Hierarchal Clusters Based Traffic Control System

FadyTaher
Dept. of Computer Science and Eng., Faculty of Elect., Eng., Menoufia University

Ayman EL-Sayed
Dept. of Computer Science and Eng., Faculty of Elect., Eng., Menoufia University

Ahmed Shouman
Dept. of Computer Science and Eng., Faculty of Elect., Eng., Menoufia University

Ahmed El-Mahalawy
Dept. of Computer Science and Eng., Faculty of Elect., Eng., Menoufia University

Abstract

Traffic jam is a crucial issue affecting cities around the world. They are only getting worse as the population and number of vehicles continues to increase significantly. Traffic signal controllers are considered as the most important mechanism to control the traffic, specifically at intersections, the field of Machine Learning offers more advanced techniques which can be applied to provide more flexibility and make the controllers more adaptive to the traffic state. Efficient and adaptive traffic controllers can be designed using a multi-agent reinforcement learning approach, in which, each controller is considered as an agent and is responsible for controlling traffic lights around a single junction. A major problem of reinforcement learning approach is the need for coordination between agents and exponential growth in the state-action space. This paper proposes using machine learning clustering algorithm, namely, hierarchal clustering, in order to divide the targeted network into smaller sub-networks, using real traffic data of 65 intersection of the city of Ottawa to build our simulations, the paper shows that applying the proposed methodology helped solving the curse of dimensionality problem and improved the overall network performance.

Keyword: Adaptive traffic signal control, Clustering, Multi-agent system, Reinforcement learning, Simulation, Traffic controller.

1. Introduction

A signal controller is a device which controls traffic flow at a specific intersection. In real life situation, most traffic light controllers work with far less information about the traffic, they follow a protocol, that is, the light is red for some time and green for some subsequent time. The time intervals usually change during rush hours but are still static [1].

Intelligent traffic systems researches aim to increase the efficiency of traffic networks. Researchers have been trying to implement intelligent systems as a replacement for static ones, such intelligent systems use different machine learning algorithms to enable signal controllers to adapt and behave based on the traffic state in the network, this comes with a cost; as deploying such adaptive controllers at intersections without proper planning could lead to limit their potential benefits moreover, it might decrease the overall performance of the network. Therefore, optimally controlling and coordinating the operation of multiple signal controllers simultaneously is required. However, this integration adds some complexity to the system.

Reinforcement learning is a machine learning algorithm, the agent learns to map states to actions to maximize a numerical reward, through taking actions and analyzing the reward, the agent must discover which actions leads to the best reward in a specific situation, learned actions not only affect the immediate situation but also the next ones [3].

Two major challenges associated with implementing intelligent controllers using reinforcement learning are, the need for coordination and the curse of dimensionality [2]. To address these limitations, we present a new method which uses clustering algorithm to divide the network into strongly connected sub-networks using traffic between the intersections as a measurement.

Many works of intelligent signal control have used unrealistic evaluation, usually investigate performance on networks of 1–4 intersections with simple structures (e.g., one-way roads). While investigating such control system and simple networks is a good starting point, more realistic scenarios must be used to investigate if intelligent controllers can be applied in real world. The work presented in this paper makes several important contributions to intelligent signal control research. First, a methodology for modeling and creation of realistic traffic scenarios is developed. It is used to create a traffic simulation based on a 9x7 section of the downtown core of Ottawa, Canada.

The simulation also uses Realistic traffic data compute routes and expected vehicle entry/exit locations and turning rates.

We also use machine learning algorithms instead of traditional geographical attributes like distances between intersections or dividing the network based on the city's different territories or segments.

This paper gives a literature review about clustering and reinforcement learning in section 2. Section 3 explores contributions and similar work of other researchers in clustering and coordinating between controllers. Section 4 illustrates the proposed algorithm. Results are presented with SUMO simulator [10] using real traffic data acquired from the city of Ottawa and the results are shown in section 5. Finally, section 6 presents conclusion and future work.

2. Overview

2.1 Clustering

Clustering is categorization of items. The clustering problem has been a focus of study in many field by researchers; this reflects its importance and usefulness in exploratory data analysis. Clustering is used in data-analysis, decision-making, and machine-learning like document retrieval, and image segmentation.

In this paper we will be focusing on and using hierarchical clustering technique which can be addressed by two algorithms, single link and complete link. Difference between algorithms is addressed in the way they characterize similarity.

In single linkage method, the distance between two clusters is determined by those two elements (one in each cluster) that are closest to each other. Figure 1 depicts an illustrative image for demonstration.

$$D(X,Y) = \min_{x\in X, y\in Y} d(x,y) \tag{1}$$

In complete link algorithm Eq2 is used, the distance between two clusters is determined by those two elements (one in each cluster) that are farthest from each other. In either case, two clusters are merged to form a larger cluster based on distance criteria, figure 2 depicts an illustrative image for complete linkage demonstration.

$$D(X,Y) = \max_{x\in X, y\in Y} d(x,y) \tag{2}$$

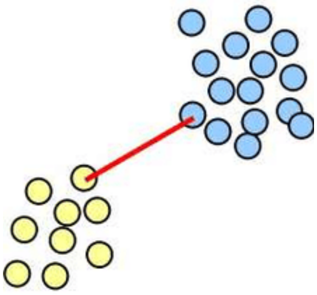


Figure 1Single linkage

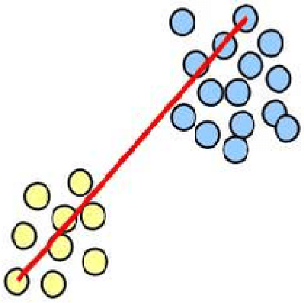


Figure 2 Complete linkage

The process of a hierarchical clustering is explained using the two-dimensional data set in Figure 3, which shows seven patterns labeled A, B, C, D, E, F, and G in three clusters. The algorithm yields a dendrogram representing grouping of patterns and levels at which clusters change, it is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering. A dendrogram corresponding to the seven points is shown in Figure 4. The dendrogram can be broken at different levels to yield different clusters of the data[14].

Typical pattern clustering activity involves the following steps [13]:

1 Feature selection, which identifies the most effective features during clustering process.

2 Pattern proximity, measured by using similarity distance function like Euclidean distance between groups.

3 The grouping step, in which objects are assigned to a specific cluster or assigned a degree of membership to a cluster.

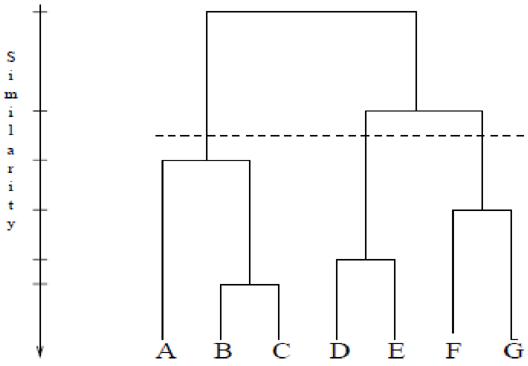


Figure 3 Clustering dendogram.

