**Abstract**

We demonstrate a new approach on optimizing traffic light control systems with reinforcement learning while comparing Trust Region Policy Optimization (TRPO) and Deep Q Network (DQN) reinforcement-learning algorithms. We compare the results with some recent similar studies like IntelliLight as well.

Keywords - TRPO, DQN, Reinforcement Learning, SUMO, GYM.

**Introduction**

Traffic congestion has become increasingly costly. For example, traffic congestion costs USA $124 billion a year, according to a report by Forbes in 2014.

Existing traffic lights are mostly operated by hand-  crafted rules. But this method does not work well since it is not using live traffic data. So, The aim of the project is to optimize traffic by controlling traffic lights with an intellegent agent which can consider current traffic data.  For this, we have used some modern reinforcement learning algorithms which are model-free and can work with contionues enviorenments 

We have decided to compare TRPO(Trust Region Policy Optimization) and DQN(Deep Q Network) algorithms on this problem.

**Related Work**

We started our work with checking similar researches made on development of intelligent traffic light systems for Traffic light optimization problem. Most of the studies on this problems suggest solutions based on reinforcement learning. For instance on the IntelliLight research project researchers uses DQN reinforcement learning algorithm to solve same problem. After research we decided on focusing on reinforcement learning algorithms.

After checking similar studies on reinforcement learning, we considered the environment, state, and reward, and decided the action of agent. Environment is custom created intersections on SUMO. We created custom car routines for environments as well. State is average queue length of cars on roads, number of vehicles on every road, average waiting time of cars, current phase of the traffic lights, the duration of phase and the image data of the intersection. We used convolution of the current image of the intersection as the image data. We weren’t sure about the contribution of the convolution on the state but we added because we show similar inputs in most of the recent studies. For reward we used minus of the queue length, the speed of the cars and the number of cars which is breaking. Action of the agents are decided as change of traffic light phases. If we have one intersection with four roads, we generally have two phases which are green light for vertical roads, red light for horizontal roads, and red light for vertical roads, green light for horizontal roads. If we have 0 as the action, we are not changing the current phase while if we have 0 as the action, we are changin 

**Methodology**

SUMO ve traci ayarlandı.  

Örnek yollar ve rutinler oluşturuldu.   
TRPO ile çalışan bir agent üretildi.    
DQN ile çalışan bir agent üretildi.   
GYM Ayarlandı.   
GYM ile SUMO gym-sumo custom environmenti oluşturularak bağlandı.    
GYM ye agentlar eklendi.   
GYM üzerinden algorithm çalıştırıldı ve sonuçlar izlendi

As enviorenment SUMO(Simulation Of Urban Mobilization) chosen. It can simulate varios traffic scenarios with various parameter options.

To get information from SUMO enviorenment we have used Traci module of SUMO. It can provide information for every step such as queue length of cars, speed and acceleration of cars etc.

To connect algorithms to the enviorenment we have used Gym to simpilize the process. Gym is a toolkit for reinforcement algorithms.

We have created a custom enviorenment in Gym which is SUMO. We have configured state, reward and actions. Then, connected this custom enviorenment to TRPO algorithm which can work with Gym toolkit.

After that, we have created an experiment in SUMO and used this experiment to compare DQN and TRPO.

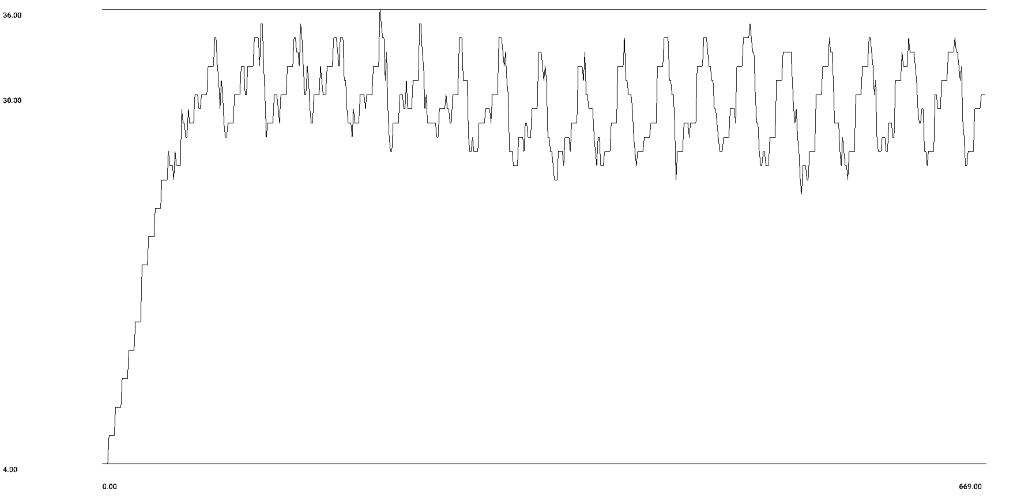
**Result and discussion**   
TRPO da DQN de fena değil

Both algorithms showed similar results. But we didn’t have time to tune parameters for TRPO. Thus, we believe that if we could tune the parameters of TRPO it would have shown better results.

TRPO yavaş ama kararlı ilerliyor optimal duruma geçiyor   
DQN daha hızlı optimize olsa da anormallikler oluşturabiliyor. Gerçek dünya da kullanmak için daha risk oluşturuyor.   
TRPOnun hyper parametreleri yeterince ayarlı değil. Ayarlanırsa daha iyi sonuç oluşturacağı düşünülmekte.

Below you can find the car count – time graph. Car count is minus reward so the lesser the better.

**DQN**



**TRPO**

