Traffic Control and Infrastructure Organization Using Reinforcement Learning

Daniel Kuknyo

2022 Fall Semester

Abstract

The purpose of this research was to develop a reinforcement learning model that can optimize traffic flow in a city by building roads, lanes and intersections between predefined nodes that are defined on a user-friendly interface. The model will take into account how fast and dynamically does traffic flow and how friendly a city is to its residents: the number of lanes and infrastructure cost have to be minimal while also minimizing the time taken to travel between junctions. To conduct the experiment, the machine learning model uses a simulated environment that incorporates an intelligent driver model in order to provide the most accurate mapping between real life and the modeled environment. Future uses cases include optimizing current road structure and generating new paths between already existing cities or settlements that use the least amount of taxpayer money while also providing them with the fastest travel times. The developed framework can also be generalized for uses in public transport: using smart forecasting to optimize bus or train lanes to cover the most area, for as many people as possible.

Contents

Part I

Introduction

1 Historic overview

In recent years the traffic of cities became a rising topic with more and more city governments realizing that a motor-focused city design is unsustainable. Cities all around western Europe have started designing their cities around humans and public transit, and not around cars. E.g. Paris is planning to be a 15-minute city, Barcelona is incorporating the superblock design, Finland is creating non-intersecting paths between the common intersections and the Dutch are using intelligent traffic light systems to manipulate traffic flow in order to optimize it for both cars and pedestrians.

If one takes a look at how the Dutch infrastructure is designed, they will see that despite having less car lanes and overall less space for cars, the traffic flows more smoothly. This is thanks to the intelligent design of intersections, traffic lights and infrastructure. The methodology of this has been known ever since the 1970s, but the auto industry has been fighting against it ever since in order to gain more market. In Northern America one can observe what happens if a city is designed with cars in mind, requiring everyone to own a vehicle in order to participate in society. This results in worse accessibility to the city for the disabled, incapable, elderly and young people as well. The methodology of how to create walkable, human-centered and intelligent infrastructure that is optimal for both pedestrians and cars is laid out in detail by books from authors such as Strong Towns, an urban planning organization.

The aim for this research is to able to model a system of roads or city, and being able to pinpoint mistakes made by development engineers, with the goal in mind to make the city more humanly livable, liquidate urban highways and make traffic infrastructure more efficient.

2 Goals and Outline

The project will focus on building an interface that models traffic flow in a graph-based structure, then train a reinforcement learning algorithm to find the optimal configuration of the roads in order to transport the most cars in the most effective way possible. Here's when the urban design principles come in: one can easily observe that the most effective way to transport as many cars as possible is if all roads are 8-lane highways. However it's also easy to see that it's miserable to live in a city where there are no quiet, auto-low streets and only 8-lane highways. This might be the best configuration for cars, but it would make the life of people living in the city absolutely horrible. The rewarding system of the reinforcement learning environment will be designed in order to reflect these principles: building cost, traffic light/roundabout tradeoffs, how humans would feel living next to the road. The agent will have to decide where to build, destruct, or make roads 1-way to make the city's transportation flow dynamic but also make it livable for humans. The rewarding scheme will reflect the principles laid down by Strong Towns and other urban planning organizations significant in the field like Happy Cities: Transforming Our Lives through urban design.

The main metrics that are taken into account during the research are:

1. Time taken to arrive to the goal junction
2. Cost of infrastructure
3. Livability by humans (number of lanes, connection density)

3 Research hypotheses

If the research is successful, the following hypotheses can be verified or nullified:

1. Can traffic flow and driver behavior be modeled inside a reinforcement learning environment accurately enough for it to be representative of the real world?
2. Can predefined road configurations be optimized for traffic flow, human livability and cost efficiency at the same time?
3. Is there a single neural network architecture that can achieve the optimization in every configuration or there is a need for a more complex neural network as the complexity of the graph increases?

Part II

Research Background

4 City Design

4.1 A brief history of the modern day street

The shape and fabric of cities is the result of hundreds or thousands of years of development, transformation and reshaping. Many geographical and historical events play a role in determining how a city's transportation infrastructure is designed. The industrial revolutions have been de most markant inductors of change in most countries as they have led to the transformation of the walking-centered traffic. The most significant factor in city design has always been some idea of happiness or philosophy. E.g. the agora in ancient Athens, which was essentially a main square. However the philosophy of good life was built into it as it can be observed through monuments, temples, the most important buildings and law courts that have been surrounding it. And this practice of building and designing architecture hasn't changed ever since [7]. Even today, when a new skyscraper is built, it will reflect the architect's, CEO's or other executive members' ideas of happiness. They will look at it, and decide if they like it: is it good to go for building or not. But what has changed since ancient greece? What was the key motive that made us design cities focused around cars?

Philosophy has always been a pendulum oscillating between sky and earth - God and human. In the Middle ages, the focus was God. Then following it in the Renessaince, the human became the center. After that came the Baroque, again with God as the main motivator. After the enlightenment and the birth of the modern day citizen, the first style trend was Rococo, focused around humans. Around this time has religion lost its status as the main, unified cultural narrative as it can be observed in the works of Nietzsche and Derrida. The last attempt at constituting God as the center of philosophy was in the style trend of Romantics. However as the citizen of the enlightenment wasn't religious anymore, the style attempted to find God in the human (übermensch - Nietzsche). This is expressed in various other fields, not just architecture or city design: art, poetry, statuary, astronomy and even mathematics.

Around a hundred years ago in the 1920s cars have begun to appear in the cities in greater and greater numbers. At first it was a new and expensive technology, hence it was mostly the wealthy class that owned them. At this point in time traffic rules were essentially nonexistent, so drivers could drive however they pleased. This quickly led to a large number of traffic related incidents. This was a massively bad PR for the auto industry, so they started influencing city design using their political and economical capital [10]. In school the children started to receive education about having to look both ways before crossing the road [13], because they are the ones that don't belong on the road and not the other way around. According to Peter Norton [12], the key turning point came in 1923, when the citizens of Cincinatti had to vote on a referendum that was planning to limit the maximum driving speed to 25 mph. The passing of this bill would have meant that car dealerships would have a very difficult jobs in selling their cars and essentially experience a drop in revenue, business and political influence. The representatives of the car dealers have did everything in their power to stop the referendum like billboards, flyers, public advertisement and even paying people to vote *no* on the ballot. Eventually the voting has turned in favor of the auto industry, and they gained political and economical influence over the course of the years.