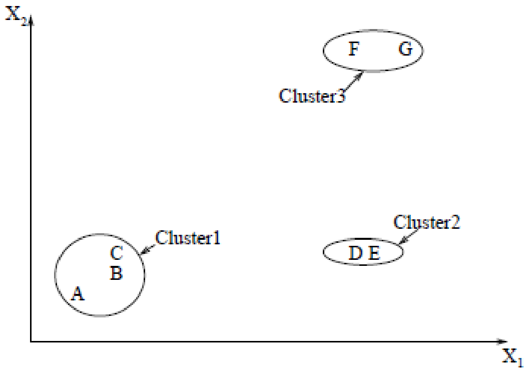


Figure 4 Points falling in three clusters

2.2 Reinforcement learning

Learning is a many-faceted phenomenon; it's the process of acquiring new knowledge, the development of motor and cognitive skill through instruction or practice, the organization of new knowledge into general, effective representations, and the discovery of new facts and theories through observation and experimentation. [15]

Researchers have been trying to implement these capabilities into computer. Solving this problem remains one of the most challenging goals in artificial intelligence.

The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. Reinforcement learning is type of machine learning problems which addresses the question of how an autonomous agent can learn to choose optimal actions to achieve its goals. Each time the agent performs an action in its environment, a trainer may provide a reward or penalty to indicate the benefit of the resulting state. For example, when training an agent to play a game the trainer might provide a positive reward when the game is won, negative reward when it is lost, and zero reward in all other states. [15]

Reinforcement Learning focuses on studying agents which interacts with environment to maximize a reward. Markov decision process (MDP) is used to model the environment, in which we assume the environment state only changes depending on agent's actions. The most common single-agent RL algorithm is Q-learning

The Q-learning agent learns optimal mapping between the environment's state (s) and the corresponding optimal control action (a) based on accumulating rewards $r(s,a)$ .Each state–action pair (s,a) has a value called the Q-factor that represents the expected long-run cumulative reward. In each iteration, i.e., k, the agent observes current state s and chooses and executes action a that belongs to the available set of actions A; then, the Q-factor is updated according to the immediate reward transition to state ś as follows [22]:

(See (3) at the end of page)

where α and γ ò$[0,\ 1]$ are referred to as the learning rate and the discount rate, respectively. The agent can simply choose the greedy action at each iteration based on the stored Q-factors, as follows:

$$a^{k+1} \in \arg\max_{a\in A}\left[Q(s,a)\right] \tag{4}$$

However, sequence Q^K is proven to converge to the optimal value only if the agent visits the state–action pair for an infinite number of iterations [21]. This means that the agent must sometimes explore (try random actions) rather than exploit the best-known actions. To balance the exploration and exploitation in Q-learning, algorithms such as ϵ-greedy and softmax are typically used [22].

3. Related Work

Intelligent agents interact in a cooperative environment where they learn by sharing information and trial and error.

A major problem of reinforcement learning approach is the need for coordination and exponential growth in the state-action space. This problem was investigated by Ming Tan [12], he showed that sharing learned policies among agents, speeds up learning at the cost of communication and for joint tasks agents engaging in partnership can significantly outperform independent agents at the cost of learning slowly.

Whitehead, Steven D [16] also showed that when used to solve multi-stage decision problems, reinforcement learning algorithms perform a kind of online, incremental search in order to find an optimal decision policy. The time complexity of this search strongly depends upon the size and structure of the state space or when the system must adapt to a change in the environment, search can be excessive. An analysis of the search time complexity of systems indicates that for a restricted, but representative set of tasks, the search time scales at least exponentially in the depth of the state space.

Over the past years researchers have been trying to change the way traffic controllers are operated, from using simple static controllers to adaptive controllers like actuated controllers [4] to using Machine Learning (ML) algorithms and more intelligent techniques, following we mention a few

$$Q^k(s^k,a^k) = (1-\alpha)Q^{k-1}(s^k,a^k) + \alpha\left[r(s^k,a^k) + \gamma \max_{a^{k+1}\in A} Q(s^{k-1},a^{k-1})\right] \tag{3}$$

1. Vital, Allan Saldanha, et al, they developed an intelligent agent able to perceive the environment through sensors and act on the environment autonomously and communicate with other agents aimed at collaborative decisions and actions into a region's traffic. Each agent has been appointed as an intelligent traffic light (LCA) and operations control center were appointed as region control agents (RCA). Where the scope of LCA's is the street and RCA is the set of traffic lights that have been defined as a region (e.g. city or neighborhood). It did not achieve the expected performance; the prototype demonstrates the feasibility to create intelligent traffic control capable of working in some social way to achieve a common goal [5].

2. Almejalli, Khaled, K.Dahal, and A.Hossain presented an intelligent decision support system based on the multi-agent approach. In that paper, an intelligent traffic control system (ITCS). In the proposed system, the network is divided into sub-networks, each of which has its own associated agent. The coordination between those agents was achieved through a high level agent called a coordinator, they also suggested that an investigation of an optimal way to split the network shall be carried out. The obtained results show its ability to identify the optimal global control action. [6]

3. S.El-Tantawy, B.Abdulhai, and H.Abdelgawadpaper presented the development and evaluation of a novel system of multi agent reinforcement learning for integrated network of adaptive traffic signal controllers. It offered two possible modes, independent mode, where each intersection controller works independently of other agents; and integrated mode, where each controller coordinates signal control actions with neighboring intersections. The major challenges were the need for coordination and the curse of dimensionality. In this system, each agent plays a game with its immediate neighbors. [2]

4. MA. Wiering presented a set of multi-agent model-based RL systems for traffic light control which can also be used for optimizing driving policies for cars. Experimental results showed that the RL systems can outperform a number of non-adaptable systems, it used global communication between traffic lights and it was noted that, for low traffic loads, constructing good (near-optimal) fixed controllers is not difficult, since all traffic noes can operate locally. Therefore, the gain in using RL for learning traffic light controllers is quite small, although learning driving policies is still useful. When they increased traffic load, the amount of interaction between traffic nodes increases and the locally well performing fixed systems, don’t work well anymore. [7]

5. P.G. Balaji X. German and D. Srinivasan presented a distributed multi-agent-based traffic signal control for utilizing green timing in a network to reduce the total travel. The proposed architecture uses traffic data collected by sensors at each intersection, stored historical traffic patterns and data communicated from agents in adjacent intersections to compute green time for a phase. The parameters values used in computing the green time is fine-tuned by online reinforcement learning with an objective to reduce overall delay. Simulation tests conducted on a virtual traffic network of Central Business District in Singapore for four different traffic scenarios showed almost 15% improvement. [17]

6. Dave McKenney and Tony White proposed algorithms capable of controlling traffic signals that rely on traffic observations made by available sensor devices and local communication between traffic lights. This allows traffic signals to be updated to better suit current traffic demand, while also allowing large problem sizes to be addressed. To have the system evaluated, a realistic traffic model was built using information supplied by the City of Ottawa, Canada. It was found, through simulation within the SUMO traffic simulation environment, that the proposed adaptive system resulted in higher overall network performance when compared to the current fixed signal plan controllers. [9]

7. SS Mousavi, M Schukat applied deep reinforcement learning algorithms with focusing on both policy and value-function based methods to traffic signal control problem in order to find optimal control policies of signaling, just by using raw visual input data of the traffic simulator snapshots. The approach have led to promising results and showed they could find more stable control policies compared to previous work of using deep reinforcement learning in traffic light optimization. [20]

8. Yilun Lin, Xingyuan Dai, et al also proposed DRL (Deep Reinforcement Learning) dedicated to large-scale UTC (Urban Traffic Control)problems to learn the relationship between the states and the actions. They test different reward functions and design a hybrid reward, in which the throughput of the traffic network, along with the balance of queueing length around intersections is chosen as the performance indexes, Tests show that this new model could be optimized within an acceptable time for a traffic grid. Compared with previous DRL models which take thousands of episodes to converge, their method takes only less than 50 episodes to converge for a more complex environment.

