)* the value of taking an action a, in state s.
R(s, a) is the immediate reward the agent gets from *Q(s, a)*
*(α * Max [Q(s, a)])* is the discounted reward. It represents the q-value of every possible action leading to next state/s for the agent.

3. *Traffic Density, TD = N/L* …… (3)

(Delay is the number of vehicles, N on the Road, Weight is length, L of the road)
Therefore equation (2) becomes

4. $Q(s, a) = R(s, a) + (\alpha * Max [Q(s, a)]) + TD$ ……(4)

Where:
TD = Traffic density i.e. the delay when N number of vehicles are on a road path whose length is L.
R(s, a) = Reward got for an action a, to move to state, s.
Q(s, a) = Q-value of the learning agent at state s for an action a. The traffic density is added to the basic Q-learning formula to acknowledge the effect of traffic delay on the learning exercise.
The recurrence formula below describes the learning policy for the vehicular agent. The policy prescribes the suitable conditions that will lead to an optimal value by selecting next states using the appropriate schemes [13-17].

$$\Pi(s) = \begin{cases} (Q = 0 \ \& \ S \neq GS) & \textit{Select state by Random Selection} \\ (Q > 0 \ \& \ S \neq GS) & \textit{Select state by maximum Q value} \\ (Q \geq 0 \ \ \& \ S = GS) & \textit{Goal State (GS), Stop} \end{cases}$$

The implementation of the research was done using C# programming language.

Research Relevance

This research framework can be fully commercialized or made available as an accessible software to the University community. It will significantly enhance the navigation judgments of commuters when there is heavy traffic load on usually optimal paths to take other paths that are not as optimal but having lower traffic costs to the same destination. It can also be proposed to government and private agencies that may be interested in developing the application into a fully commercial application in managing the teeming national road network.

Table 4: Updated Learning Matrix

S	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	0	8.5	0	11	0	0	0	0	0
S2	0	0	0	0	0	0	0	0	0
S3	0	0	0	0	0	0	0	0	0
S4	0	0	10.83	0	0	11	0	0	0
S5	0	0	0	0	0	0	0	0	0
S6	0	8.5	0	0	0	0	0	10	0
S7	0	0	0	0	0	0	0	0	0
S8	0	0	0	0	0	11	10.8	0	11
S9	0	0	0	0	0	0	0	0	0

Conclusion

The updated learning matrix represents the value function for each state-action pair from the starting state-action pair to the destination state-action pair. This is the long-term reward that determines the desirability of every state in the learning matrix. The node with the highest Q-Value represents the state/action pair that translates to optimal path selection.
This research work attempted to achieve optimal traffic routing by considering not just the road networks but recognizing other factors like the road length and the traffic density as well. This makes the research work compliant with actual road traffic routing scenarios where the major reason for non-optimal routing can be attributed to the number of vehicles on a road network [18-23].
This project work is a veritable solution to traffic delay and congestion; hence a lot is still to be desired in terms of further research to enhance the solution that can be derived from it.

REFERENCES

[1] Johan Kwisthout. 2009. The Computational Complexity of Probabilistic Networks. Ph.D. thesis, University of Utrecht.

[2] Richard S. Sutton and Andrew G. Barto. "*Reinforcement Learning: An Introduction*," MIT Press, Cambridge, MA, A Bradford Book, pp61-65, 2002.

[3] Lim, S.; Balakrishnan, H.; Gifford,D.; Madden, S.; and Rus, D. 2009. Stochastic Motion Planning and Applications to Traffic. Algorithmic Foundation of Robotics VIII 483–500.

[4] Orda .A. *etal*, "*shortest path and minimum delay algorithms in networks with time dependent edge- length"*, Journal of the Association for computing machinery, 37(3):607-625, 1990.

[5] Nikolova, E.; Brand, M.; and Karger,D. 2006. Optimal route planning under uncertainty. In Proceedings of International Conference on Automated Planning and Scheduling.

[6] Florian, M.; Mahut,M.; and Tremblay, N. 2008. Application of a simulation-based dynamic traffic assignment model. European Journal of Operational Research 189(3):1381–1392.

[7] Miller-Hooks, E. & Mahmassani, H., "Least expected time paths in stochastic, time-varying transportation networks", *Transportation Science, [Baltimore]: Transporation Science Section of ORSA, 1967-*, 2000, *34*, 198-215.

[8] Delling, D. & Wagner, D., "Time-Dependent Route Planning", *Robust and Online Large-Scale Optimization, Springer*, 2009, *5868*, 207-230.

[9] Bander, J. & White, C., "A heuristic search approach for a nonstationary stochastic shortest path problem with terminal cost", *Transportation Science*, 2002, *36*, 218 – 230.

[10] Chris Gaskett. "*Q-Learning for Robot Control*", Ph.D. dissertation, The Australian National University, 2002.

[11] Work, D.; Blandin, S.; Tossavainen, O.; Piccoli, B.; and Bayen, A., A traffic model for velocity data assimilation. Applied Mathematics Research eXpress, 2010.

[12] Kok, A.L., Hans, E.W. & Schutten, J.M.J, "Vehicle routing under time-dependent travel times: the impact of congestion avoidance, Operational Methods for Production and Logistics". University of Twente, 2010.

[13] Feldman, R. & Valdez-Flores, C., "Applied probability and stochastic processes", *Springer*, 2010.

[14] Michail G. Lagoudakis, Ronald Parr, and Michael L. Littman. "*Least-square methods in reinforcement learning for control"*, Shannon Laboratory, AT&T Labs – Research, Florham Park, NJ 07932, U.S.A, 2002.

[15] William D. Smart. "*Making Reinforcement Learning Work on Real Robots*," Ph.D. dissertation, Brown University, Providence, Rhode Island, United States, 2002.

[16] Sloman, A. and Logan, B. "*Evolvable Architectures For Human-Like Minds"*, Presented at 13th Toyota Conference on "Affective Minds", 2000.

[17] Ian D. Kelly. "*The Development of Shared Experience Learning in a Group of Mobile Robots*," University of Reading, Department of Cybernetics, 1997.

[18] Wolfram Schultz, Peter Dayan, and P. Read Montague. "A neural substrate of prediction and reward", Science, 275:1593-1598, 1997

[19] Crites R. H. "Large-Scale Dynamic Optimization Using Teams of Reinforcement Learning Agents", PhD thesis, University of Massachusetts, Amherst, MA, 1996.

[20] Crites R. H. and Barto A. G. "Improving elevator performance using reinforcement learning", In D. S. Touretzky, M. C. Mozer, M. E. H., editor, Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference, pages 1017-1023, Cambridge, MA. MIT Press.

[21] Schwartz A. "A reinforcement learning method for maximizing undiscounted rewards", In Proceedings of the Tenth International Conference on Machine Learning, pages 298-305. Morgan Kaufmann, 1993.

[22] Christopher J. C. H. Watkins. Learning from Delayed Rewards. Ph.D. thesis, King's College, Cambridge, UK, 1989.

[23] Boutilier, C.; Dearden, R. & Goldszmidt, M., "Stochastic dynamic programming with factored representations", *Artificial Intelligence, Elsevier,* 2000, *121*, 49-10