CMPE 491

Senior Design Project I

Dynamic Routing with Ant Colony Optimization Algorithm

By

Burak Güler (11575012)

Tuğberk Göç (115200084)

Supervised

By

Uzay Çetin

Faculty of Engineering and Natural Sciences

Istanbul Bilgi University

May 2020

Table of Contents

Abstract

With the advancement of innovation, our speed of getting to information has expanded. This speed allows human beings to find their solutions much efficient than it used to be. Whilst someone travels one place to multiple places, he/she does not need to search for all the points to find their route. Basically, they want to have a research on a map, then they have discovered what route they must be followed as much as cost effective such as fuel cost, time cost, etc. Researchers found that the big collection of data can processed with mathematics and statistics, there are many algorithms designed to measure shortest path from one to another. This article focuses on algorithms and conducted experimental studies with a different perspective. Aim of the project is to discover the cost-effective root and show this work to the user in a web interface. The flow as follows, user points what places he/she needs to go, then the application shows the route as a map with a proper way.

1. Introduction

The main purpose of this senior design project is to build web application which helps human beings to find their best cost-effective route with using Ant Colony Optimization Algorithm (ACO). Thus, they can see and edit their route in this web application. In addition to that, we are going to be combining ACO with LSTM (Long short-term Memory) which helps us to consider previous inputs based on what we have found with our selection. In this method, we will have a best route optimization for a truck or anything which needs to go more than one places to deliver what it has got.

In this web application, it is important to use Django, which is Python web framework on the back side, Vue.js on the front side. They are both very powerful to manage complex algorithms. Also, we are going to implement server-side rendering for the front side which provides fast-rendering for the end user.

First of all, we have determined which topic we are going to focus on as groupmates. We had several conversations with our supervisor and surmised that solving dynamic route optimization with ant colony optimization will be truly beneficial for daily life, especially for crowded and large areas. Then, we have planned our schedule during the whole semester about how we are going to proceed step by step. We have strained at following our organized plans and started to set up the mentioned hardware and software technologies for our senior design project. Also, we have successfully completed making literature review and become sufficient to make illustration and prepare our first report for the project.

In this report, we are going to cover our topic with our methods, tools and resources exhaustively.

2. Methods

In this part, methods that have been chosen explained in detail.

2.1 Ant Colony Optimization

Ant colony optimization is an algorithm of probabilistic technique that tries to solve computational problems through artificial ants finding best paths on graphs. This algorithm is one of the swarm intelligence methods that takes inspiration from the social behaviors of the several livings. The past of science shows that ants are extremely significant for inspiring lots of successful techniques, ant colony optimization is a strict proof of it.

The main idea of the ant colony optimization is based on foraging action of ant species. While flocks of ants are looking for food nearby, they deposit a chemical on the ground called pheromone to signal the other members of the colony for following the related track for the food resources. The usage of the ant colony optimization is basically born for this behavior of the ants in order to solve optimization problems.

Ant colony optimization is firstly approached by Marco Dorigo for his doctoral dissertation to point the basic principles of finding best route optimization in 1992. He has named this algorithm "Ant System", calculated the rate of convenience of the parameters of algorithm and experimented on various extensions of Traveling Salesman Problem, managed on solving the small scaled ones. Thereupon, ant colony optimization has been come into use by many researchers and become one of the prevalent artificial intelligence techniques for optimization at the present time.

2.1.1 The Biological Inspiration of Ant Colony Optimization

J.L Deneubourg have researched the behavior of the placing pheromone and following ants. This experiment includes two bridges that connect an ant nest with a food source. Deneubourg created two different experimental conditions in terms of the length of the bridges, the first experiment has two bridges of equal lengths [See Figure 1.a] and the second one has two bridges of various lengths [See Figure 1.b], the one is much longer than the other. When he considered the first experiment, the ants initially made randomly choices for their path and deposited pheromone onto the ground. Still, because of stochastic waving of the actions, one of the two equal bridges had higher rate of pheromone after some time and became more attractive for the rest of the ant colony members. This situation led to farther amount of pheromone on that bridge and all ants started to prefer to go to the food source through the same bridge in progress of time.

When S. Goss, the colleague of Deneubourg, considered the second version of the experiment in that one bridge is clearly longer than the other, the impact of the random fluctuations was reduced. This is because the ants, who randomly selected the short bridge, arrived earlier than the ants on the long bridge. The short bridge perceived the pheromone initially, thus, this showed that the probability that the remaining ants select the short bridge rather than the long bridge is increased. Then, the researchers have developed a model for the observed action of the ants: assuming that at the given moment in time, the number of ants that used the first bridge is m_1 and the number of ants that selected the second bridge is m_2, the probability p_1 for an ant to choose the first bridge is:

$$p_1 = \frac{(m_1+k)^h}{((m_1+k)^h+(m_2+k)^h)},$$

where parameters k and h are some experimental data, and definitely, $p_2 = 1 - p_1$. For experimental parameters, $k \approx 20$ and $h \approx 2$ values seemed acceptable fit by Monte Carlo simulations.

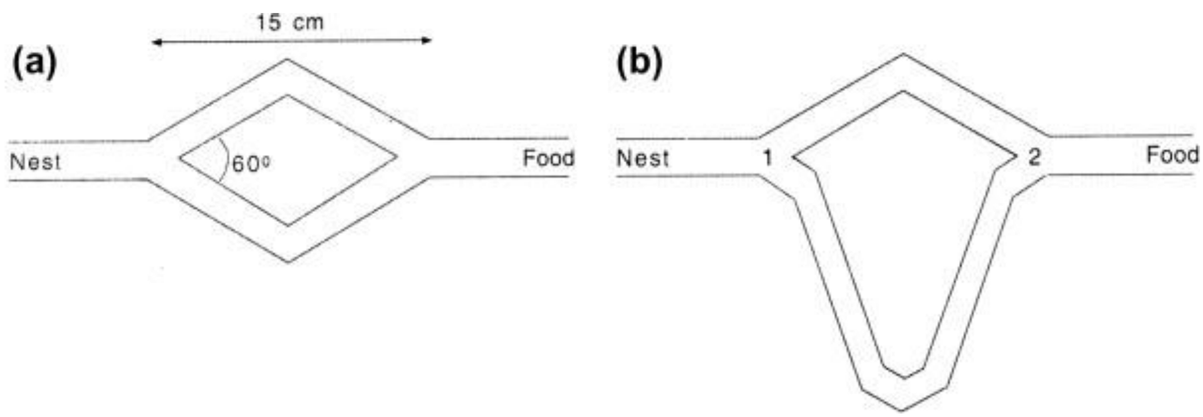


Figure 01: Experimental setup for the double bridge experiment. (a) Branches have equal lengths. (b) Branches have different lengths.