4. The proposed algorithm

We propose to cluster the network to smaller and coordinated sub-networks; actuated controllers [19] shall be implemented at any singleton cluster. Achieving that, shall remove all unnecessary communications between network agents, reduce learning time and improve the overall network performance. Figures 5, 6 present the difference between the currently used model and the proposed model.

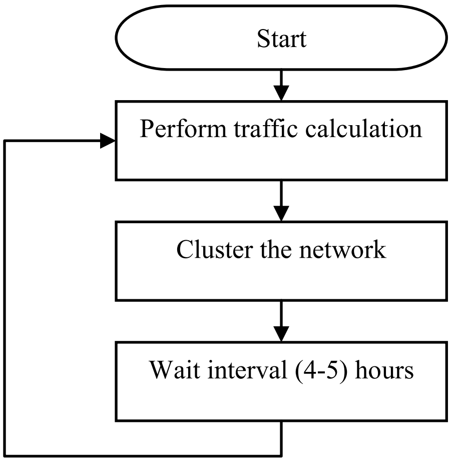


Figure 5 Proposed model

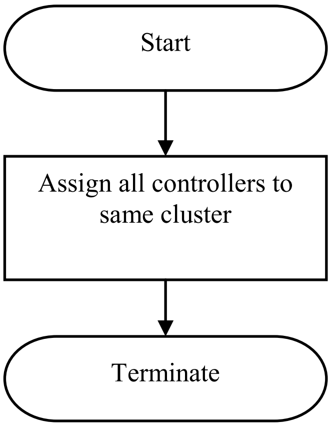
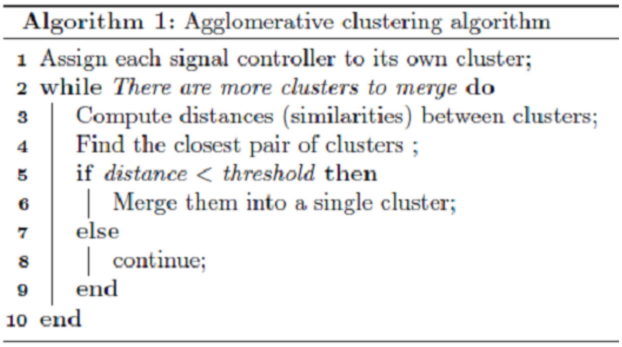


Figure 6 Current model

After performing 8 simulations runs we could obtain traffic counts between junctions, those counts were then used to construct a distance matrix to find how close each junction to others using averaged traffic counts as metric, figure 7show illustrative heat map image, junctions names are listed along X-Axis and Y-Axis and describes how well connected are these junctions, darker dots mean more traffic between junctions, the network was then clustered into group of junctions. Algorithm 1 describes how to cluster the network using a distance matrix.

In agglomerative approach, each object starts as a singleton and successively merges with other objects until a stopping criteria is met, on the other hand, divisive approach begins with all items belonging to a single cluster and keeps splitting until a stopping criteria is satisfied.



Algorithm 1: Agglomerative clustering algorithm

```
Assign each signal controller to its own cluster;
while There are more clusters to merge do
    Compute distances (similarities) between clusters;
    Find the closest pair of clusters ;
    if distance < threshold then
        Merge them into a single cluster;
    else
        continue;
    end
end
```

5. Performance Evaluation

5.1 Simulation setup

To demonstrate the effectiveness of an intelligent traffic control system that system should be tested on realistic traffic scenarios. For this reason, a realistic traffic model based on a section of the downtown area of the City of Ottawa was developed for use within the SUMO [10] microscopic traffic simulation environment. The open source SUMO simulation environment was chosen for a number of reasons including portability, presence of an active development community and availability of a graphical user interface.

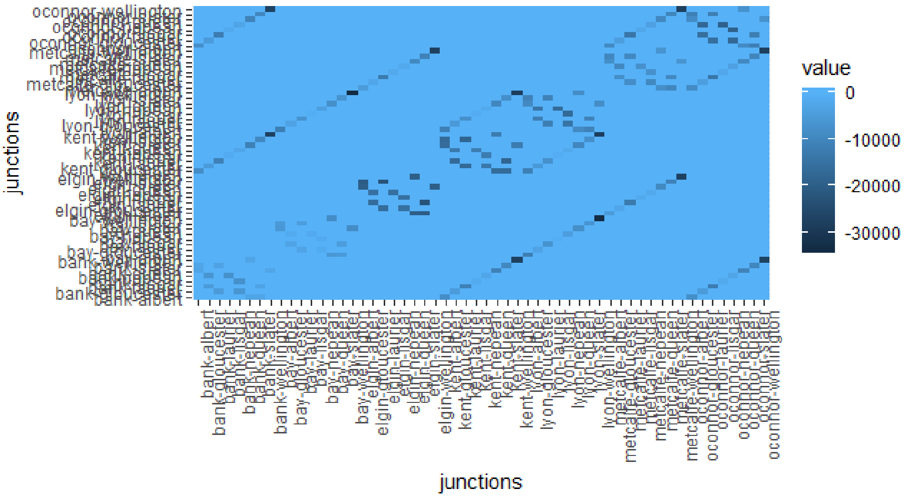


Figure 7 Distance matrix

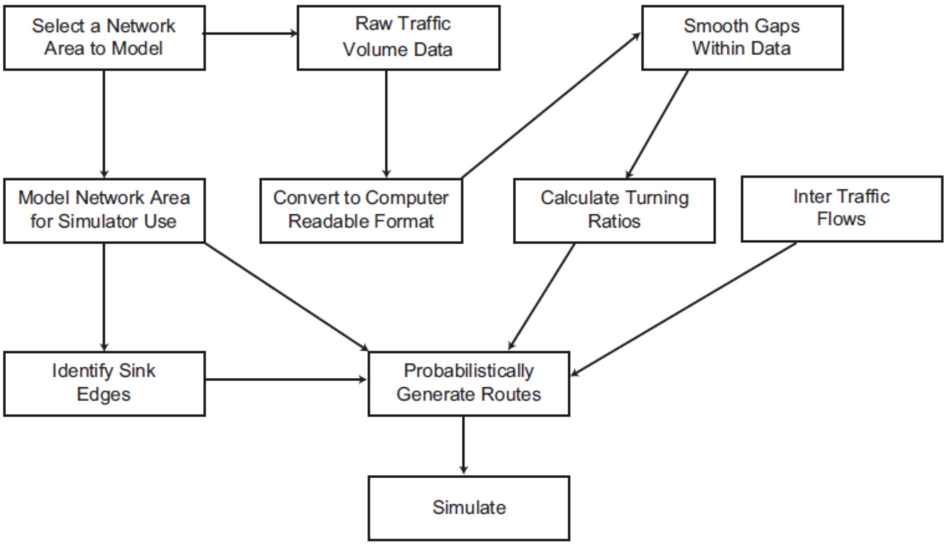

Select a Network Area to Model
Raw Traffic Volume Data
Smooth Gaps Within Data
Model Network Area for Simulator Use
Convert to Computer Readable Format
Calculate Turning Ratios
Inter Traffic Flows
Identify Sink Edges
Probabilistically Generate Routes
Simulate

Figure 8 Flowchart showing the traffic model creation process

Furthermore, SUMO has been used by other intelligent systems researchers, such as Garcia-Nieto et al. [23] and Passos and Rossetti [24].