4.2 The current state of city design

Ever since, car dependent infrastructure and lifestyle is overall in the USA. Owning a private vehicle has become a minimum requirement to take part in society as the sprawling has led to a run-down public transit and railway ifrastructure. As of the current state, the automobile is dominating the transport in the USA and hence becoming one of the most important determinants of the American lifestyle[15]. This can be attributed to the way cities and residential buildings are designed. A typical American city has overwhelmingly two types of residential buildings: skyscrapers and high-rise buildings downtown and low-density suburban sprawl around it. What's missing is a tradeoff between the two: mid-density, mixed-use housing that can at the same time house a sufficient amount of people in a relatively small area, but at the same time is relatively cheap to build in comparison with skyscrapers. This mid-density residential housing planning would be the key guarantee to design walkable neighborhoods in the United States, however it's missing. Of course, there are some areas where this type of architecture can be found but this is not the norm, it's the exception. This is referred to as the missing middle problem [14]. This is party because zoning laws in the USA don't make it possible to build this type of building.

From the way the cities are built comes car dependency as a consequence. However this has also led to public transit being under-developed: the housing density in the suburbs doesn't reach the threshold to make it worth making it being part of a bus or train route. No matter where the city puts the bus stop, the time taken to get there on foot in the area is too much for people not living near it. Putting bus stops overall would require too much public transit infrastructure that would have to be subsidized from taxpayer money and would result in near-empty buses. The optimal scenario is when a transit route can reach just enough people so that the maintenance and operation can be financed from ticket fares and minimal amount of subsidies. This is called the critical density. If there is a critical density reached and a single transit stop can be serve enough people, which will make it worth in fare revenue. It's also rare to find dedicated bus and cycling lanes in the USA. Buses are generally stuck in traffic with the cars in rush hour, which makes it even less inviting for passengers. It's worthwhile to note here that adding lanes to an already existing road does not solve traffic, so it's not a solution to just make roads wide enough so that buses don't have to wait in line together with motor vehicles. As an example consider the recent expansion to the I-10 highway south of Houston known as the Katy Freeway. Together with a recent expansion, it's now at 26 lanes at its widest part but traffic jams are a regular occurence. Adding lanes also reduces the safety of the road, as found by [11] who were analyzing the relationship between the road geometry and safety, and also [1], who found a positive correlation in the numbers of crashes and the number of lanes in urban highway road intersections. Dedicated bike lanes are also missing from the infrastructure. The only bike lanes one can find are painted bicycle gutters without any separation from high-speed motor traffic that makes it very dangerous to ride a bicycle. This results in even more unequality in society.

Apart from the reasons mentioned before, car dependency has another different aspect, CO_2 emissions. A study has found that despite being 5% of the total population, the United States is responsible for 45% of all transportation CO_2 exhalation [4]. A different work in the field states that the most transport emissions could be reduced by converting low-density housing to mid-density [5]. Even more than converting mid-density to high-density.

4.3 The stroad

The stroad is a term created by a non-profit urban planning organization called Strong Towns that has been referred to and processed by several relevant pieces of literature like the book Confessions of a Recovering Engineer - Transportation for a Strong Town [9]. The idea of the stroad lays down the guidelines which have to be reflected in training the agent. To be able to understand the the stroad as a concept, it's necessary to first grasp the functions and purpose of a road and a street.

4.3.1 The road

The road is a high speed connection between places. Because of the high speeds involved the roads are wide and forgiving. The exits on the road are far apart to minimize the number of intersections and to keep vehicles at a high speed for the maximal amount of time. The more exits a road has, the more drivers need to pay close attention as the vehicle in front of them might be coming to a brake more often. This yields an extra danger factor to the driving experience. The solution to this problem is that on many roads there are separate turning lanes, which are dedicated to slowing down before taking an exit from the road. The lane next to the road is called the clear zone. It's a lane devoted to be left empty in case there's an emergency and a vehicle needs to stop or if an emergency vehicle needs to bypass a line of traffic. The signs the font on them are large, as they are meant to be read from a distance. An example for the road is the M3 highway in Hungary. Roads are very important pieces of infrastructure as they are optimized for safety while going fast.

4.3.2 The street

The street is a high-density environment that is designed for humans. It's a complex space where life in the city happens, so you will find people here. The street is dedicated to low speeds as it's a destination, not a through fare. Fast traffic speed is not compatible with human activity as it adds another danger factor to commuting. Buildings are right next to the sidewalk to be easily accessible by pedestrians. The feel of the street resembles something of an outdoor room. Everything is reduced to the scale of humans and it's designed to be encouraging and inviting for people to walk, socialize window shop or just simply get some fresh air. There are many entrances and exits to and from the street for shops, restaurants, caffés and services. Having an apartment near a street can also add value to the property. One can compare the price of a flat near Váci street and M3 highway and immediately see how much society values human factors taken into account when designing our city. Streets are very important factors for tourism, human livability and quality of life inside the city.

4.3.3 The stroad

The stroad is a name that's the result of the combination of the words street and road. It describes an urban motorway that is neither a street nor a road, despite trying to be both of them. The stroad is a street that's designed like a road, and doing so it fails to bring forth the advantages of either of them, but is struggling with the disadvantages of both of them. First the stroad is analyzed from the viewpoint of a road.

On the stroad there are many entrances and exits to businesses and services, like one would find on a street. However there are multiple lanes of high-speed motor traffic in both directions. This is an added factor of danger, as there are typically no dedicated turning lanes, so drivers have to pay constant attention if the car in front of them is taking a turn. Because of the high speeds involved this means that drivers have to come to a quick halt. It's not rare that stroads allow the 70 km/h speed limit, which drivers often surpass, because the stroad is straight and wide. The geometry of the street allows the drivers to reach higher speeds while still feeling safe, however this is a fake sense of safety, as there are more points of conflict than on a dedicated highway [8]. Roads can keep vehicles on high speeds because there are few exits. Traffic jams are also a common occurence. Because of the large distances incolved, walking feels very uncomfortable in this kind of environment. People don't feel safe walking next to high-speed motor traffic. It's also common that one can find large parking lots, which are also not inviting for the human as there's nothing to see or do there. It's also difficult to cross the stroad unless using a dedicated crosswalk and treading very carefully not to be involved in an accident. This is clearly a place for cars so it's not surprising that pretty much everyone drives here.

Next the stroad is evaluated from the point of view of the road. On the stroad the lanes are wide and there are a lot of them. Traffic flow is more chaotic than on the road because there are a lot of exits, hence cars change lanes more often, easily giving the change to dynomen situations: when there's no imminent danger, but the conditions can turn dangerous at any moment. This also reduces the speed of traffic flow. Cars changing lanes are the leading cause of the caterpillar or butterfly effect. Traffic volumes are also high as often there are no viable alternatives to driving: bus and cycle lanes are nonexistent. This kind of car dependent infrastructure creates distances that only cars can bridge, but not everyone owns a car in society. This city design inherently widens the gap between the different layers of society.