2.1.2 The Optimization Technique of ACO

Since the model developed by Deneubourg and his co-workers about the foraging action of the ants mainly inspired the structure of ant colony optimization, various ant colony optimization have been proposed by researchers. The initial and the original version of the ant colony optimization algorithm is proposed by Marco Dorigo as “Ant System” in early nineties. Although all ant colony optimization algorithms share the same idea basically, each of them can be applied to solve different problems depending on how the algorithm fits on the problem in terms of parameters and usability.

2.1.2.1 Ant Colony Optimization for Traveling Salesman Problem

When we consider the traveling salesman problem, we have the set of cities and the distances between each other as parameters. The main purpose is to find the shortest path satisfying that each city is visited for only one time through a connected graph, this procedure is also called Hamiltonian tour. When ant colony optimization is implemented for this problem, vertices and edges are operated as cities and connection between cities, respectively. The simulation allows several ants to move around this connected graph and deposit a chemical on the ground, pheromone. Since ant colony optimization is an iterative algorithm, the information about the concentration of the pheromone on the related path can be perceived by the remaining ants. At each iteration, ants make a stochastic choice based on the pheromone level and create a route by moving vertex to vertex with the condition of not visiting the ones that they have already visited. In other words, after some time, ants select their following path to the unvisited vertex according to the deposited pheromone rate. Obviously, the pheromone level is proportional to the probability of the selection of the edges. When iteration ends, a solution created by the ants is generated and the remaining pheromone values on the roads affect the ants for a better solution for the future.

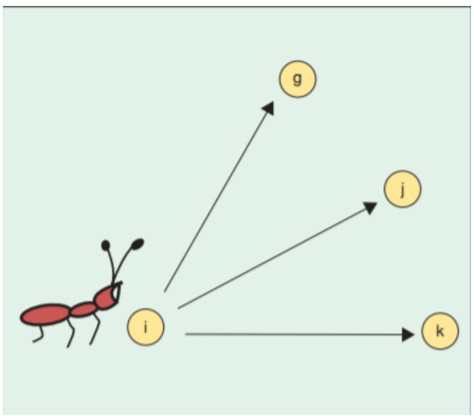


Figure 02: Representation of an ant in city i choosing the next unvisited city to visit depending on the pheromone level on the edges.

2.1.2.2 Main Ant Colony Optimization Algorithms

I) Ant System (AS)

Ant System is the first ant colony optimization in the literature designed by Marco Dorigo. What distinguishes it from the others is that the pheromone values are always updated by the m number of ants in the iteration. The calculation of the current pheromone value on the edge between cities i and j, τ_{ij} is:

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k},$$

where ρ is evaporation rate of pheromone, m is the number of ants on the edge and is the current quantity of the pheromone on the associated edge, dropped by ant k:

$$\Delta\tau_{ij}{}^{k} = \begin{cases} Q/L_K, & \text{if ant k passed on this edge} \\ 0, & \text{otherwise} \end{cases},$$

where Q is a constant, L_k is the length of the ride completed by ant k. Since ants make their selection in a stochastic way, the probability that ant k in city i moves to city j is:

$$p_{ij} = \frac{(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\sum(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})},$$

where η_{ij} equals to $1/d_{ij}$, where d_{ij} is the distance between cities i and j, refers to the desirability of the edge i, j.

II) MAX-MIN Ant System (MMAS)

The second of the most successful ant colony optimization algorithms is MMAS which is the developed version of the Ant System. Its major differences from Ant System are that the pheromone value is updated only by the best ant and the minimum and maximum values of the pheromone are explicitly bounded. The pheromone update can be shown as:

$$\tau_{ij} \leftarrow \left[(1-\rho).\tau_{ij} + \Delta\tau_{ij}{}^{best}\right]_{\tau_{min}}^{\tau_{max}},$$

where τ_{max} and τ_{min} are upper and lower bounds imposed on the pheromone, respectively.

III) Ant Colony System (ACS)

Ant Colony System is another type of the beneficial ant colony optimization algorithms. A local pheromone update is firstly mentioned in this algorithm in addition to the pheromone update executed at the end of the constructing process, called offline pheromone update. It is performed by all the artificial ants after each construction step. Each member of the ant colony performs it to the last edge traversed:

$$\tau_{ij} = (1 - \varphi).\tau_{ij} + \varphi.\tau_0,$$

where τ_0 is the initial pheromone value and $\phi \in (0,1]$ representing pheromone decay coefficient.

The main characteristic of the local pheromone update is diversifying component against exploitation. In other words, it reduces the pheromone level on the traversed edges, it conduces that ants embolden remaining ants to select other edges and to produce possibly better solutions. This prevents many ants to create identical solutions that would decrease the productivity of the algorithm.

The offline pheromone update is executed by only one ant at the end of each iteration, similar to MMAS. It has two options; either the iteration-best or the best-so-far, but the mathematical expression is some different from MMAS:

$$\tau_{ij} \leftarrow \begin{cases} (1 - \rho).\tau_{ij} + \rho.\Delta\tau_{ij}, & \text{if i, j belongs to the best tour} \\ \tau_{ij}, & \text{otherwise} \end{cases}$$

Also, in ACS, the decision rule is applied by the ants during the iteration that makes a difference between ACS and AS. In other words, a method called pseudorandom proportional rule is executed, it means that the probability for an artificial ant moving from city i to city j depends on an arbitrary variable q homogeneously distributed over [0,1], and a parameter q_0: If $q \leq q_0$, the component that maximizes the result $\tau_{il}\eta_{il}^{\beta}$ is chosen, otherwise it is applied as the Ant System.