The remainder of this section details the key steps in the model creation process which is outlined in figure 5. The area addressed is a 9x7 block of down-town Ottawa, with over 50 intersections requiring control. This is not an extremely large network but, it is much larger than most of the simple networks used in previous intelligent traffic signal research. This area also contains a wide range of street types ranging from small one-way residential streets with low traffic volumes, to main streets of high volume and multiple lanes.

After selecting an area to model, the road network for that area had to be captured and converted for use within the SUMO traffic simulator.

Initially, OpenStreetMap [18] was used to export an area of the city, shown in figure 9, which could then be imported directly into SUMO, shown in figure 11, (which offers a tool for importing OSM networks). While this process allowed for a quick generation of a network, it also included unnecessary parts (such as roads extending far beyond the considered area) which had to be removed. Also, a few lanes had to be manually added to the network, as the OpenStreetMap import failed to capture them accurately.

Figures 9, 10, 11 depicts the network as seen in OpenStreetMap and in JOSM editor after removing unnecessary parts, then finally then network after importing in the simulator.

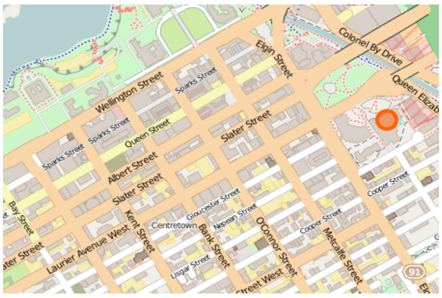


Figure 9 Downtown in open street map

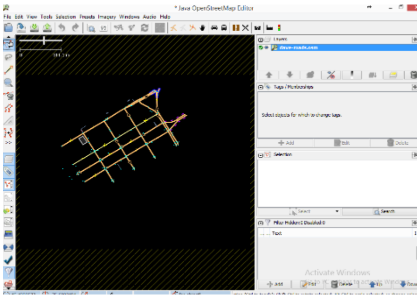


Figure 10 Area imported in JOSM editor

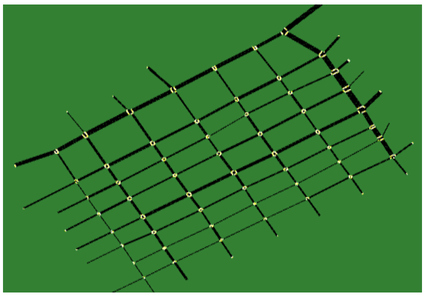


Figure 11 Network in SUMO simulation

$$Volume\_interval_i = round\left(Last\_volume + i * Delate\_volume\right) \qquad (5)$$

To build a traffic model which simulates a possible realistic scenario, traffic count and required data for each intersection was acquired from the City of Ottawa. An example of the supplied data is included in Table 1, showing the number of vehicles heading right (RT),left (LT) and straight (ST) over intervals throughout the day.

The data within Table 1 has gaps from 10:00 to 11:30 a.m. And 1:30 to 3:00 p.m. To fill these gaps [9], a simple linear interpolation function was used, adding 30 min intervals and allowing for a smooth transition from one volume to the next. This was achieved using Eq. (5), where last_volumerepresents the vehicle volume at the end of the last known interval and Delta_volume is the required linear change in volume to match the next known volume, after the addition of the specific number of 30 min intervals

The SUMO route generator used here relies on vehicle flows, which determine the number of cars to introduce/remove from a lane orroad segment over each interval, and turning rates, which determine the which vehicle to proceed in which direction. Using the data provided for each intersection (see Table 1 for an example), it is easy to calculate the turning ratios for the exit point of any roadiusing Eq. (6).

All traffic flows for a road segment can be calculated by subtracting the total number of cars entering the road from the cars exiting the same road over a time interval. With this data calculated, the SUMO routing program generates 15 sets of car routes to be used in the simulation. This program generates sets of car routes by using the specified traffic flows to determine where/ when cars should be allowed into the network (which begins a route) and using the turning ratios at the end of each road segment to determine which way the car should proceed. When a car's route enters a `sink' road, the route is terminated.

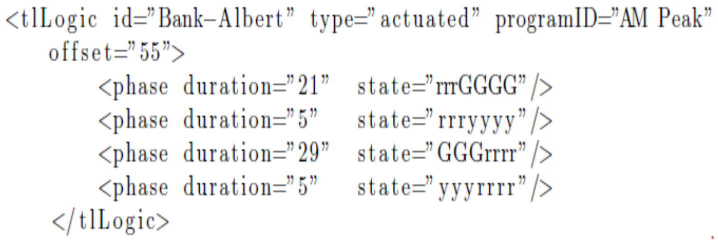
$$Ratio_i = \frac{Count_i}{Total} \qquad (5)$$

Vehicle’s route files were acquired from work presented in D. McKenney and T.White [9] and was used to generate traffic during simulation.

Figure 12 depicts a signal timing plan sample for intersection bank-albert as acquired from Ottawa traffic office. Figure 13 depicts xml snippet for the intersection after conversion in order to be used in the simulation. It shows how to determine traffic controller operation mode, the “type” attribute has two options, “static” or “actuated”, setting the type to “static” makes the intersection follow the coordinated timing plan while "actuated" makes it adaptive to traffic. Operation mode was set to static to follow optimized city plan and provide us with the required type of coordination in a way that produces green waves [11].

	AM Peak	Off Peak	PM Peak	Night	Weekend
	1	2	3	4	5
Cycle	60	55	55	55	55
Offset	55	40	8	26	40
NS Thru	26	28	25	28	28
WB Thru	34	27	30	27	27