Stroads are also very expensive, as they are built to a highway standard despite the fact that they can't be used as efficiently: with high speeds, while maintaining a strict level of safety as in the degree of slope, sharpness of turns and several other design decisions that have dedicated highway engineering and maintaining offices in almost any country. The density of intersections requires the city to install traffic lights, as they are the only type of intersection that can be operated with a related safety for a road of this scale. Traffic lights in the US can cost up to $750.000 for an intersection. Roundabouts and right hand priority intersection are not viable options here because of the high speeds involved. Drivers need to see if they have to slow down ahead of time.

The majority of crashes within the United States happen on stroads. People walking and cycling are especially in danger as there's no infrastructure that separates them from high-speed motor traffic. The wide and open design of the stroad encourages drivers to drive fast and not pay attention to their surroundings [3]. As an example, during the coronavirus lockdowns in the beginning of 2020, people weren't travelling as much, hence miles travelled by drivers decreased. Despite 355 million less miles driven fatal car crashes per mile increased by up to 34% [2]. This leads to the conclusion that the bad design of the stroad isn't killing even more people is that it's because they are usually so jammed up that the drivers can't go fast enough to kill eachother.

5 Deep learning in Infrastructure organization

5.1 The spectrum of expert areas

Predictions in new literature all point in one direction regarding traffic control and artificial intelligence: AI will most likely become the brain of infrastructure and traffic. With the rise of autonomous vehicles this possibility seems closer than ever. There's a huge potential in operating, managing and monitoring transportation. The infrastructure is going to be very similar to that of the internet currently: a decentralized network that is operated by mesh and local servers, data flow being sensor readings coming to and going from the several vehicles on the road. There are currently existing studies on the viability of smart networks that infuse different kind of sensors with road infrastructure [6]. For public transportation the LOA and GOA frameworks lay down the principles of creating an ethical AI-driven solution for trams and trains respectively [16]. There are also works that use traffic light governance systems enhanced by artificial intelligence to aid faster traffic flow [17]. Reinforcement learning has also been applied to control the cycle, phase and red-green ratio of traffic light systems [18]. All of the studies in the field can be placed on a spectrum that has the vehicle on one end and the road infrastructure on the other. Some works develop systems that enhance the vehicle, some are creating instruments for the road that directly influence traffic safety or speed like traffic lights. Others connect the road and the car together by infusing them with intelligent sensors that add a level of safety to the travelling experience.

5.2 Infrastructure and building modeling

Building information modeling (*BIM*) is the collective name for technologies and methods that aid the generation and modeling the digital representations of places in order to have a better overview of the building process. This introduces cost effectiveness, a level of safety and better ways to understand environmental impact. BIMs are essentially computer files that contain information on an object in order to help decision makers and planners with design, building and overview. Different kinds of dynamic data can also support BIMs like sensor measurements and signals from the building systems. Advances in fields like computer vision and artificial intelligence has given rise to a technological revolution in building information modeling. There have been attempts where building models were constructed using a combination of LIDAR and photometric sensor data.

Part III

Methodology

6 Constructing a city

Everything starts with a city that the agent will have to optimize the structure of. At this point the city is best described by a graph. Graphs are high-level descriptors of connections between nodes. This gives an easy to use but highly customizable method to define a city to optimize. To define a city using the graph there are two things needed to be defined:

1. List of nodes of the graph with coordinates: $N_1, N_2, \ldots, N_k$.

2. List of connections between the nodes: $N_i \to N_j$.

Together they are called a *construction protocol* from here on.

The developed software product will provide an interface where the user can make a graph, describing the intersections (nodes) and roads (edges) of the city in question. This for example can be done in GeoGebra and exported into a construction protocol in order to work as an input for the model. Below is an example simple city constructed:

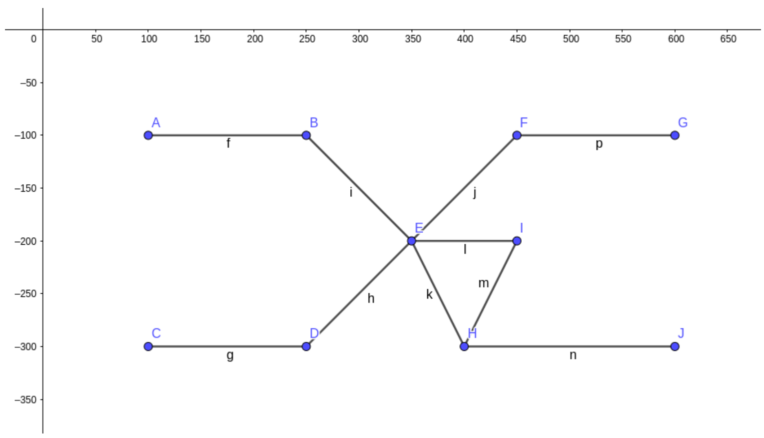


Figure 1:

The interface will read a construction protocol, construct a graph and all the possible pathways from it. As a starting configuration every road is 2-way on the edges that have been defined in the graph. The vehicle rate and distribution can be controlled before starting the simulation. A constructed "starting" city according to the previously shown graph will look like as follows:

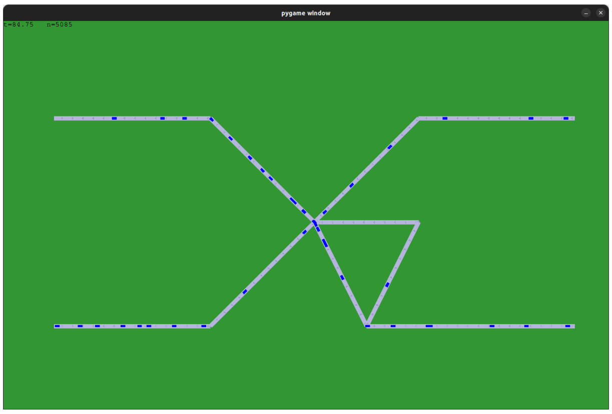


Figure 2:

So far this is a simple setup for demonstration. The vehicles are passing from one entry point to another, without necessarily choosing the shortest path, or being evenly distributed among all the roads, just as one would find in real life. The driver model will incorporate an intelligent behavior, like slowing down after the car in front is slowing down or gradually speeding up after a light has turned green with a comfortable acceleration parameter.

The road configuration will be examined with multiple metrics like how many steps does it take for the roads to transport 100 cars or how much the road infrastructure would cost. If the agent is handed a road configuration it will be able to find the optimal one, with the least cost, least unnecessary roads and fastest transportation for a given amount of cars.