2.2 LSTM (Long short-term Memory)

LSTM is a very popular algorithm which transfers past knowledge to the future. This part is covered LSTM from scratch to its real mechanism. (Olah 2015)

2.2.1 Understanding RNN Mechanism

LSTM is a very special type of Recurrent Neural Network (RNN) which works properly for many tasks, much better performance comparing to the RNN standard version. Practically all exiting outcomes dependent on recurrent neural networks are accomplished with them.

Before LSTM, it is essential to explore recurrent neural networks. People don't begin their thinking without any preparation consistently. Whilst you read this report, you see each word dependent on your comprehension of previous words. You don't discard everything and begin thinking without any preparation once more because your ideas and thoughts are persistent. Traditional neural networks don't have this kind of mechanism, and it appears to be a significant deficiency for them. For instance, imagine you want to prioritize what kind of situation is happening at every single moment in a documentary. It is clear that without previous situations, it is hard to guess lion is going to hunt some animals in the forest. Thus, it's hazy how a traditional neural network could utilize its thinking about past occasions in the documentary to advise later ones. That's why RNN address these kinds of issues. They are networks with loops in them, and they make information to be persistent.

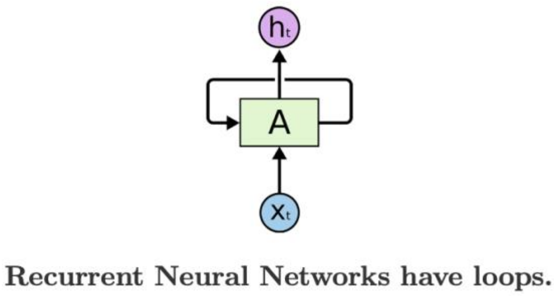


Figure 03: Recurrent Neural Network with a loop

As you can see above, one of the chunks of neural network shown as A, the input value x^t, and the output value is h^t. They are neural network systems with loops in them, permitting data to be passed one to another chunk.

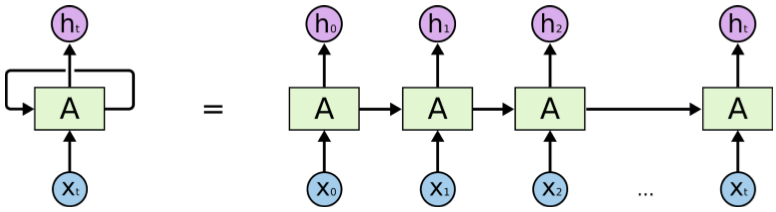


Figure 04: An Unrolled Recurrent Neural Network

This chain-like nature uncovers that repetitive neural systems are personally identified with successions and records. They're the common engineering of neural system to use for such information. Over the most recent couple of years, there have been inconceivable achievement applying RNNs to an assortment of issues such as speech recognition, language modeling, translation, image captioning. Essential to these achievements is the implementation of LSTMs, which is a very successful kind of recurrent neural network for many tasks, and its performance much better than the standard version of the RNN.

2.2.2 LSTM Networks

Long Short-term Memory networks are special kind of RNN, and they are able to learn long-term dependencies. They were presented by Hochreiter & Schmidhuber (Hochreiter, Rovelli, and Winckler 1997), and they have been used by many people thanks to its successful results on a huge assortment of issues comparing to the classic RNN algorithms. Thus, it is the most popular type of RNN. Their main design purpose is to keep away from the long-term dependency issue. Basically, remembering data for a long term is their default action.

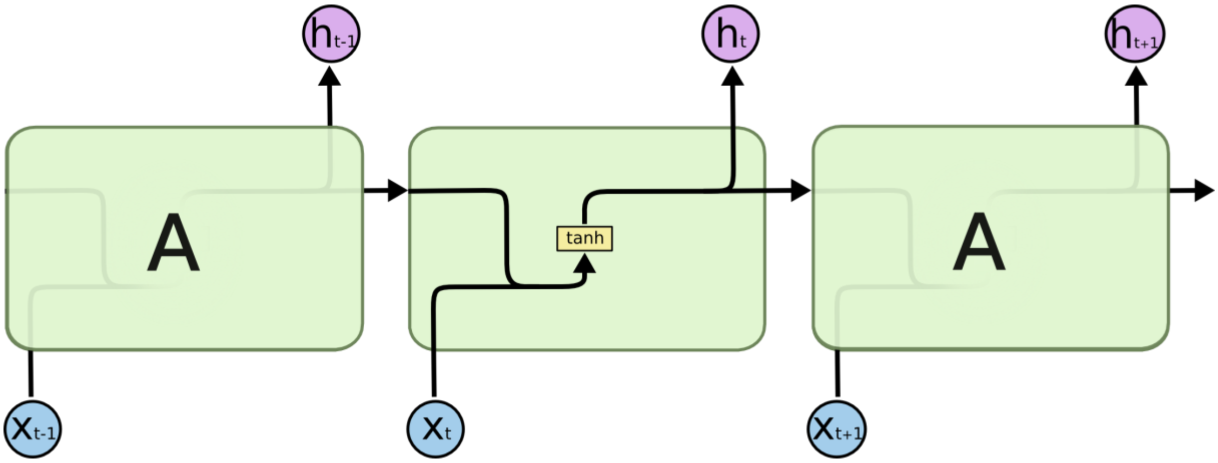


Figure 05: The repeating module in a standard RNN contains a single layer

As you can see above, it is a type of recurrent neural network which is repeating itself with a very simple function, such as a single $tan(h)$layer. LSTMs also have these kinds of chains, but they have a different structure which is more complex than the standard one such as having four neural networks instead.