Figure 12 Bank-Albert signal timing



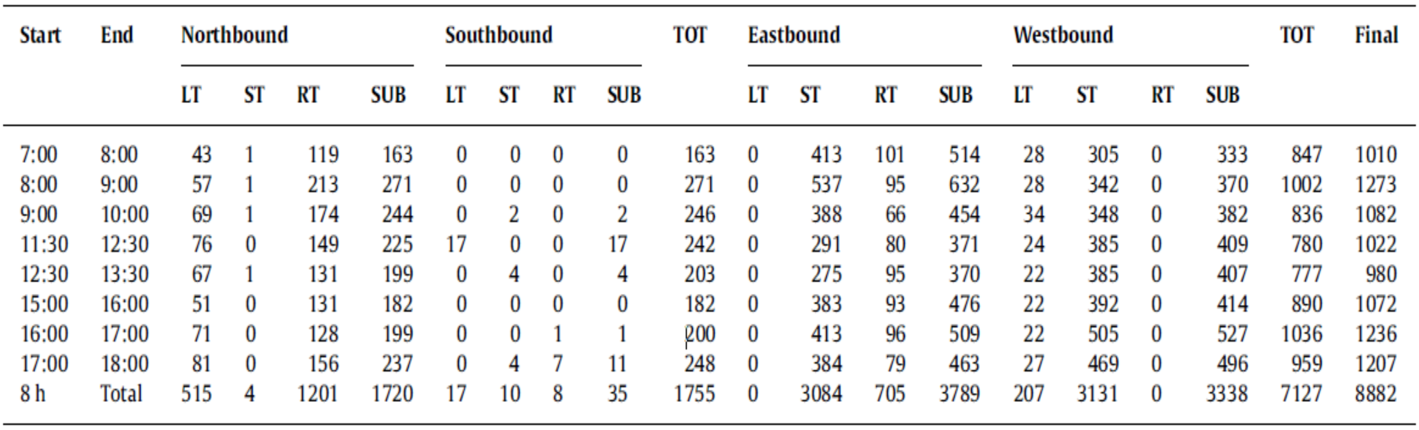
```
<tlLogic id="Bank-Albert" type="actuated" programID="AM Peak"
   offset="55">
        <phase duration="21"  state="rrrGGGG"/>
        <phase duration="5"   state="rrryyyy"/>
        <phase duration="29"  state="GGGrrrr"/>
        <phase duration="5"   state="yyyrrrr"/>
   </tlLogic>
```

Figure 13 Bank-Albert signal timing plan in simulator

5.2 Evaluation Metrics

Two values were considered to measure the network performance: 1) Average number of departed cars that is the total number of cars which have reached their destination, 2) Mean Travel Time that is the average travelling time for all cars since entering the network until reaching their destination. 3) Expected learning time.

Table 1 Example of the available traffic volume data for each intersection



Start	End	Northbound				Southbound				TOT	Eastbound				Westbound				TOT	Final
		LT	ST	RT	SUB	LT	ST	RT	SUB		LT	ST	RT	SUB	LT	ST	RT	SUB		
7:00	8:00	43	1	119	163	0	0	0	0	163	0	413	101	514	28	305	0	333	847	1010
8:00	9:00	57	1	213	271	0	0	0	0	271	0	537	95	632	28	342	0	370	1002	1273
9:00	10:00	69	1	174	244	0	2	0	2	246	0	388	66	454	34	348	0	382	836	1082
11:30	12:30	76	0	149	225	17	0	0	17	242	0	291	80	371	24	385	0	409	780	1022
12:30	13:30	67	1	131	199	0	4	0	4	203	0	275	95	370	22	385	0	407	777	980
15:00	16:00	51	0	131	182	0	0	0	0	182	0	383	93	476	22	392	0	414	890	1072
16:00	17:00	71	0	128	199	0	0	1	1	200	0	413	96	509	22	505	0	527	1036	1236
17:00	18:00	81	0	156	237	0	4	7	11	248	0	384	79	463	27	469	0	496	959	1207
8 h	Total	515	4	1201	1720	17	10	8	35	1755	0	3084	705	3789	207	3131	0	3338	7127	8882

5.3 Results

To test the capability of the proposed algorithm to effectively improve the traffic, the performance of the system using the proposed method was compared to the performance of the network using the signal timing plans provided by the City of Ottawa. Using SUMO's route generation program [10], 8 scenarios were generated from the available data, each of which represents realistic vehicle volumes for an 11-hour period (7a.m.–6p.m.). Each scenario was simulated once with the network being clustered using the proposed algorithm and once without clustering. Results presented in this section are averaged over the 8 simulation runs.

In the experimental simulation of the proposed methodology, mean travelling time i.e.; average time taken from source to destination and mean cars throughput i.e. number of cars which reached their destination, are the two measures used to evaluate the improvement.

The average traffic count calculated between intersections was used as a threshold to perform the clustering algorithm, resulting in the clusters dendrogram, for demonstration of the dendogram, figures 14, 15 depicts each cluster (junctions belonging to same group) are bounded by a green rectangle using single link and complete link algorithms respectively. The clustering pattern for complete linkage distance tends to create compact clusters, also complete linkage dendogram shows that, intersections are grouped in a cluster of smaller size (1-2).On the other hand, single linkage tends to add one point at a time to the cluster, creating long stringy clusters. Single linkage dendrogram shows that, most intersections are grouped in one cluster, and a few are clustered as singleton junctions.

After performing the clustering, all singleton intersections signal timing configuration were removed from the coordinated network setup by changing their signal timing operation mode to be "actuated" instead of "static" as previously explained in figure 10.

In figure 16 we can see the network junctions outlined on X and Y axes before clustering, all intersections are considered as one huge network, or one cluster and each junction is considered as an agent which will communicate with all other agents in the network and such huge network will increase the RL state-action space dramatically and increase learning time as a result.

In figure 17, 18 the network is shown after clustering process has been performed using single link and complete link algorithms respectively, intersections which belong to the same cluster have the same symbol for demonstration. We can see that, the complete-link algorithm produces tightly bound or compact network (1-2 junctions per cluster), the single-link algorithm, by contrast, suffers from a chaining effect. It tends to produce clusters that are straggly or elongated (more than 10 junctions for a cluster). However, from a pragmatic viewpoint, it was observed that, the complete link algorithms are more useful in producing hierarchies in many applications than the single-link algorithm [13].

Referring to Whitehead, Steven D [16] and Min Tan [12], decreasing the number of cooperative agents shall decrease the learning time. This is a first of step of overcoming the "curse of dimensionality" problem, other approaches include applying function approximation methods to reduce and/or interpolate the searchable value-space, and it tries to generalize a value function.