As a distant goal it would also be reasonable for the application to accept Osmosis data and construct a graph based on that. This would require a more sophisticated preprocessor as the Osmosis data would have to be stripped of metadata and guarantees of format conversions between world coordinates and graph node coordinates would have to be implemented. However it is possible to convert into the inner representation format of the road simulation module. Osmosis construction has the following steps:

1. Highlight a map segment

2. Load it into a graph

3. Load the graph into a simulation

The steps of the transformation are visualized on the following graphics from left to right:

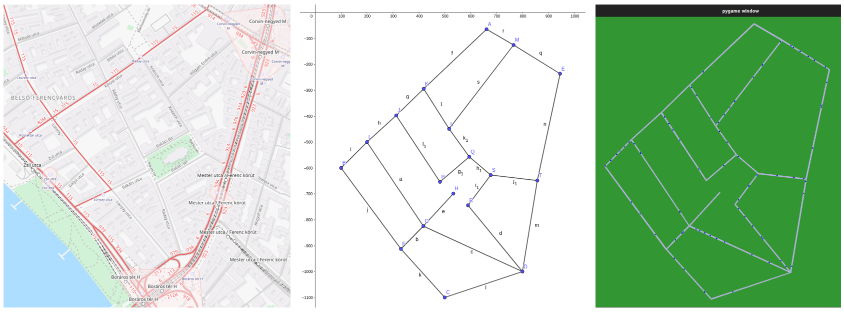


Figure 3:

7 Intelligent driver model

The intelligent driver model is a time-continuous car following model for the simulation of urban traffic. It describes the behavior of the drivers. and the positions of vehicles. The model defines 6 parameters that influence how a driver behaves:

1. v_0: The desired velocity of the vehicle.
2. T: Safe following time.
3. a: Maximum acceleration.
4. b: Comfortable deceleration.
5. δ: Acceleration exponent.
6. s_0: Minimum distance.

Using these variables and the positions of other vehicles it is possible at every time step to completely determine the position of every vehicle. Furthermore, vehicles can influence each others' positions, as there can be car pileups in an intersection with cars waiting for each other. This model will consider only cars of uniform size, without taking into account other types of vehicles like vans and semi trucks. Every driver is assumed to have the same skill set e.g. safe following time, comfortable deceleration and so on. This can be a fairly good approximation of a real world driver's parameters.

The simulation of the drivers' behavior is highly dependent on time, as they are regulated by differential equations that are implemented using computer code. Each time the agent will execute a training iteration, it will have to run a simulation. However, the simulation is not possible to run so many times in real-time. The game-time will have to be implemented using discrete time steps, that can be sped up to arbitrary speed.

8 Modes of operation

There can be several different ways to use the framework depending on what the target of the prediction is. The modes depend on what actions are allowed. The more actions are allowed, the more the city will be redesigned from the ground up.

8.1 Lane capacities only

In this operation mode the nodes and junction types are fixed. The agent is only trying to predict how many lanes should be going from one intersection to another. The agent can add and remove lanes to existing infrastructure. This is the ideal mode of operation if there is a currently existing city, with already existing streets as it's not feasible to demolish residential apartment blocks just to build a new road between two intersections.

8.2 Lane capacities and roads

This is the same as the mode of operation mentioned before, the only difference is that the agent now has the capability to add a new road between junctions. This is the optimal configuration if it's the focused map segment is not already inside a city, but rather in a rural area where fast connection to nodes is a matter of hours of travel time on the highway.

8.3 Junctions only

The junctions only modality focuses on the nodes of the graph exclusively. The goal here is to determine what intersection types would be best fit to control traffic flow in an already determined road configuration. The agent has to choose between right-handed, roundabout and traffic light for every junction in the map. Junctions only is the optimal setting in an environment where there's a narrow city e.g. like in Western Europe but there's room for optimization in terms of intersections in order to determine the configuration that yields the fastest traffic flow.

8.4 Junctions and roads

This is the hardest task in terms of optimization and urban design. The mode allows the agent to set each intersection type, add lanes to already existing roads and add new roads between junctions. Newly built cities, already existing districts that are to be redesigned and between-city highways, country roads can also use this method of optimization. The research will focus on this mode as this is the most general.

Part IV

Deep learning architecture

9 State space

9.1 Representation

First, the plan will focus on how to represent a certain type of road between two nodes. The state between graph node A and B will have to be represented by a single number in every case. The bases that this scalar value will have to cover the number of lanes between $A \to B$ and the type of intersection in the node B.

For a state-vector element corresponding to the one-way connection between nodes A and B the possible values are as follows:

1. One-lane road between $A \to B \Rightarrow 1$.
2. Two-lane road between $A \to B \Rightarrow 2$.
3. Two-way road, one lane in each direction: $A \to B \Rightarrow 1$; $B \to A \Rightarrow 1$.
4. Two way road, two lanes in each direction: $A \to B \Rightarrow 2$; $B \to A \Rightarrow 2$.
5. One-lane road ending in roundabout between $A \to B \Rightarrow 11$.
6. Two-lane road ending in a traffic light junction between $A \to B \Rightarrow 22$.

The state vector keeps track of the connections in a directed fashion, storing the number of lanes from $A \to B$ and $B \to A$ in separate values. A single scalar of k means that it's a k-laned road ending in a right-hand priority intersection without any dedicated infrastructure. A scalar of value $10 < k < 20$ means that the road ends in a roundabout. A scalar of value $20 < k < 30$ means that the road ends in a traffic light based junction. The last digit always translates to the number of lanes going from $A \to B$.

9.2 Naive implementation

The state space requires precise designing and execution as there's a possibility of exponential explosion if it's represented using an all-to-all fashion. If the state space keeps track of all the incoming connections from all the nodes to all other nodes, for a graph G of nodes $N_1, N_2, \ldots, N_k$the state space will require a vector of length k^2. In this representation a self-loop $N_i \to N_i$ denotes the type of intersection found in the node.

N_1	N_1	...	N_1	...	N_k	N_k	...	N_k
N_1	N_2	...	N_k	...	N_1	N_2	...	N_k
$x(N_1, N_1)$	$x(N_1, N_2)$	...	$x(N_1, N_k)$	...	$x(N_k, N_1)$	$x(N_k, N_2)$	...	$x(N_k, N_k)$

Where $x(N_i, N_j)$ is the state vector scalar that belongs to nodes i, j e.g. the number of lanes between the two junctions. This gives reason to explore the possibilities of being able to shorten the state space vector by eliminating connections that are for sure not going to participate in the learning. The rationale behind this is that it's an erroneous design decision to connect the furthest junctions with direct roads. This would mean destroying currently existing residential zoning and thereby cut the city in half. The state vector must remain the same size during the learning, so if it's decided before training that an edge won't participate in the model, it can't be added later on as it would change the input dimensions to the neural net.

