Traffic Light Control Using SARSA with Three State Representations

Thomas L. Thorpe
IBM Corporation
PO Box 1900
Boulder, CO 80301-9191
tlt@VNET.IBM.COM

Charles W. Anderson
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
anderson@cs.colostate.edu

Category: Control, Navigation, and Planning: reinforcement learning. No part of this work has been submitted or appeared in other scientific conferences.

Abstract

SARSA is applied to a simulated traffic light control problem and compared with a fixed-duration strategy and a simple, rule-based strategy. The performance of SARSA with three different representations of the current state of traffic is analyzed. SARSA is shown to be superior to the best, fixed-duration light timing and close to optimal for the simulation used in this article.

1 INTRODUCTION

A variety of traffic control strategies are being studied in real traffic networks and in simulation. The Denver Regional Council of Governments works with the Colorado Department of Transportation and citizens to identify and modify problem intersections (Garnaas, 1996). Computers are used to monitor the traffic flows for critical intersections through the Denver region. The computers have the capability to change traffic light timing remotely but are only used to collect data for traffic analysis. Recently a major traffic artery was re-timed from 90 seconds in the heavy traffic flow direction to 100 seconds. This resulted in an 87% reduction in times stopped at lights. Stockholm, Sweden, uses remote television cameras to monitor high traffic flow areas (Olsson, 1996). The traffic conditions are directly observed and speed limits and traffic light timing can be slowly adjusted remotely. Vehicles can also be rerouted remotely to reduce congestion. A neural network has been used as a traffic light controller in a simulator based on nine actual intersections in Manhattan, New York (Chin, 1995). The neural net used simultaneous perturbation stochastic approximation and was able to reduce the vehicle wait time by 10% relative to the existing control strategy.

Current traffic controllers in wide use are very primitive and require frequent manual adjustments to keep traffic flowing smoothly. In this article, we show that reinforcement learning with complete knowledge of vehicle locations approaches the best traffic light performance that can possibly be achieved for the limited traffic simulation we used (Thorpe, 1996). In Section 2, we describe the traffic simulation and several conventional traffic light controllers. Our application of SARSA to the traffic light control problem is described in Section 3. Our results are summarized in Section 4, and Section 5 presents our conclusions and describes on-going work.

2 TRAFFIC SIMULATOR AND CONTROL STRATEGIES

Our simulator (Thorpe, 1996) uses one-second, discrete time steps for the traffic light controller agent and the simulation of vehicle movement. Vehicle movement is simulated with realistic velocity and acceleration limits and is constrained by minimum following distances. The speed limits on each lane vary from 20 to 40 miles per hour and vehicles do not exceed these limits. We have simulated a 4 x 4 grid of east-west and north-south streets. The simulated distance between traffic lanes is 440 feet apart. Our simulator is not realistic in that it does not avoid oncoming traffic while making turns or passing through intersections. Otherwise, it is fairly realistic: vehicles are stopped at red lights, vehicles maintain safe following distances, and the physics of motion of the traveling vehicles are closely approximated.

Fifty cars are inserted in the network with their starting and destination intersections chosen randomly before the simulation is started. The routes are chosen using an Iterative Deepening A* algorithm. The route for each vehicle may require more than one turn because the under estimate used in route selection is based on the travel time through a lane of traffic. By using the same random seed for all testing, vehicles follow the same routes. This ensures the testing results between the reference and experimental implementations are directly comparable.

Three fixed, traffic light control strategies were tested. The *all green* strategy simply sets all lights to be green. This strategy is used as a baseline that other methods might asymptotically approach. In the real world this would be the equivalent of a four way stop and should be fairly easy to out-perform, but in our simulation vehicles do not stop at intersections to avoid other traffic. Therefore, the all green strategy results in the best performance. The *fixed duration* strategy cycles through green and red lights at predefined intervals. This is the normal type of traffic control used today. The duration can be varied but is usually based on time of day and must be preset manually. The timed light controllers should provide the upper bounds on traffic performance. The *greatest volume* strategy sets to green the traffic lights in the east-west or north-south lanes depending on which direction contains the most vehicles. The lights in the other direction are set to red.