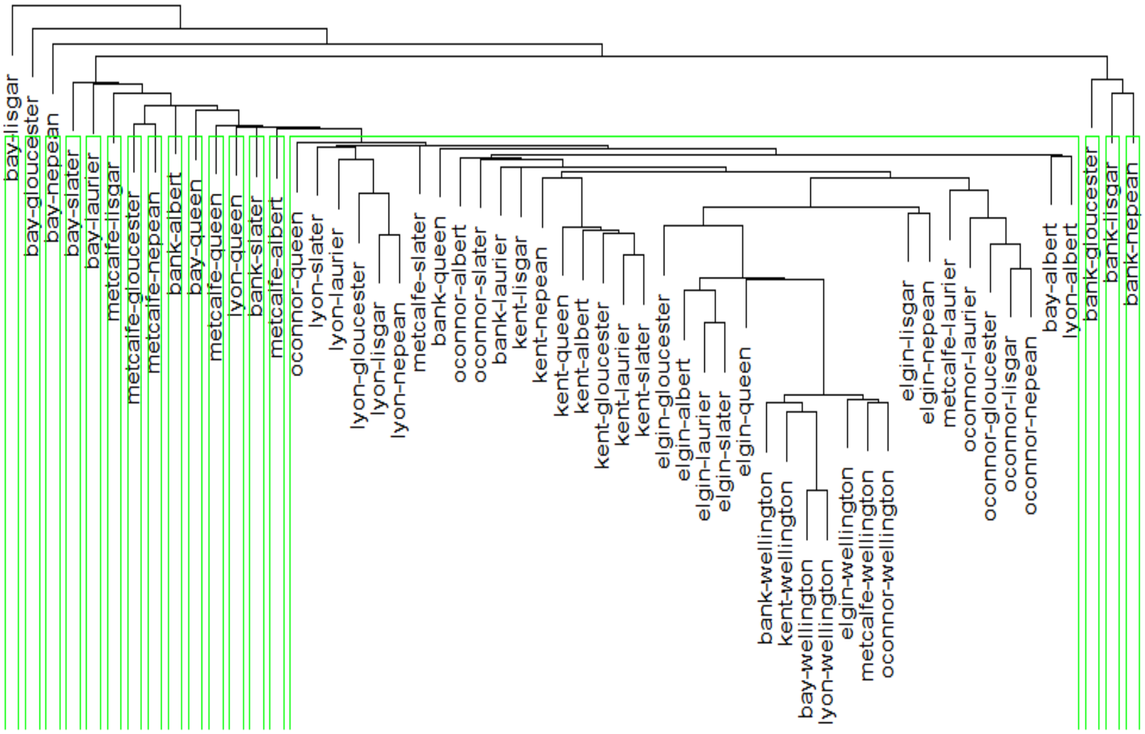


Figure 14 Single link clustering dendrogram

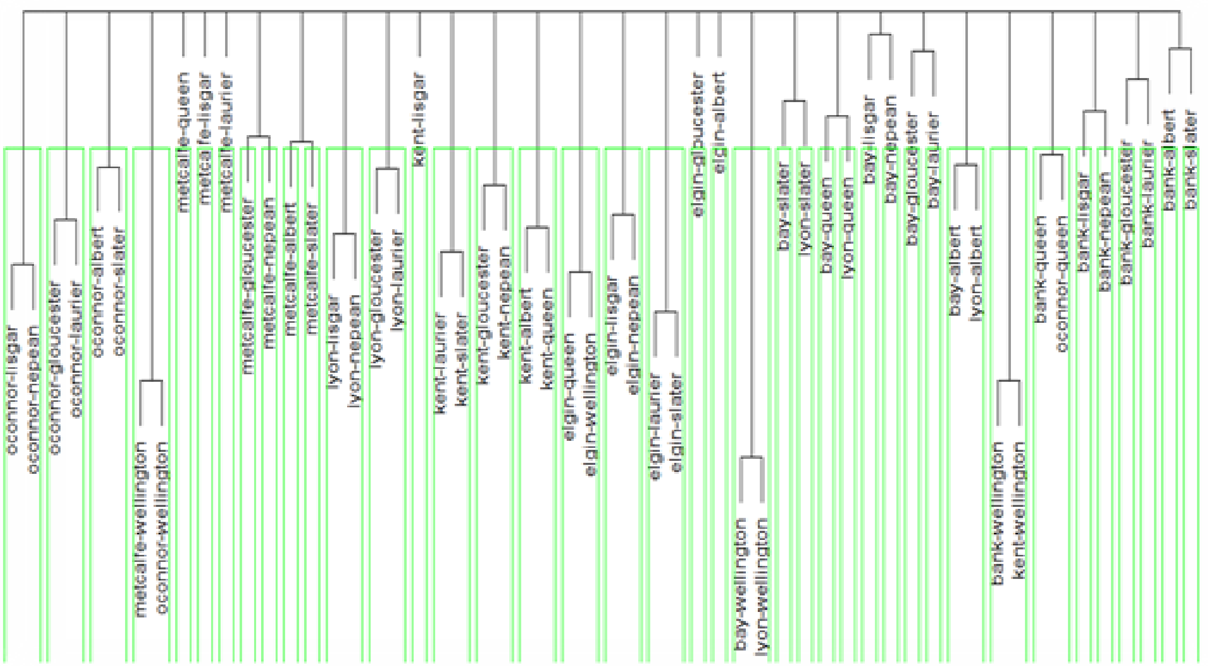


Figure 15 Complete link clustering dendrogram

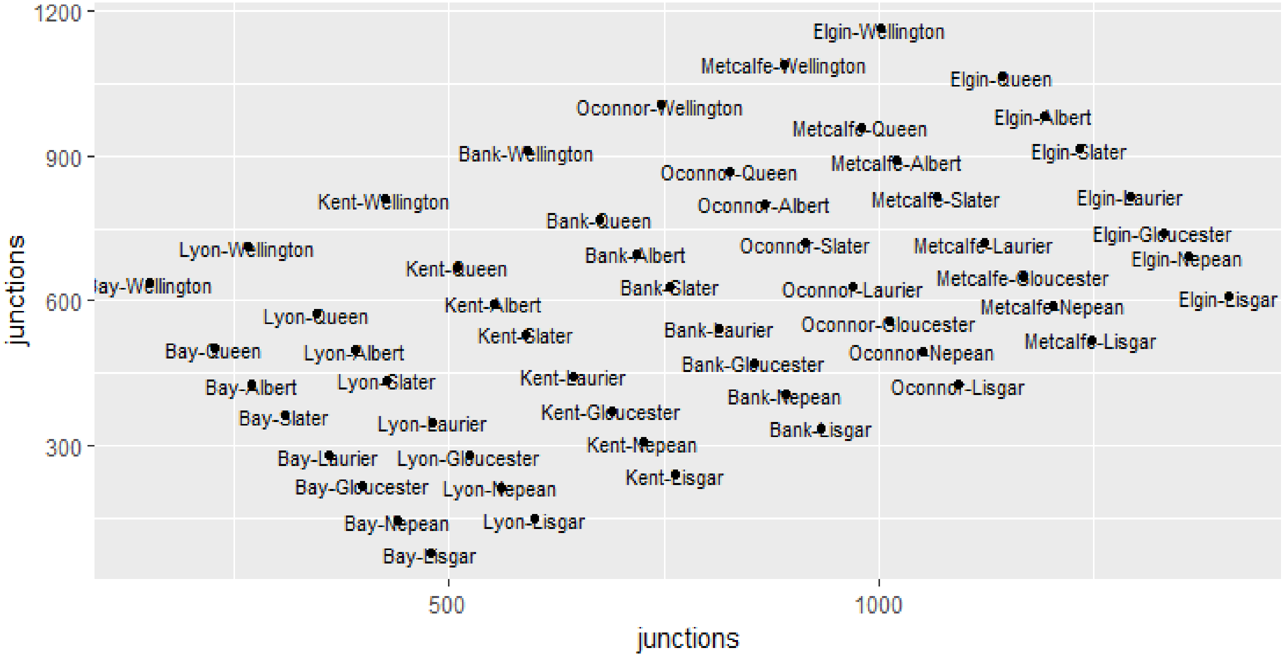


Figure 16 Non-clustered intersections

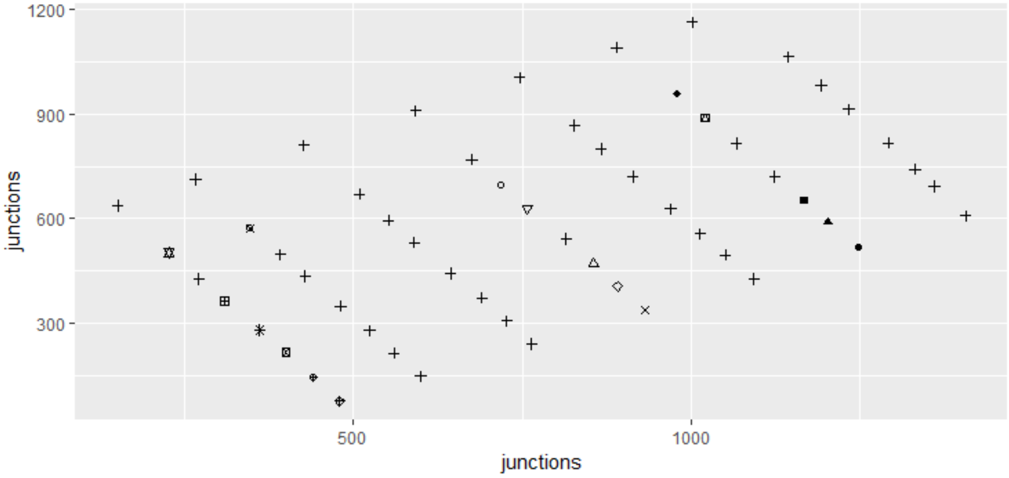


Figure 17 Clustered intersections using single linkage

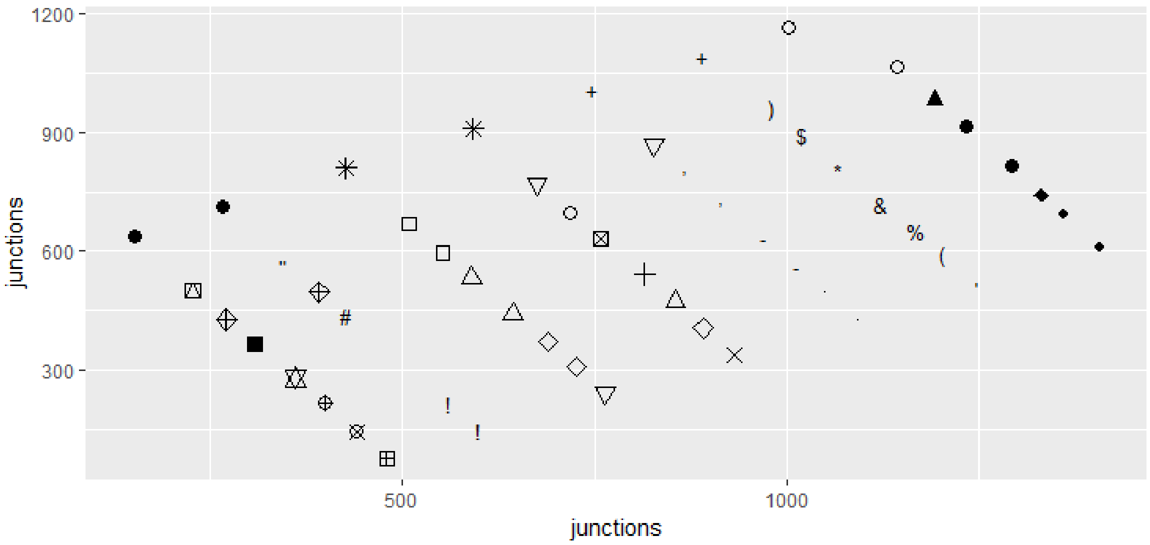


Figure 18 Clustered intersections using complete linkage

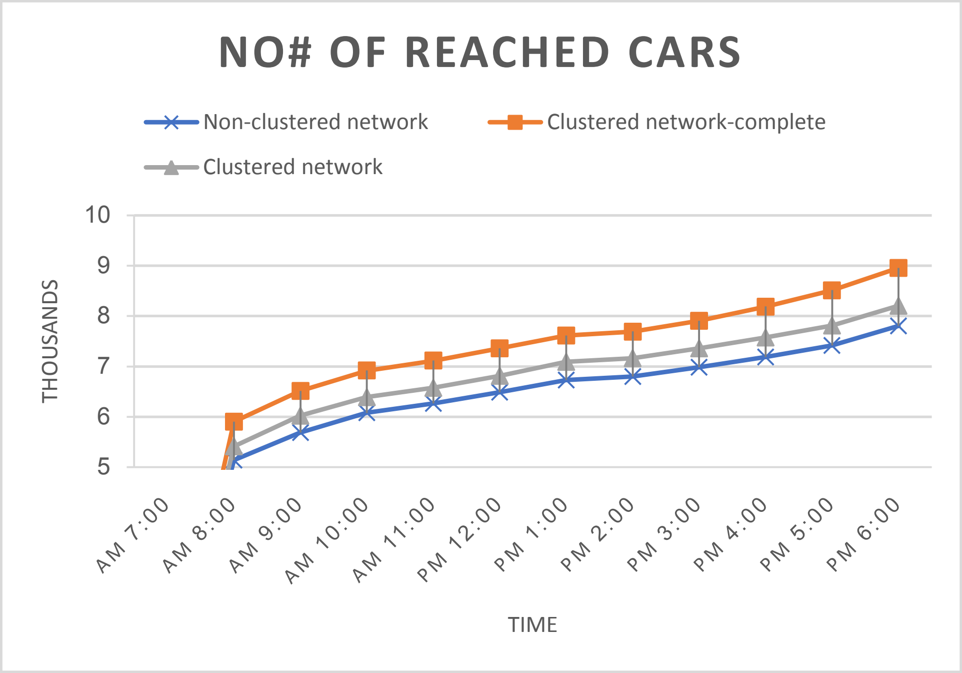


Figure 19 Throughput rate over day

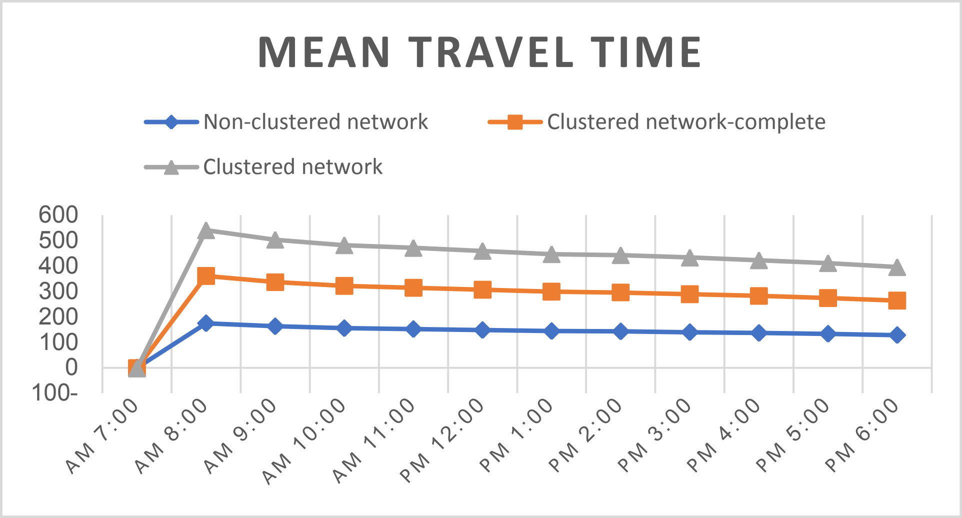


Figure 20 Mean travel time over day

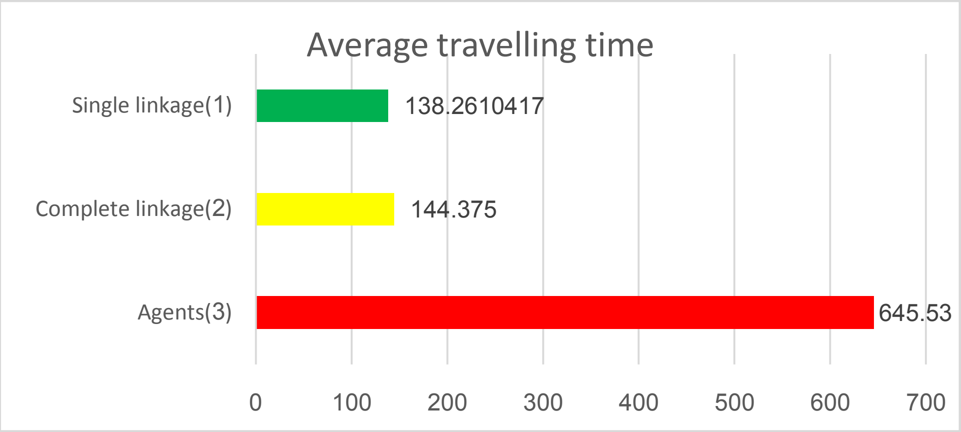


Figure 21 Average travelling time of different setup in seconds

In figure 21 we can see a comparison of average travelling time between the single linkage clustered network, completed linkage clustered network and the work done by [D.McKenney and T.White] , which was performed on the same district. Both clustered networks had much lower average travelling time (seconds) than the agent system, but it should be noted that, the agent system delay was caused due to having vehicles' routes reaching up to (259 seconds) of average travelling time.

The benefit of following the adaptive approach to generate signal plans, is that the plans implemented at an intersection are modified to serve the current traffic flows more effectively and the hybrid usage of fixed and adaptive traffic controllers in the clustered network provides faster trips than the adaptive system developed in [9].

6. Conclusion and Future Work

Generally, it is not easy to define appropriate state-action spaces in all real-world RL problems. Usually the tiling of the state space has to be rather fine to cover all possibly relevant situations and there can also be a wide variety of actions to choose from. Therefore, there exists a combinatorial explosion problem when trying to explore all possible actions and states. In our approach to improve the traffic signal controllers system and to overcome reinforcement learning challenges which are caused mainly due to exponential growth in the state-action space resulting in more communications between agents and increased learning time, the controllers were clustered using machine learning Hierarchal algorithm instead of using traditional techniques which are based on geographical attributes, into strongly