9.3 Radius-based approach

It's easy to notice that it isn't needed to have roads connecting all intersections with all other intersections. This would make the city a convoluted mess. A valid approach to reduce the size of the state vector is to define a radius around the nodes and only store the number of lanes between them. For this a radius parameter will have to be input in order to define the possible connections before running the simulation. On the image below an example of such a circle is shown for graph node E with the radius of 150.

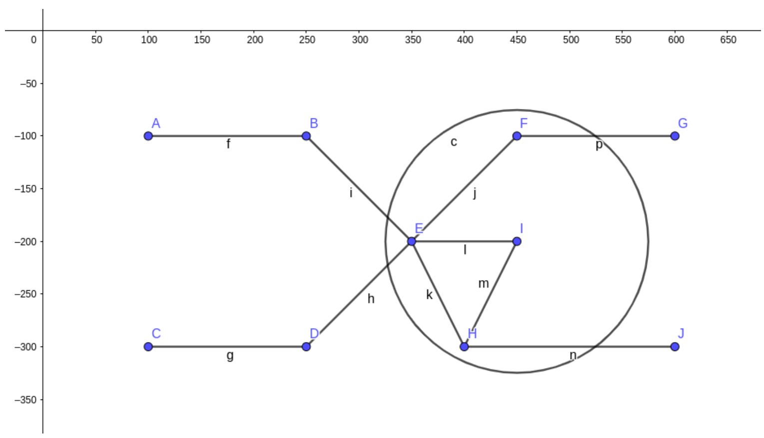


Figure 4:

It is fairly straightforward to see that this would result in a configuration where no road's length exceeds the radius of the neighborhood. This can be considered as a good design approach because it would eliminate the need to build long and straight roads through the city which are shown to be more dangerous regarding traffic accidents. The reason for this is that motorists can go with higher speeds as the road is wide and forgiving like a highway. It is empirically shown that the best way to reduce speeds is to force the drivers to obey speed limits by creating the geometry of the road in a way that enforces it. One won't go with 70-90 km/h on a street that is say 150 meters long.

The disadvantage of this method is that the state vector would be unevenly distributed between the nodes of the graph. Some of the nodes would have more elements in the vector than others resulting an unnecessary high representation. It's also possible that an outlier node gets cut off from the rest, and becomes an 'island' that is unreachable from anywhere. This happening is considered a design fail.

9.4 NN-based method

The number of tracked connections in the state vector can also be reduced if only the m nearest neighbors to a node is allowed. This would allow for a more evenly distributed state representation as for each node there's only a need to register m connections instead of $k-1$. This approach would however also limit the number of incoming connections into a junction. E.g. if the parameter is set to $m = 4$ the previously shown graph would not be possible to construct and some node may be separated from all other ones, isolating them from the rest of the infrastructure. This is considered a bad resolution of the problem, but the approach can be used in combination with other approaches. The procedure is visualized before for node J with $K = 3$. The red dotted lines represent the allowed connections from the node.

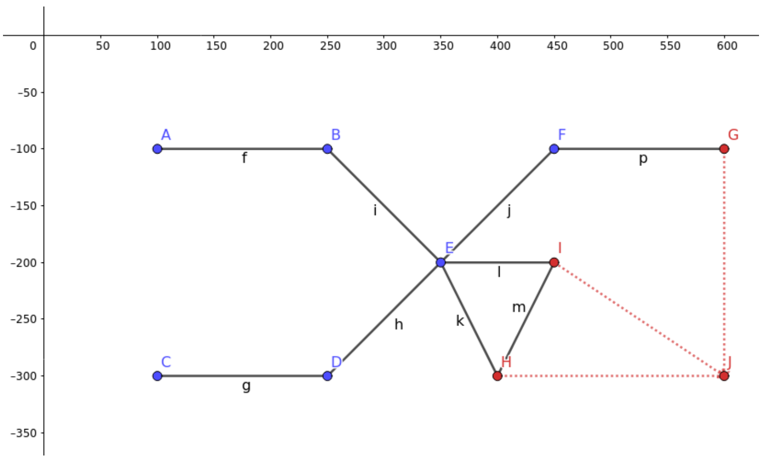


Figure 5:

9.5 Practical implementation

A combination of the previously mentioned methods can also be taken into consideration in order to reduce the state vector size, e.g. allowing connections from a specific radius but in cases where there's not a sufficient number of possible neighboring nodes the system will have to find the m closest nodes and keep track of roads coming and going from them. It's also worth noting that combining the advantages of the two methods intelligently can yield a representation where each node has a specific, even number of connections that is significantly less than the total number of nodes in the system. It's also a requirement that each graph node shall have at least one connection towards another, because the contrary would yield an erroneous resolution of the task.

9.6 Matrix-based approach

The previous paragraphs described how can the state be structured in a one-dimensional manner. However it's also possible to create a representation in a matrix form. This opens up new possibilities and challenges as well. In the matrix-based approach each entry to the matrix describes a connection between two nodes, like a table. For k nodes the descriptor matrix is of size $k * k$. The diagonal elements describe the self-connections, so the type of intersection that can be found at a junction in the city. The representation is as follows:

	N_1	N_2	$\dots$	N_k
N_1	$x(N_1, N_1)$	$x(N_1, N_2)$	$\dots$	$x(N_1, N_k)$
N_2	$x(N_2, N_1)$	$x(N_2, N_2)$	$\dots$	$x(N_2, N_k)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N_k	$x(N_k, N_1)$	$x(N_k, N_2)$	$\dots$	$x(N_k, N_k)$

This strategy yields advantages and disadvantages as well as the one-dimensional scheme. The downside is that the matrix's size is fixed: there can be no longer any edges that can be left out of the learning. If they are fixed on the value 0 to denote this, there is still a necessity to process them at each training iteration as deep learning libraries rarely have an option to manually skip a connection. This would lead to complicated program code and would raise more errors rather than gain execution speed.

The upside is that with this method there is now a possibility to use two-dimensional convolutional processing as the matrix can be processed as a single-channel monochrome image. This leaves possibilities to explore.

Another key insight is that the order of the columns doesn't actually matter: they can be ordered in any order so long until the connection descriptor elements are also in the right place. This can be a useful trait to exploit as the nodes can be added in order of closeness to a keypoint or sorted by any other comparison metric. The matrix would now become not just arbitrarily ordered, but also the 'pixel' positions would also have a meaning to convey.