3 TRAFFIC LIGHT CONTROL WITH SARSA

We applied the SARSA (Sutton, 1996) algorithm to the traffic light control problem using replace traces (Singh and Sutton, 1996) and a zero probability of taking non-greedy actions. The traffic controller is trained using experience at a single intersection with 4 lanes of traffic leading into it. A *lane*, relative to an intersection, is the block-long (440 foot) stretch of road approaching the intersection. The number of vehicles placed in the north-south and east-west lanes varies between zero and 50 and is chosen randomly. The vehicles' directions are also chosen randomly. After the vehicles are added to each lane, the vehicles' position and speed within each lane is randomly determined to the degree that all vehicles fit within their lane and are properly spaced to prevent collisions. In this way, a large variety of initial states are visited.

The current state is characterized for SARSA by the number and positions of vehicles in the north, south, east, and west lanes approaching the intersection. The action for a given state sets the color of the north-south traffic lights, and indirectly sets the color of the east-west lights which are always set to the opposite color of the north-south lights. These features were quantized and combined in three ways and the performance with the three resulting state representations was compared.

The first representation, called the vehicle *count* representation, is formed by summing the number of vehicles in the two north-south lanes and the vehicles in the two east-west lanes. These two sums are quantized into 10 partitions for values of 0, 1–5, 6–10, 11–15, ..., 36–40, and 40+. All pairs of partitions of the two sums are combined with the two possible colors of the north-south light to form 200 (10 x 10 x 2) discrete three-dimensional states. Thus, for this representation the input to the learning agent was a 200-component binary vector with a single nonzero component.

The second representation, called the *fixed distance* representation, retains an indication of the relative distance of vehicles from the intersection. The intersection's lanes are divided into 110-foot intervals, forming four partitions on each of the four lanes. The presence or absence of vehicles in each partition is determined and recorded by setting an "occupied" bit for each partition. The occupied bits for the first 110-foot partition of the north or south lanes are combined with a boolean "or" operation, as are the north and south occupied bits for the other three corresponding partition pairs. Thus, the input to the neural network with this representation is a 9-component vector with the first 8 components being the occupied bits and the 9th being a binary value indicating the color of the north-south lights.

The third representation, called the *variable distance* representation, partitions each lane like the fixed distance representation, but the divisions are at unequal distances from the intersection. Partition boundaries are at 50, 110, 220, and 440 feet from the intersection, making four partitions. The input to the learning agent with this representation consists of 9 components, just like the fixed distance representation.

Four performance measures were calculated for all tested strategies: the number of simulation steps required for all vehicles to reach their destinations; travel times for each vehicle; wait times for each vehicle; and the number of stops made by each vehicle. Tests were performed by duplicating the current state of the SARSA controller, trained on one intersection, at every intersection of a traffic network with

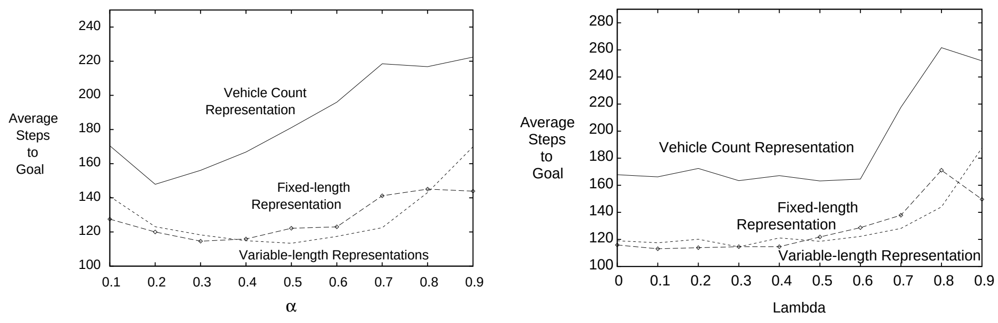


Figure 1: Average steps to goal versus α and λ for 50 cars in 4 x 4 test network.

4 x 4 intersections. The performance measures were calculated on this test network after every 20 learning trials.