connected sub-networks using the traffic volume as a similarity measure, a realistic scenario was performed and results were acquired. The increase of departed vehicles (reached its destination), in the case of single linkage clustered network is 5%, this slight increase is due to that, most intersections in the clustered network belongs to the same cluster which make it similar to the original network, except for intersections which were isolated while in the complete linkage clustered network it reached up to 15%. The proposed system was compared to [D.McKenney and T.White] which was performed on the same district as this work and it was shown the suggested technique resulted in faster travelling time. Decreasing the number of cooperative agents shall decrease the learning time referring to Whitehead, Steven D and Min Tan Looking at the simulation result, we can see that the clustered network achieved promising and higher overall performance than non-clustered network. Another direction for future work is to apply the proposed system to reinforcement learning based network to test and record the system overall performance.

References

[1] Kuyer, Lior, et al. "Multiagent reinforcement learning for urban traffic control using coordination graphs." Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2008. 656-671.

[2] El-Tantawy, Samah, BaherAbdulhai, and HossamAbdelgawad. "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 on14.3 (2013): 1140-1150.

[3] Labadie, John W. "Advances in Water Resources Systems Engineering: Applications of Machine Learning." Modern Water Resources Engineering. Humana Press, 2014. 467-523.

[4] Messer, Carroll J., and Ramanan V. Nageswara. Improved Traffic Signal Coordination Strategies for Actuated Control. No.

[5] Vital, Allan Saldanha, et al. "Development of Intelligenet traffic Lights using Multi Agent Systems." 12th WCTR july 11-15, 2010 lisbon, portugal

[6] Almejalli, Khaled, KeshavDahal, and M. AlamgirHossain. "Intelligent traffic control decision support system." Applications of Evolutionary Computing. Springer Berlin Heidelberg, 2007. 688-701.

[7] Wiering, M. (2000, June). Multi-agent reinforcement learning for traffic light control. In ICML (pp. 1151-1158). SWITCH/96/465110-1. Southwest Region University Transportation Center, Center for Transportation Research, University of Texas, 1996