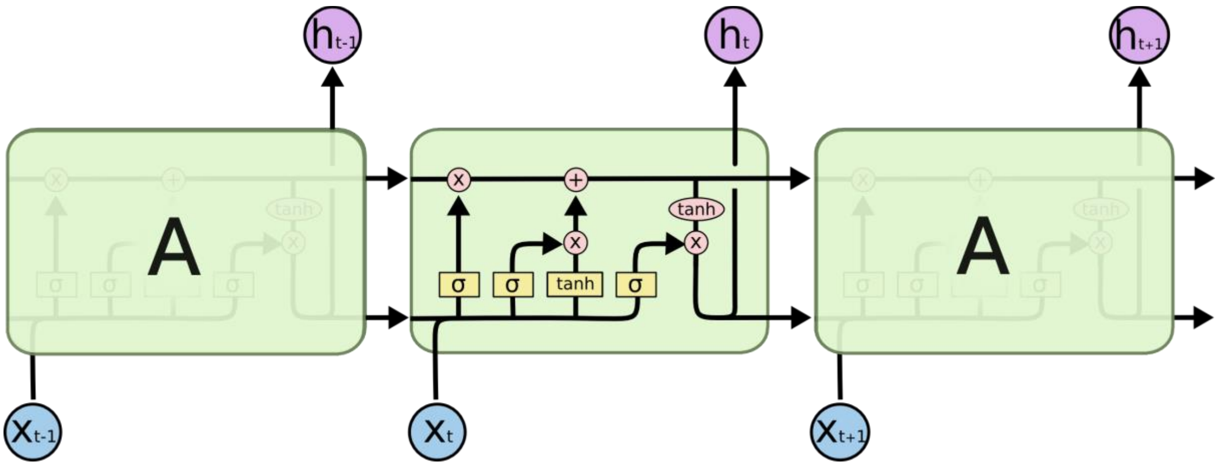
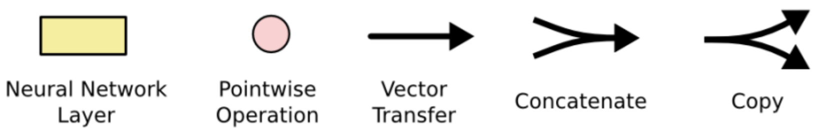


Figure 06: The repeating module in LSTM contains four interactions layer

The pink circles represent pointwise operations, such as vector summation, concatenation, division, and multiplication, and the yellow rectangles represents the neural network layer. You can see all the representations as below.



Let’s deep dive into to the LSTM layers. The main layer is the cell state which locates horizontally to the top of the diagram. It takes the information, and transfer with applying some minor change to flow through it unchanged.

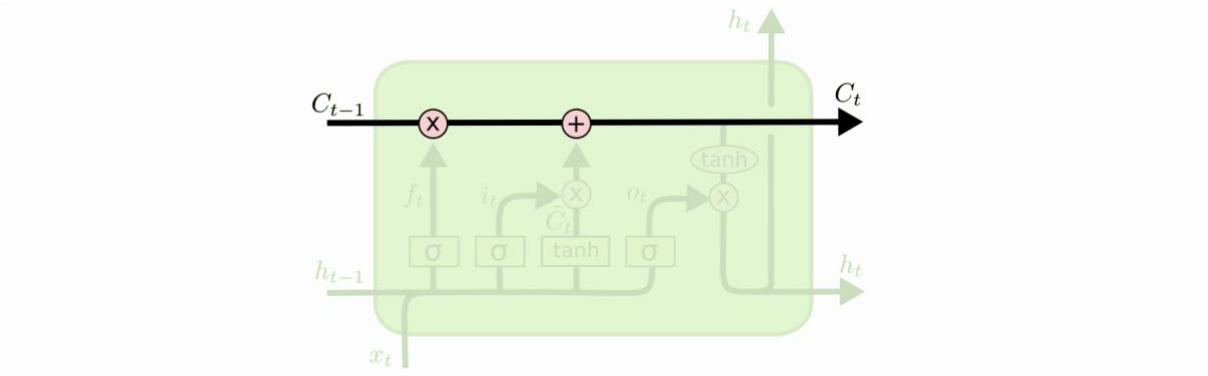


Figure 07: Cell state in LSTM

LSTM contains three gates such as input gate, forget gate and output gate. All of them consists of a sigmoid function, this function takes the input, and it produces double value between 0 and 1. This value represent how much each input needs to be included. A value of zero stands for including nothing, and a value of one stand for including everything. Thus, adding or removing information to the LSTM is done by sigmoid function.

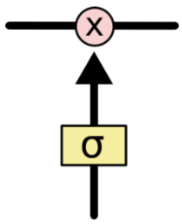
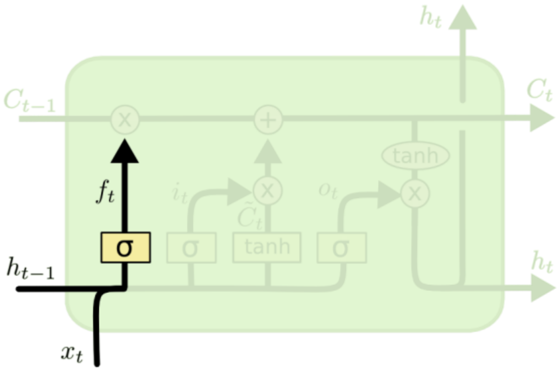


Figure 08: Sigmoid Function in LSTM

I) Forget Gate Layer

This gate decides what information need to pass or thrown away by using sigmoid function. As you can see below, it has $h_{t-1}$and $x_t$as inputs, and $f_t$as output. If the output value is 1, $C_{t-1}$is taken completely, but if it is 0, then it is just thrown away.