A reward of $r = -1$ was used for each step of a trial. No discount was used, since the trials are finite in length. After some experimentation, the number of trials per run was chosen to be 4,000. The Q values were initialized to zero before the start of each run. For each state representation, the learning rate, α, was varied from 0.1 to 0.9 and the eligibility trace decay rate, λ, was varied from 0.0 to 0.9. Values greater than 1.0 caused floating point overflows. The 90 training runs for a single representation used 4,000 training trials and 240 testing trials per run with a maximum of 1,200 steps per testing trial. This took about 8 days of wall clock time on a 100MHz Pentium processor. Three different computers were used 24 hours a day to complete the training.

4 RESULTS

The fixed duration strategy was tested using 5, 10, 15, 20 and 25-second intervals. A duration of 10 seconds produces the best results when considering all performance measures. This strategy is included on subsequent plots for comparison.

The performance of the SARSA algorithm was tested with the three representations described above. To find the best values of α and λ for each representation, the number of steps required for all vehicles to reach their destinations in the 4 x 4 test network was averaged over all trials. Results are shown in Figure 1. This gives an indication of the noise in the system which is important in an actual implementation. The need to find the learning parameters that give the best results must be balanced by a policy that does not change drastically during on-line learning. The graphs show that the vehicle count representation is much noisier in comparison to the fixed distance or variable distance partitions. Performance with the fixed and variable distance representations are very similar. The best values of α are 0.3 to 0.5 and for λ are 0 to 0.6.

The remaining figures compare the best runs of the various strategies. The ordering of the control strategies from best to worst performance was found to be: 1) all green lights; 2) SARSA with the variable-distance representation; 3) SARSA with the fixed-distance representation; 4) greatest volume; 5) SARSA with the vehicle

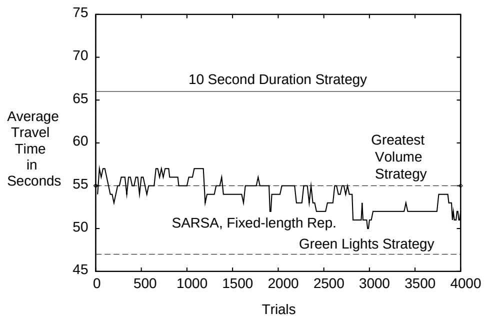


Figure 2: Average travel time for 50 cars in 4 x 4 test network.

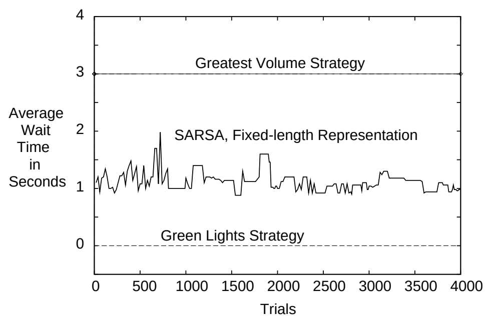
count representation; and 6) fixed duration. We did not expect the greatest volume strategy to do as well as it did. The limited size and duration of the simulation probably helped the greatest volume strategy perform so well. In an actual traffic network at rush hour, it would probably flip flop too fast causing an even greater traffic jam.

The best SARSA run evaluated by the total simulation steps required to clear all vehicles from the simulation used variable partitions, $\alpha = 0.4$, and $\lambda = 0.4$. When evaluated by the average travel time for all vehicles to exit the simulation, the best results were obtained with fixed partitions, $\alpha = 0.6$, and $\lambda = 0.1$. Results are plotted in Figure 2. SARSA started out worse than the greatest volume strategy and gradually improved over it.

The performance improvements using SARSA are much better when judged by the average wait time and total stops the vehicles experienced during the simulation. These results are shown in Figure 3. The average wait time for the greatest volume strategy and 10-second fixed duration strategy were the same at about 3 seconds. The average wait time for the all-green strategy was about 0.02 seconds and the best wait time for SARSA varied from 1 second to 2 seconds using fixed partitions, $\alpha = 0.6$, and $\lambda = 0.1$. The SARSA controller requiring the least number of stops during the simulation used variable partitions, $\alpha = 0.8$, and $\lambda = 0.2$. The SARSA controller was significantly better than the fixed-duration and the greatest volume strategies with one test result requiring no stops during the simulation. One stop was required for all green lights, 22 stops for the greatest volume strategy and 53 stops for the 10-second fixed duration strategy. The number of stops experienced by the SARSA controller ranged from zero to 26.