[8] Cichosz, Pawel. Data Mining Algorithms: Explained Using R. John Wiley & Sons, 2015.

[9] McKenney, Dave, and Tony White. "Distributed and adaptive traffic signal control within a realistic traffic simulation." Engineering Applications of Artificial Intelligence 26.1 (2013): 574-583.

[10] Behrisch, Michael, et al. "SUMO–Simulation of Urban MObility." The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain. 2011.

[11] Wu, Xiaoping, et al. "Green-Wave Traffic Theory Optimization and Analysis." World Journal of Engineering and Technology 2.03 (2014): 14.

[12] Tan, Ming. "Multi-agent reinforcement learning: Independent vs. cooperative agents." Proceedings of the tenth international conference on machine learning. 1993.

[13] Jain, Anil K., and Richard C. Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988.

[14] Jain, Anil K., M. NarasimhaMurty, and Patrick J. Flynn. "Data clustering: a review." ACM computing surveys (CSUR) 31.3 (1999): 264-323.

[15] Carbonell, Jaime G., Ryszard S. Michalski, and Tom M. Mitchell. "An overview of machine learning." Machine learning. Springer Berlin Heidelberg, 1983. 3-23.

[16] Whitehead, Steven D. "A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning." AAAI. 1991.

[17] Balaji, P. G., X. German, and D. Srinivasan. "Urban traffic signal control using reinforcement learning agents." IET Intelligent Transport Systems 4.3 (2010): 177-188.

[18] OpenStreetMap, February2011. /http://www.openstreetmap.orgS.

[19] Comparing different techniques for controlling traffic signals vol 7 No.3 2016

[20] Mousavi, SeyedSajad, et al. "Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning." arXiv preprint arXiv:1704.08883 (2017).

[21] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8, no. 3-4 (1992): 279-292.

[22] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 1998.

[23] García-Nieto, José, Enrique Alba, and A. Carolina Olivera. "Swarm intelligence for traffic light scheduling: Application to real urban areas." Engineering Applications of Artificial Intelligence 25, no. 2 (2012): 274-283.

[24] Passos, Lúcio Sanchez, and Rosaldo Rossetti. "Traffic light control using reactive agents." In Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on, pp. 1-6. IEEE, 2010.

[25] Lin, Yilun, Xingyuan Dai, Li Li, and Fei-Yue Wang. "An Efficient Deep Reinforcement Learning Model for Urban Traffic Control." arXiv preprint arXiv:1808.01876 (2018).