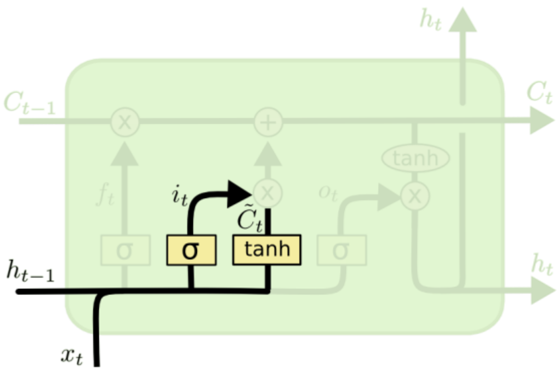


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 09: Forget gate layer in LSTM

II) Input Gate Layer

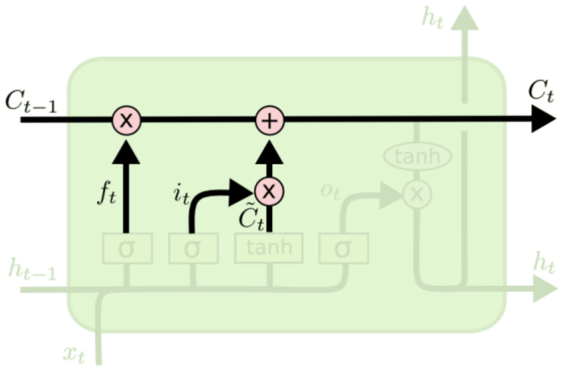
This gate decides what information need to pass to the future input, so it stores the given information to use in the future. Also, it consists of two parts. One of them is sigmoid layer, and another one is $tanh$ layer which produces a vector of new candidate values, $\widetilde{C}_t$, that might be used to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 10: Input gate layer in LSTM

Now, it is time to use what has been found earlier to find the previous cell state, C_{t-1}into the next cell state C_t. The calculation is made by using the function below in order to find next cell state. In this calculation, $f_t * C_{t-1}$decides the information needs to be thrown away or not using previous cell state, and $i_t * \widetilde{C}_t$ decides how much needs to update each state value.

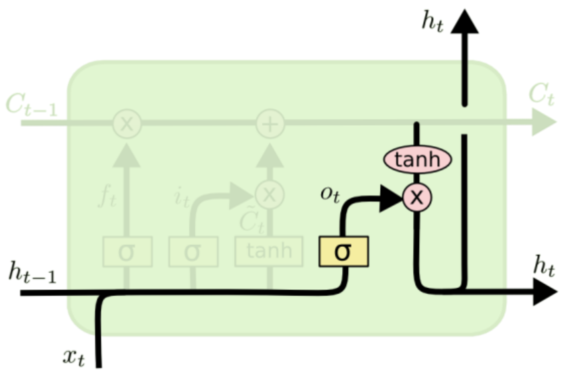


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 11: Updating the cell state in LSTM

III) Output Gate Layer

It is time to decide what the system is going to output. This output will depend on current cell state, but filtered version. In this layer, sigmoid function decides how much cell state needs to be included as output. Then, $tan(h)$function gets the current cell state, and it produces values between -1 to 1 in order to multiply with the output of the sigmoid function that has been found earlier.



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 12: Output gate layer in LSTM

2.3 Dynamic Route Optimization

In this modern world, traffic conditions are getting worse such as increased wasted hours, fuel costs, and travel unreliability. These are the problems that intelligent systems are trying to find solutions. For instance, they calculate waiting time for each traffic light based on their traffic condition, their length between red to green and their historical data.

For solving these issues, many researchers have been working to find a traffic simulator approach. One of the researches has found a solution for that (Kim, 2016). In this solution, we can keep track of the current location of the car with using some mathematical calculations.

$$v = \begin{cases} v_{\max} & \rho \leq \rho_{\mathrm{th}} \\ \dfrac{v_{\min} - v_{\max}}{1 - \rho_{\mathrm{th}}} \left(\rho - \rho_{\mathrm{th}}\right) + v_{\max} & \text{otherwise,} \end{cases}$$

In this calculation, v is the velocity of the car where, v_{max} and v_{min} represents the speed limit, and the minimum speed limit of the segment, and p represents the ratio between number of current vehicles, where $p_{th}$is a constant between 0 and 1. In addition to that, road segment (N_{max}) can be found as following.

$$N_{max} = \frac{d * L}{l_{car} + h} \text{ where;}$$

d: length of the road segment

L: number of lanes

l_{car}: average length of a vehicle

h: average headway

3. Similar Works and Comparison (Related Works)

As you can see in the following chart, similar works and comparison shows as a table, this table shows all the features availability for each one of the similar works. SV means standard version which is open to use to the users.

FEATURES	**Route4me**	**Google Map (SV)**	**Yandex Map (SV)**
Multiple Location Path	**Yes**	**No**	**No**
Traffic Condition	**No**	**Yes**	**Yes**
Fuel Cost Consideration	**No**	**No**	**No**
Mobile App Support	**Yes**	**Yes**	**Yes**
Real Time Management	**Yes**	**No**	**No**

Route4me is a cloud-based management tool for transportation businesses. It provides you to find the route in multiple stops, and if something changes in a route, it updates itself automatically. User can see their route in their mobile phone with using application. And, the owner of the company can track all the trucks in a map. Google Map provides users to find their best optimized path one place to another with showing in graphical interface. It is the most common map in the world. Yandex map does the same thing with what google map does.

4. Timeline

Timeline of the project can be seen as below.