5 CONCLUSIONS AND FUTURE WORK

SARSA learned traffic control strategies that approached optimal performance. The best state representation for SARSA was the fixed or variable partitioning method. The representation based on just vehicle counts worked better than expected, but would probably not generalize well in an actual implementation. Small learning rates and small to moderate decay rates give smooth policy transitions during learning which is important to avoid costly traffic jams. The methods studied here, while they worked very well in this simulation, would probably not work well if actually implemented. Real world traffic controllers do not have complete knowledge of ve-



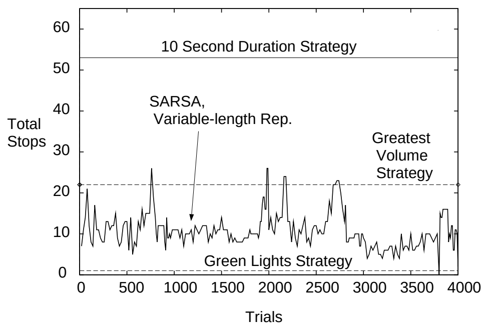


Figure 3: Average wait time and total stops for 50 cars in 4 x 4 test network.

hicle locations and do not terminate nicely as the simulator does. When computer vision is more economical, the method applied here could be used to determine the partitions that are occupied and, when incorporated with limits to prevent long or short light cycles, could be directly implemented.

The simulator implemented here dealt with fairly small traffic volumes. The real test of an agent would be to see if it could manage gridlock conditions effectively. Reinforcement learning techniques applied in other areas could be tested. Reinforcement learning techniques for network data flow could be applied to traffic light controllers to see how they perform. Auto traffic is somewhat similar to network traffic and data packets flowing through networks. The main difference is that data packets can continue to flow at varying rates, but are not required to physically stop as automobiles must do to avoid collisions. The traffic control problem is similar to elevator dispatching (Crites and Barto, 1996). The arrival of vehicles at an intersection and the direction of travel are stochastic in nature. The elevator controller could be modified for the traffic control problem to see how well it performs. If successful, this would be an implementation that could be refined and actually used in real traffic.

Since traffic flows continuously throughout the day, discounted rewards should be implemented so the controller can continue to learn while directing traffic over much longer simulations. The stochastic nature of the traffic flow needs to be considered by removing the knowledge of vehicle locations from the learning agent. Input from sensors that are installed at intersections could be used to predict lane densities and vehicle locations. A linear programming problem could be set up to estimate the number of vehicles in lanes by reducing the error of several simultaneous equations that account for vehicle turns and realizing that the number of vehicles entering an intersection must also exit the intersection.

REFERENCES

Chin, D.C., Smith, R.H., Spall, J.C. (1995) A system-wide approach to adaptive traffic control. In *Proceedings of 3rd Symposium on Research & Development*, Johns Hopkins University, November, 1995. Abstract available at http://www.jhuapl.edu/symposium/3rd_RandD/Traffic.htm.

Crites, Robert H., Barto Andrew G. (1996) Improving elevator performance using reinforcement learning. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA: MIT Press.

Garnaas, Steve, May 10, 1996, Denver Post, "120th is giving commuters the green light", p 1B and "Caught in Traffic, Timing the thing for Colorado's signal supervisor", p 2B.

Olsson, G. (1996) Fewer traffic jams thanks to computers. At http://www.stockholm.se/bm/projects/vagtrafik-en.html.

Singh, S.P. and Sutton, R.S. (to appear) Reinforcement learning with replacing eligibility traces. *Machine Learning*.

Sutton, R.S. (1996) Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA: MIT Press.

Thorpe, T. (1996) A physically-realistic simulation of vehicle traffic flow. Technical Report 96-118, Department of Computer Science, Colorado State University.