10 Action space

A starting configuration will provide the agent with a 2-way road between each graph node. From here on the goal is to add / destroy roads, lanes and intersections in order to optimize the throughput and cost of the road. Each action of the agent will take two graph nodes as a parameter and the roads will be configured accordingly. Any action for node A and B will assume a single directed edge $A \rightarrow B$ starting for A and ending in B. There are cases where semantically it would make more sense to have more or less than 2 parameters but these cases can be generalized to an action with two parameters and hence be channeled into a neural network output of the same shape and size as all other cases.

The list of actions for the discrete action space:

1. *add_lane*(A, B)*:* Adds a single one way lane going from $A \rightarrow B$.
2. *remove_lane*(A, B)*:* Removes a single lane going from $A \rightarrow B$.
3. *add_road*(A, B)*:* Adds two lanes between the nodes A and B going $A \rightarrow B$ and $B \rightarrow A$. Only valid in case of nodes that don't have edges connecting them.
4. *remove_road*(A, B)*:* Removes two lanes between nodes A and B going from $A \rightarrow B$ and $B \rightarrow A$. Only valid between nodes that have edges connecting them.
5. *add_trafficlight*(A, B)*:* Creates a traffic light system to all roads entering the intersection of graph node B.
6. *add_roundabout*(A, B)*:* Adds a roundabout to all roads entering the intersection of graph node B.
7. *add_righthand*(A, B)*:* Removes current traffic light and roundabout infrastructure to create a right-hand priority intersection in graph node B.

The action space presented here corresponds to the operation mode '*Junctions and roads*'. The action space corresponding to *'Lane capacities only'* is $\{1, 2\}$. For *'Lane capacities and roads'* the set of actions is $\{1, 2, 3, 4\}$ and for *'Junctions only'* it's $\{5, 6, 7\}$.

11 Rewarding mechanism

At each time step in the reinforcement learning agent-environment framework, the agent takes an action and as a result, the environment changes its state and returns a reward to the environment. The agent can also observe the environment through the state variable described in the previous section. This part will deal with the principles that will be reflected during sending the reward signal to the agent. The main directives like exact costs and traffic engineering perspectives will be further explored in the literature review. For now it's enough to deal with the main.

The main goal for the agent is to build a city that is livable for humans by keeping the number of lanes and roads to the minimum at all times. If this constraint was not present, the agent could simply build 8-lane highways in streets that are a few hundred meters long. This would render being a pedestrian miserable. It's also necessary to not build roads between nodes that is not necessary from a transportation perspective. If there's a reasonable detour between two endpoints, the cars should take that road.

11.1 Inter-node infrastructure

The cost of building a lane has to be taken into consideration. It is more expensive to add new lanes to existing infrastructure than to build the first lanes at the beginning. This is because the sidewalk has to be destructed and then rebuilt. Also traffic flow is slower in the time of the construction because of closed roads, leading to congestions. Out of the 7 possible actions of the agent 4 are in relation to building roads and lanes. The relationship between them is the following:

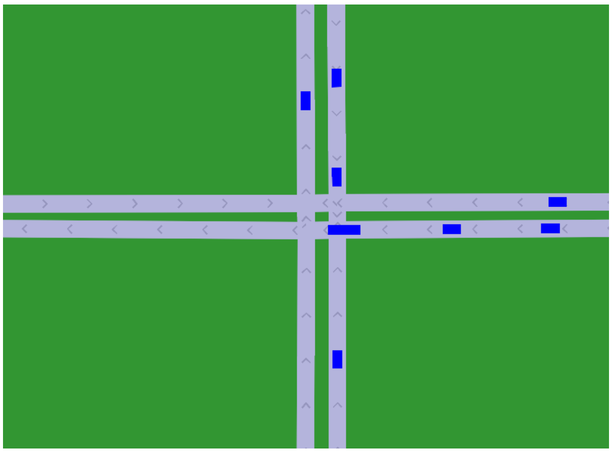
1. The cost of building and destroying is linearly dependent on the length of the segment. The length cost of a unit of lane is l_{lane}.
2. The cost of building a unit (e.g. meter) of one-way lane is: c_{lane}.
3. The cost of building a lane is taken as $b_{lane} = c_{lane} * l_{lane}$.
4. The cost of building a road is equivalent to building two lanes: $b_{road} = 2b_{lane} = 2 * c_{lane} * l_{lane}$.
5. The cost of destroying a lane is less than building a lane. The coefficient parameter is denoted δ: $0 < \delta < 1$ Then the cost of destroying is: $d_{lane} = \delta b_{lane} = \delta * c_{lane} * l_{lane}$.
6. Then the cost of destroying a road: $d_{road} = 2\delta b_{lane}$.

The necessary inputs are: c_{lane}, δ.

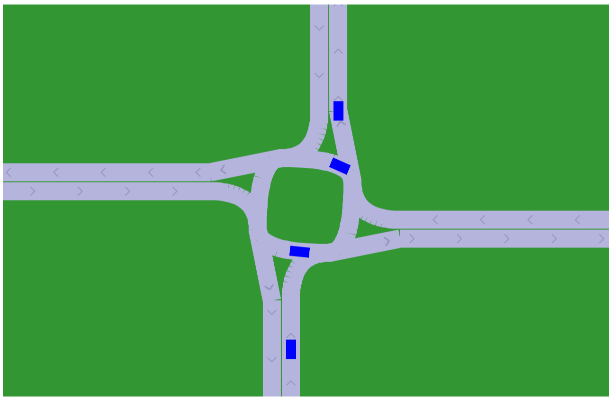
11.2 Intra-node infrastructure

The next section will describe the cost of building different types of junctions. The types of junctions in increasing order of cost are:

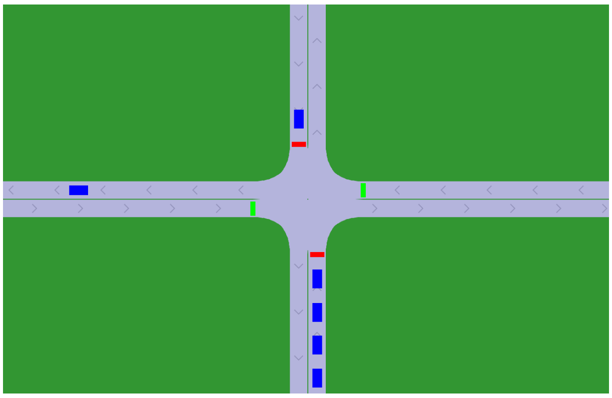
1. Right-hand intersection: it is the cheapest, however it is also the most unsafe. Cheap to build and to maintain. Also inexpensive to convert to from all other types of junctions.

Figure 6: 

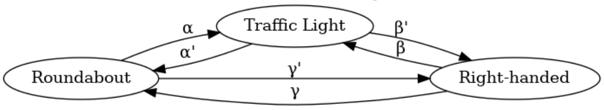
2. Roundabout: a reasonable trade off between cost and safety, however it is difficult to destroy an intersection and build a roundabout in place of it as it requires widening the intersection.