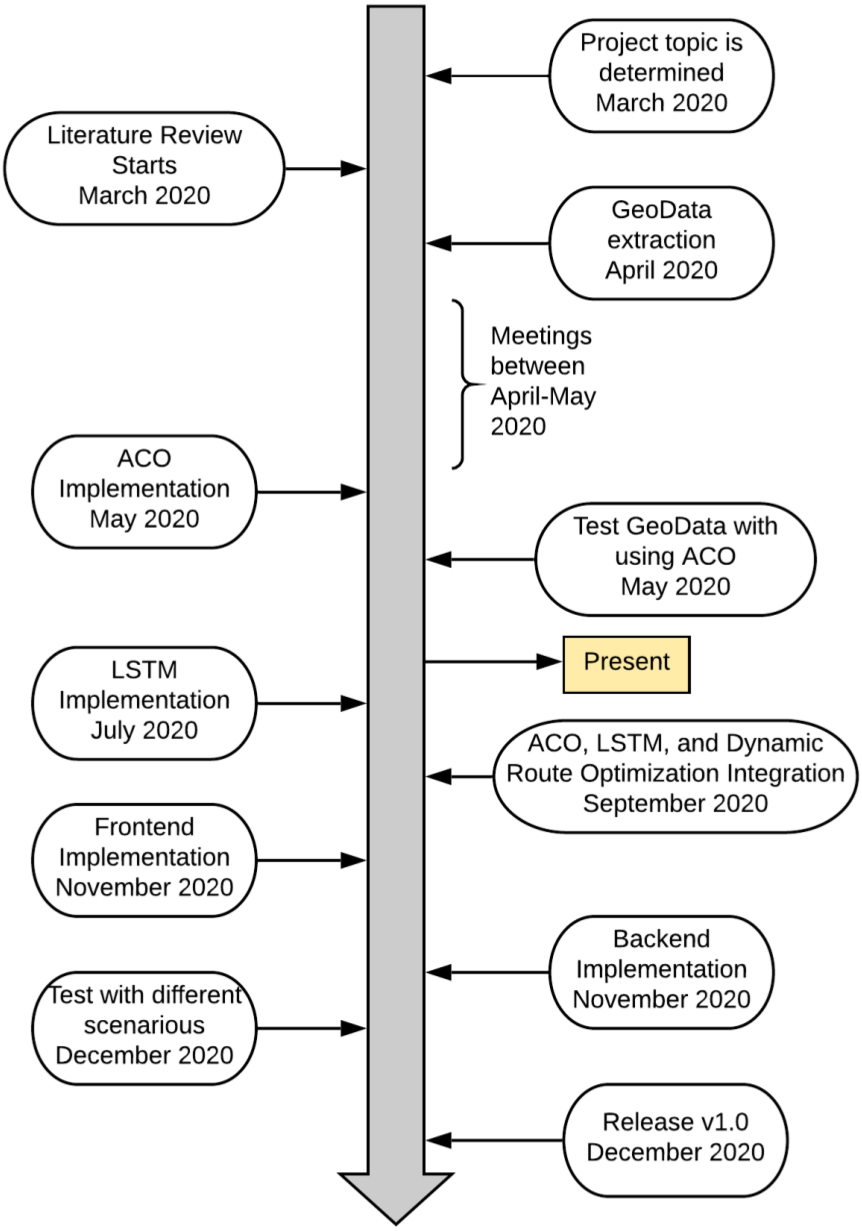


Figure 13: Timeline

5. Risk Analysis

Risk analysis is made by using risk matrix. In this matrix, level of impact represents how many degrees affect to the product, and likelihood represents how much this risk likely to be happened. Risk matrix are defined as below. Green color represents low-level risk, yellow color represents medium-level risk, and red color represents high-level risk.

Figure 14: Risk Matrix

6. Aim and Importance of the Project

With the increasing amount of supply and demand curve, the need of the companies for delivering products has become an essential. However, when it is considered including all modern requirements in the daily life, it becomes extremely difficult to find time and cost efficient solutions for each one of the problems. This causes to serious complexity for companies and clients in terms of time and materiality. For instance, a shipping truck with several routes to visit may have huge difficulties in metropolitan. This is because traffic or irregularity of the roads might negatively affect its performance. Hence, these issues may cause companies to make a loss, and clients to fail satisfying their requirements.

The main goal of our project to solve this problem, providing a web application which shows users to find their best cost-effective route according to their complicated way. Especially in big companies, it can make a huge impact such as reducing their fuel cost, using time effectively for the employee, satisfying the customers. This may lead them to move into profit more and get rid of one of the important issues in the business life.

7. Design of the Project

It is time to explore design of the project step by step.

7.1 Dataset

The dataset is taken from OpenStreetMap with using osmnx library (Boeing 2016) which helps you to clear unnecessary data such as unnecessary nodes or edges. Nodes refer to breakaway points and edges refer to paths. As you can see below, there is an example of Istanbul graph, in this graph, blue represents nodes and gray represents edges. It has more than five hundred thousand edges, and more than 2 hundred thousand nodes. Thus, calculating shortest path one to another costs a lot of memory and CPU power. So, it is necessary to start with small graphs which includes less nodes and edges.

Figure 15: Graph of Istanbul

7.1.1 Osmnx Library

Osmnx library is created by Geoff Boeing, and a lot of developers have helped him to develop this library. Nowadays, it has more two thousand stars (See Appendix B), so it is very popular in retrieving, modeling, analyzing and visualizing street networks. It uses OpenStreetMap api to create these kinds of networks. It automatically downloads the graph which using boundaries (latitude and longitude), country or place name and shapefiles which is polygon or etc. When the map is downloaded, it constructs street networks as simplified and corrected network topology. Then, it can be saved as shapefiles, GraphMl or SVG to the disk. As you can see below figure, Istanbul Graph is downloaded and saved to the disk with an extension as GraphMl and a given name which is ‘istanbulfull.graphml’.

```
import osmnx as ox
# Gets the map from OpenStreetMap, and save it to the main folder
print("The map is downloading and Graph is creating...")
G = ox.graph_from_place('Istanbul, Turkey', network_type='drive', which_result=2)
ox.save_graphml(G, filename='istanbulfull.graphml')
print("Graph Saved")
```

Figure 16: Gets Istanbul Map as Graphml

It is essential to start small things to grow incrementally. So, it is useful to create a small place graph and work on it. That’s why, we need to have one point (latitude and longitude) in a map that is known by a lot of people. In this way, it is easy to simulate common situations as a real example. For that, there is an osmnx method which is called ‘graph_from_point’, and it takes point, distance and network type. If we give a specific point in Akaretler, Besiktas, a distance which is 1 kilometer and network type as a driving network, we have a map that includes a certain area close to that specific point. As you can see in the figure below, we have a small graph which includes 423 nodes and 919 edges.

```python
import osmnx
G = ox.graph_from_point((41.042915, 29.007339), distance=1000, network_type='drive')
ox.plot_graph(G)
print('This map includes ' + str(len(G.nodes)) + ' nodes.')
print('This map includes ' + str(len(G.edges)) + ' edges.')
```

```
This map includes 423 nodes.
This map includes 919 edges.
```

Figure 17: Akaretler Area Map

7.1.2 Openstreetmap

Openstreetmap is created by Steve Coast in 2004 (*OpenStreetMap - Wikipedia* 2004). It is a collaborative project which means human beings join this society, and they share their geographical information about where they live. End of the day, all geographical data are gathered as an open source map of the world, and they use in many areas such as creation a network, finding shortest path, showing the route, even in using this information for big data. This data is called as geodata which contains type, generator, copyright, timestamp and features. The most important part is the feature part because it shows the type of it as line string with its given coordinates by people. (See Appendix C)


```
▼ object {5}
    type : FeatureCollection
    generator : overpass-ide
    copyright : The data included in this document is from www.openstreetmap.org.
                The data is made available under ODbL.
    timestamp : 2020-04-16T02:44:02Z
  ▶ features [913]
```


Figure 18: Sample GeoData

In this senior design project, Osmnx library help us to clean and eliminate unnecessary nodes and edges, and it makes sense of them. It uses OpenStreetMap in the background to provide data itself.

7.2 Folder Structure

As you can see below figure, folder structure is created as a project boilerplate, also called as project starter which includes all configuration and files are needed to run.

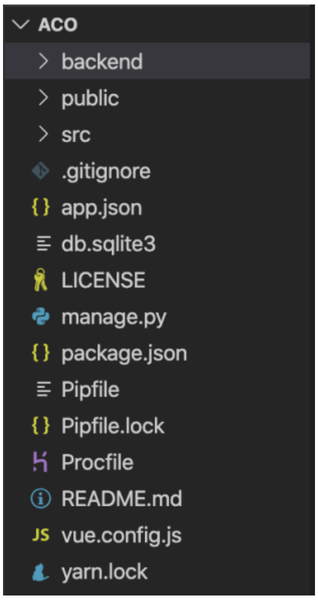


Figure 19: Folder Structure

7.2.1 Backend

Django manages all the web application logical part on the back side such as forwarding requests to the appropriate view, reading and writing data, etc. In addition to that, it is designed as much as possible to be secure with its own web server. So, it is not necessary to worry about securing the website against others. Also, it is well established which means it has its own admin panel in order to manage all permissions. It works with most major databases.

It is designed as MVC pattern which includes Model, View and Controller. In model, there are a lot of queries that connects to the database, and it reads and writes the data to the table. The database is created with sqlite3 which is integrated to the Django framework properly.

7.2.2 Database Model

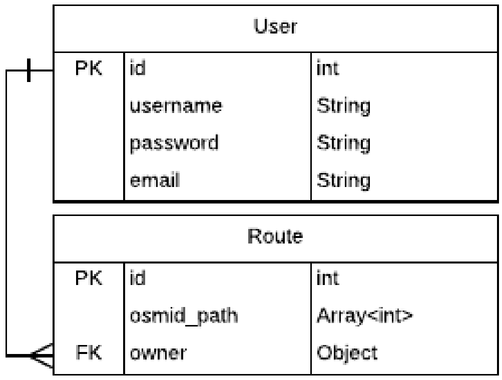


Figure 20: Database Model

As you can see above, the database model includes User and Route tables. User table contains user information which are their username, email and password. In the Route table, we have been storing their path with using osmid values which are nodes values in graphml file. They are stored as integer array. The database model is ready to upgrade next level. It might change according to the product needs.

7.2.3 Frontend

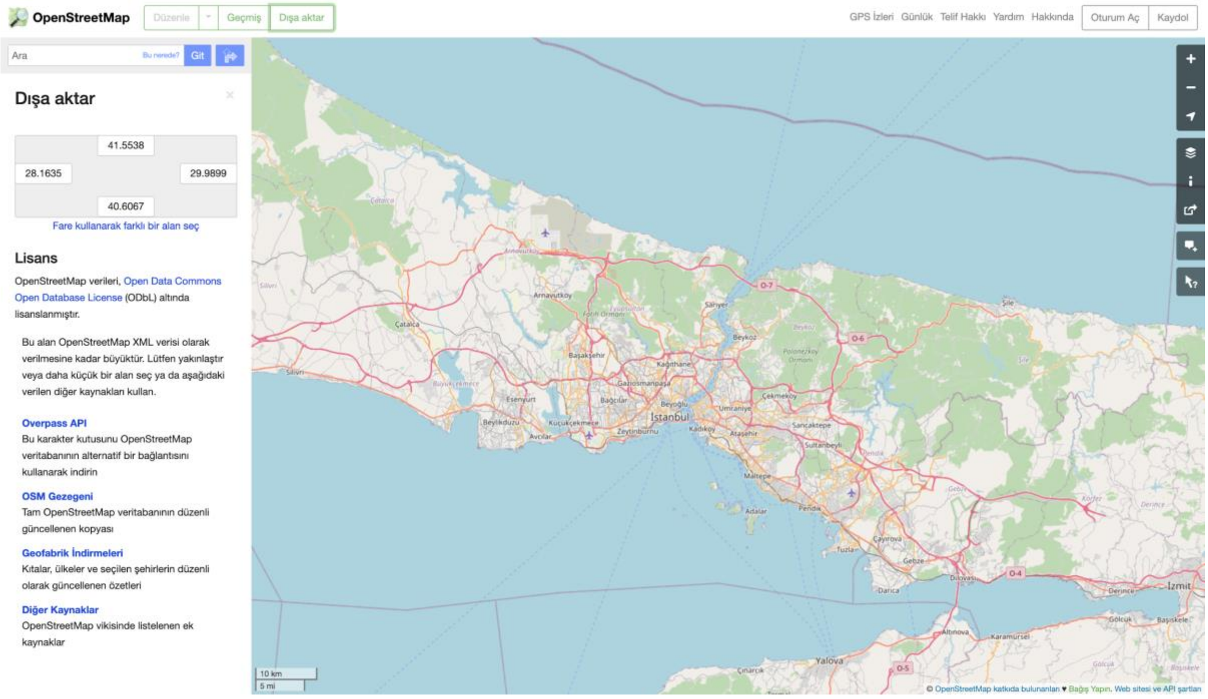
On the front side, VueJs framework (Vue.Js n.d.) is really powerful to handle complex systems. It has its own state management, routing system. Also, it is component-based framework which divides the DOM into the components. Thus, it renders html pages faster than other frameworks. In addition to that, there is another framework which is called NuxtJs (*Nuxt/Nuxtjs.Org* 2020). It works with VueJs. It helps rendering these pages on the server side, then it sends complete html page without needed extra effort to render such as compiling js files, css or sass files.