Figure 7: 

3. Traffic light: the most expensive and the safest type of junction is the traffic light one as it requires dedicated infrastructure and the maintenance costs are higher than in case of all the other junctions.

Figure 8: 

As a graph node must be at all times assigned to a type of junction, and each type can be converted into any other type, the relationship between them can be defined by a triangle. The cost of converting from one to another can be parametrized and can be subject to change depending on how expensive building infrastructure is on a given terrain or economical environment. The conversion table is shown below:

Figure 9: 

The conversion will require the inputs to and from each type of intersection, as these are not necessarily functions on each other. Some dependency can be added later but that is independent of the current high-level model. The required inputs are: $\alpha, \alpha\prime, \beta, \beta\prime, \gamma, \gamma\prime$. Each variable represents a conversion cost between two types of infrastructure. The prime version of the variable is the reverse of the same building procedure. The prime ($'$) will always reflect a *'destruction'* e.g. demolishing an expensive piece of infrastructure and building a cheaper one in place. The non-prime variable denotes the building of a more expensive piece of infrastructure from a cheaper one.

There are limiting factors to different types of junctions e.g. how many roads can it intersect. The roundabout is famous for being able to intersect five or more roads without any problem, because of the structure of it. Traffic lights can also have more roads than average coming in and out of it. For the most part right-hand intersections can handle at most four bidirectional roads. This is because human attention cannot be focused to so many places at the same time. More roads will translate to a more dangerous environment and will require highly dedicated infrastructure. It is possible, but it's rarely seen in the real world and the cost will have to be tuned accordingly.

12 Network architecture

The modeling structure is vastly dependent on the task that is given to it, therefore this section will only give a general overview of the inner workings of the deep learning model. The main steps of a single iteration of simulation are given below

---

Algorithm 1 The modeling process

Input: a state vector x that describes connections $A \rightarrow B$ in a graph G.
Output: the road configuration which is optimal for travel time, cost and livability.
Repeat until stopping criterion isn't met:

1. Agent observes state vector x.
2. Agent chooses a *'from'* node $A \in N_1, N_2, \dots, N_k$.
3. Agent chooses a *'to' node* $B \in N_1, N_2, \dots, N_k; B \neq A$.
4. Agent chooses an action $a \in a_1, a_2, \dots, a_7$.
5. The environment executes action a on the connection: $A \xrightarrow{a} B$.
6. The simulation runs to a given stopping criterion (e.g. time elapsed, number of cars finished).
7. The environment passes the agent the reward signal r and the modified state vector x.
8. The agent executes a learning step based on the reward r.
9. Back to step 1.

It's easy to observe that the machine learning model has a particularly difficult job in this case. The action it has to take in each iteration consists of multiple variables, not just a single one: $A \xrightarrow{a} B$. For this reason a highly specialized deep learning configuration is needed. The input vector is the size of the state space as described by the node connections in section 5.2 of this chapter and the number and size of the hidden layers is highly dependent on the size of the input graph, so the rest of this section will deal with the problem of predicting A, B and a. First the *'to'* and *'from'* node prediction will be taken into account. For this there are two proposed architectures, the single and the multi-headed ones.

12.1 Single-head architecture

This is a simple configuration that uses a feedforward network to predict nodes A and B. The input is the state vector, and there are an arbitrary number of hidden layers. This has been denoted as ... in the figure. In the prediction phase of the modeling the model first predicts A, then based on that prediction produces a prediction for graph node B.

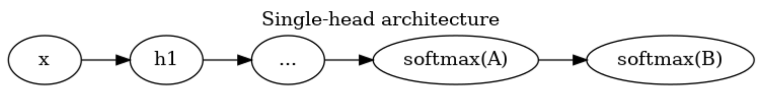


Figure 10:

Both of the predictions use softmax nodes that estimates the probability $\hat{p_k}$ that the action belongs to action class k thereby choosing an action from $a_1, a_2, \ldots, a_7$, where $k \in \{1, 2, \ldots, 7\}$. For k as the number of classes the softmax function is defined as:

$$\hat{p_k} = \sigma(s(x))_k = \frac{exp(s_k(x))}{\sum_{j=i}^{k} exp(s_j(x))}$$

Where $s(x)$ is a vector containing the scores for each action of the current state x and $\sigma(s(x))_k$ is the estimated probability that the state vector x belongs to class k, given the scores for each class of that instance. The final prediction is the class where the probability is the highest.

$$\hat{y} = \underset{k}{argmax} \sigma(s(x))_k$$

In this case the prediction is a node from the set of all nodes. Another important criterion is that $A \neq B$. In the case where the two predictions match the agent should either run the prediction again or receive a negative reward to penalize the weight within the neural network. This research denotes a softmax unit that creates a prediction for class C as $softmax(C)$.

12.2 Multi-head architecture

Another way of predicting the starting and ending nodes is by using two heads on the same neural network. In this case the last hidden node will be in connection with both the softmax neurons:

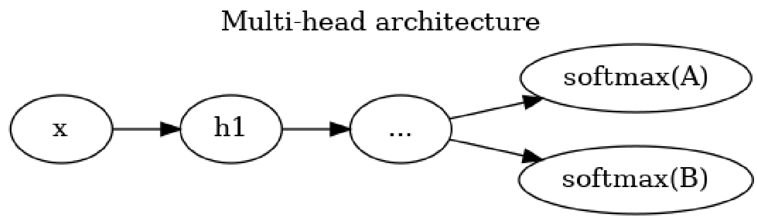


Figure 11:

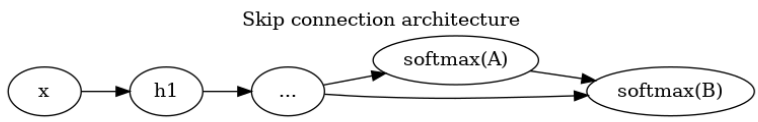
In the same way as before, the two predictions must be different.

12.3 Comparison

This part is dedicated to describing the advantages and disadvantages of the two proposed network architectures. The main idea behind the single-head (SH) architecture is that the neural network has connections between the two prediction nodes, so the B node will be dependent on the A node. This is very useful in learning what junctions are close to each other and are relevant in building the infrastructure and also has a builtin way of learning that the two nodes should be different. However in the SH configuration the B node has no prior information about the state vector as it has already been accumulated in the softmax node. The output of the softmax node is a single class, therefore the B node should be only predicted based on that integer number.