VueJs framework is a java-script framework. That’s why it uses npm package manager behind the scene, so it has a package.json file which contains all the packages with their versions. This help to manage all the dependencies on one hand.

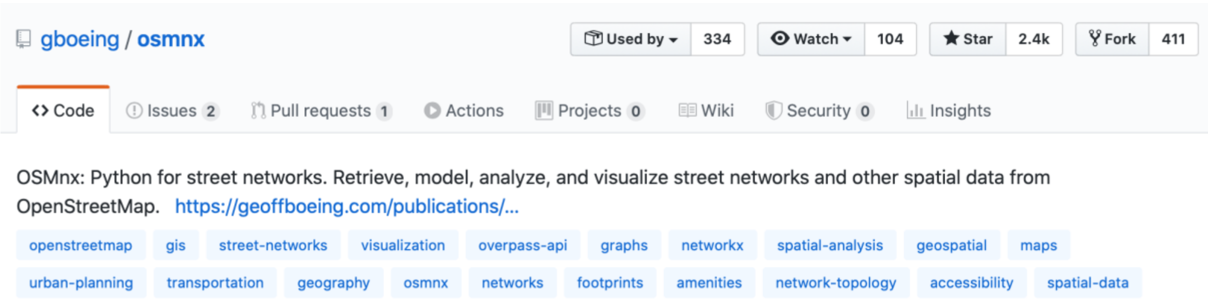
8. Conclusion

This article focuses on algorithms that are related to finding route with several artificial intelligence techniques such as ant colony optimization and LSTM. Aim of the project is to bring the best cost-effective route according the what user needs and show this route as map to the user with a web interface. The steps that are needed to be followed; user enters its source coordinates and its target coordinates as a multiple choice, then the map shows the route that he/she needs to follow. The expectation of people for solving daily routing problems is quite high, this project aims making a contribution to the social life of the humanity.

Appendix A



Appendix B



Osmnx Library Repository on Github

Appendix C

The below figure shows what feature data contains in geodata.


```
▼ features [913]
  ▼ 0 {4}
      type : Feature
    ▼ properties {3}
        @id : way/4406227
        highway : living_street
        name : Celal Atik Sokağı
    ▼ geometry {2}
        type : LineString
      ▼ coordinates [10]
        ▼ 0 [2]
            0 : 28.9952791
            1 : 41.0578814
        ▶ 1 [2]
        ▶ 2 [2]
        ▶ 3 [2]
        ▶ 4 [2]
        ▶ 5 [2]
        ▶ 6 [2]
        ▶ 7 [2]
        ▶ 8 [2]
        ▶ 9 [2]
      id : way/4406227
```

REFERENCES

Boeing, G. (2016) *OSMnx: Python for Street Networks* [online] available from <https://geoffboeing.com/2016/11/osmnx-python-street-networks/>

Dorigo, M. (2007) "Ant Colony Optimization". *Scholarpedia* [online] 2 (3), 1461. available from <http://iridia.ulb.ac.be/~mdorigo/Published_papers/All_Dorigo_papers/DorDic1999cec.pdf> [6 December 2019]

Dorigo, M., Birattari, M., and Stützle, T. (n.d.) *Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique* [online] available from <https://courses.cs.ut.ee/all/MTAT.03.238/2011K/uploads/Main/04129846.pdf>

Ergüder, H. (2019) *Recurrent Neural Network Nedir?* [online] available from <https://medium.com/@hamzaerguder/recurrent-neural-network-nedir-bdd3d0839120> [24 May 2020]

Hochreiter, E., Rovelli, R., and Winckler, G. (1997) "Centrální Banky a Využití Ražebného? Studie Tří Transformujících Se Ekonomik". *Politická Ekonomie* [online] 45 (2), 193–206. available from <http://www.bioinf.jku.at/publications/older/2604.pdf>

Kim, Kwangsoo & Kwon, Minseok & Park, Jaegeun & Eun, Yongsoon. (2016). Dynamic Vehicular Route Guidance Using Traffic Prediction Information. Mobile Information Systems. 2016. 1-11. 10.1155/2016/3727865.

Nuxt/Nuxtjs.Org (2020) available from <https://github.com/nuxt/nuxtjs.org> [27 May 2020]

Olah, C. (2015) *Understanding LSTM Networks -- Colah's Blog* [online] available from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

OpenStreetMap - Wikipedia (2004) available from <https://en.m.wikipedia.org/wiki/OpenStreetMap> [30 May 2020]

Stutzle, T. and Dorigo, M. (2002) “A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms”. *IEEE Transactions on Evolutionary Computation* 6 (4), 358–365

The Web Framework for Perfectionists with Deadlines | Django (2019) available from <https://www.djangoproject.com/>

Vue.Js (n.d.) available from <https://vuejs.org>