The multi-head architecture (*MH*) can counter this problem: both prediction nodes are in connection with the last hidden state vector, therefore they both have knowledge of the information extracted. This leads to a problem where the two nodes have no connection to each other, hence there is no way to actively penalize predictions where the two outputs match.

A practical realization of this problem is to introduce a skip connection between the last hidden unit and the B node. Using this method the *'to'* node would acquire knowledge about the *'from'* node and the information extracted by the neural network. A high-level architectural diagram of this approach is shown below:

Figure 12: 

12.4 Action-prediction architecture

The last task of the network is to predict what action to take between the nodes A and B. The action denotes if the agent will build or destroy lanes or change the type of the intersection. For this purpose it's necessary that the network has knowledge about the predictions made on the two softmax nodes defined in the previous section. It's also essential that the network can access the state vector transformed by the hidden units and the current state of the graph connection, e.g. how many lanes it currently has and what type of intersection is currently built there. The proposed architecture is as follows:

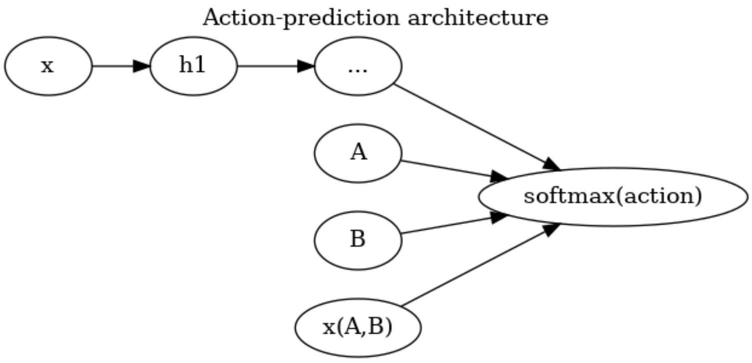


Figure 13:

Where $x(A, B)$ denotes the current state of the graph connection between A, B.

12.5 Practical implementation

If the architecture was made by the method defined before, it would require two separate neural networks, one for predicting A, B and one for a. This would be erroneous in practice as two complete neural networks would have to be trained separately. This approach is redundant and can lead to information being lost and increased computation times. In practice it's beneficial to have a single neural network that trains, and can be able to predict 3 target variables in a single pass. This is also a good approach in the sense that there are dependencies between the variables: $A \twoheadrightarrow B$; $A \twoheadrightarrow a$; $B \twoheadrightarrow a$; $x(A, B) \twoheadrightarrow a$. These dependencies can be brought into the same neural network using the architectural design below:

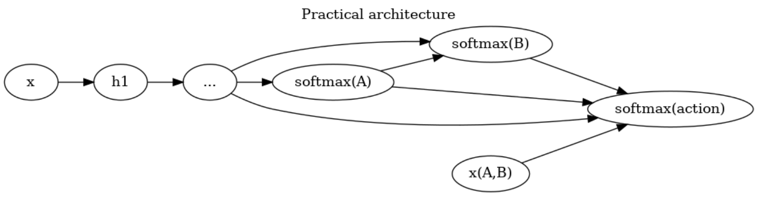


Figure 14:

Where each softmax node serves as a prediction thereby defining $A \xrightarrow{a} B$. This architecture encompasses every dependency between the softmax units under the same scope while being able to provide every prediction. Deciding if the input unit $x(A, B)$ can be left out of the network is up to experimentation in the following sections. This architecture is based on actor-critic reinforcement learning, where the actor predicts the Q-values based on the state vector, this gets input to the critic together with the original state vector to end up with a log probability distribution of the required size. The practical implementation presented here is based on the same idea, the only difference is that AC is 2-step while the practical configuration is a 3-step process.

References

[1] Mohamed A Abdel-Aty and A Essam Radwan. Modeling traffic accident occurrence and involvement. *Accident Analysis & Prevention*, 32(5):633–642, 2000.

[2] National Highway Traffic Safety Administration. National center for statistics and analysis. (2020, december). early estimate of motor vehicle traffic fatalities for the first 9 months (jan-sep) of 2020 (crash stats brief statistical summary. report no. dot hs 813 053). 2020.

[3] Douglas Bates, Martin Maechler, Ben Bolker, and Steven Walker. Mixed-effects models using lme4. *J Stat Softw*, 67(1):1–48, 2015.

[4] John DeCicco, Freda Fung, and Feng An. *Global warming on the road: the climate impact of America's automobiles*. Environmental Defense, 2006.

[5] Conor K Gately, Lucy R Hutyra, and Ian Sue Wing. Cities, traffic, and co2: A multidecadal assessment of trends, drivers, and scaling relationships. *Proceedings of the National Academy of Sciences*, 112(16):4999–5004, 2015.

[6] Neil Hoult, Peter J Bennett, Ivan Stoianov, Paul Fidler, Maksimovi edo, Campbell Middleton, Nigel Graham, and Kenichi Soga. Wireless sensor networks: creating smart infrastructure. In *Proceedings of the Institution of Civil Engineers-Civil Engineering*, volume 162, pages 136–143. Thomas Telford Ltd, 2009.

[7] Susan Jette. Book review: Happy city: Transforming our lives through urban design, by charles montgomery, 2013.

[8] Charles L Marohn. *Thoughts on building strong towns*. CreateSpace Independent Publishing Platform, 2012.

[9] Charles L Marohn Jr. *Confessions of a Recovering Engineer: Transportation for a Strong Town*. John Wiley & Sons, 2021.

[10] C Marshall, KT Boyd, and CG Moran. Injuries related to car crime: the joyriding epidemic. *Injury*, 27(2):79–80, 1996.

[11] John Milton and Fred Mannering. The relationship among highway geometries. 1998.

[12] Peter D Norton. Street rivals: Jaywalking and the invention of the motor age street. *Technology and culture*, 48(2):331–359, 2007.

[13] Peter D Norton. *Fighting traffic: the dawn of the motor age in the American city*. Mit Press, 2011.

[14] Daniel G Parolek. *Missing Middle Housing: Thinking Big and Building Small to Respond to Today's Housing Crisis*. Island Press, 2020.

[15] John Pucher and Christian Lefevre. The united states: the car-dependent society. In *The urban transport crisis in Europe and North America*, pages 175–200. Springer, 1996.

[16] Henrikki Salo-Pontinen. Ai ethics-critical reflections on embedding ethical frameworks in ai technology. In *International Conference on Human-Computer Interaction*, pages 311–329. Springer, 2021.

[17] Aayush Sukhadia, Khush Upadhyay, Meghashree Gundeti, Smit Shah, and Manan Shah. Optimization of smart traffic governance system using artificial intelligence. *Augmented Human Research*, 5(1):1–14, 2020.

[18] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(2):12–18, 2021.