

Substantial Utilization of MST to Reduce Taxi-Delay in the Metropolitan City of Johannesburg

Mike Nkongolo

School of Computer Science and Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
Mike.nkongolo@damelin.co.za

Laby ILumbe

Department of Transport and Supply Chain
University of Johannesburg
Johannesburg, South Africa
labyi@uj.ac.za; labyilumbe@gmail.com

Abstract

This paper presents a design of solving Taxi-Delay during peak hours utilizing techniques from graph theory. The Minimum Spanning Tree algorithm (denote MST) is efficient for searching optimal transport routes between locations. The intuition of the MST problem is that it depicts different sort of scenarios where it is crucial to use this theoretical approach and how to use this tool for discovering a solution. Graph theory incorporates various algorithms for searching the minimum spanning tree and this paper uses one of them. To model the situation of Taxi-Delay optimization we use connected weighted graph where vertices represent locations and the edges represent the transport routes between the locations. The weight of an edge represents the time to drive the Taxi between two locations. A theoretical discussion and a model example are carried out to assess the MST.

Keywords:

Discrete Optimization, Graph Theory, Decision Tree, TTM, Queuing Theory.

1. Introduction

According to Thomson [16] and Ang [1], Transport Traffic Management (TTM) is a logistics sub field, interested in the planning, control and purchase of transport services necessary for the physical movement of vehicle. This is depicted in Figure 1 bellow. In metropolitan cities such as Johannesburg TTM requires special attention because of the time it takes for a taxi to reach a specific destination. Many people have experienced this fact, especially in the city of Johannesburg.

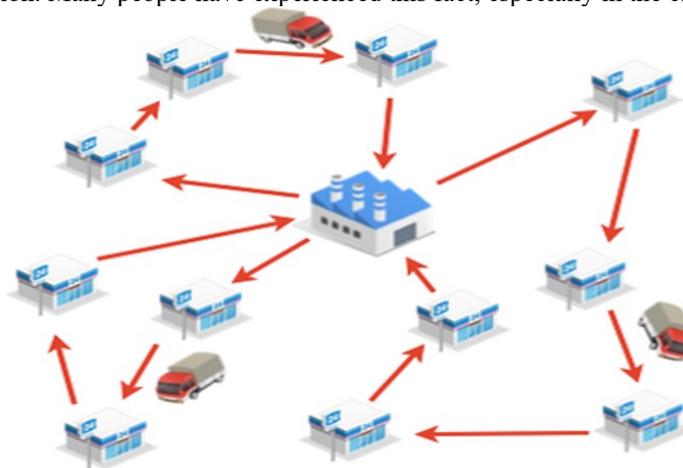


Figure 1. TTM

In this case, it would be imperative to study the existing possibilities that could allow us to optimize the time needed for a taxi driver to move from one location to another. For instance, if a driver is in Braamfontein what is the minimum time he would take if he wants to go to Sandton? Note that there are several roads from Braamfontein to Sandton, this implies that the time varies from one location to another as depicted in Figure 2. In Figure 2, De Villiers, M9 and Jan Smuts avenue are three different routes which the taxi driver can select to reach Sandton. According to Google map, De Villiers is the optimal route; unfortunately, it is not always true because Google map cannot always yield optimal results. For example, taking Jan Smuts avenue, the taxi driver could reach Sandton and the time could be less than 20 minutes.

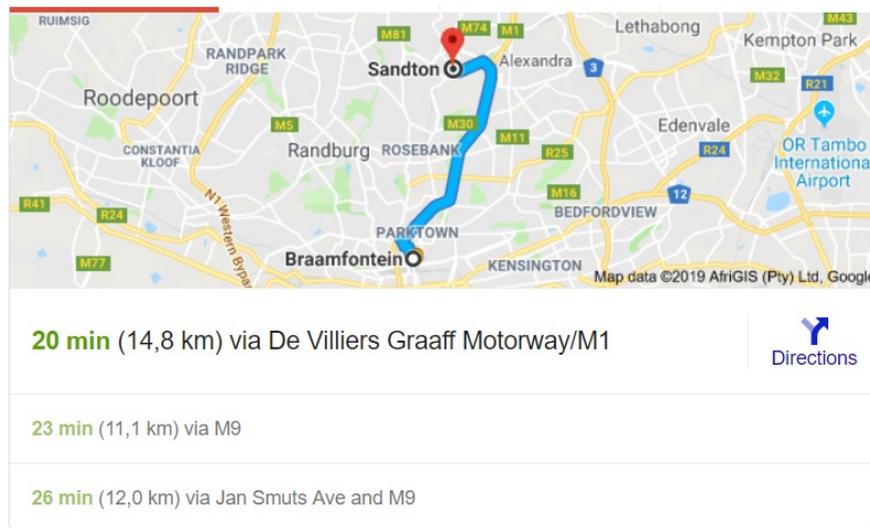


Figure 2. Distance Between Braamfontein to Sandton-Time Calculator with Google map

The increase of population in metropolises such as Johannesburg has consequences on traffic jams which proportionally increase the time involved in TTM. This increase of population resulted in a major problem in terms of taxis delay. TDO involves solving Vehicle Routing Problem (VRP) (see for instance [6, 19, 2]). This paper introduces a new network flow model for TDO using Travelling salesman algorithm embedded in a Decision Tree algorithm (DT). By juxtaposing Travelling salesman algorithm and DT, the network model provides a taxi driver with a minimized travelling time even if he resides in a high-demand area. Several app-based transportation network have been used by taxi companies, such as Lyft, Taxify and Uber to reduce driving cost, improve fuel efficiency and decrease road congestion.

The TDO problem can be mathematically modeled by one of the well-known optimization problems which is the TSP [13, 7, 21]. In this paper, in order to enhance the solution, we propose a new mathematical programming model for the TSP.

1.1. Research Motivation and Literature Review

TSP has mysterious origins which are difficult to identify, however the first example of such a problem was formulated in the German manual *Der Handlungsreisende V on einem alten Commis Traveler* for seller traveling through Germany and Switzerland in 1832 [20]. The TSP can be formulated as follows, a traveler is somewhere (for instance, Braamfontein) he wants to go somewhere (for example Sandton). Before reaching the destination (Sandton), the traveler must visit each city at least once (De Villiers, M9 and Jan Smuts) and he must return to the departure city (Braamfontein). The issue is that the traveler would like to minimize the duration of the trip.

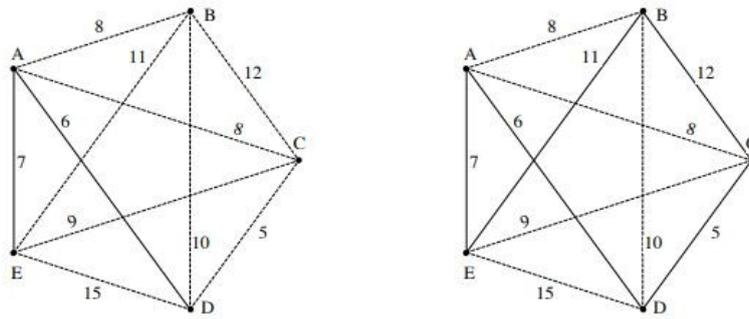


Figure 3. TSP

The traveling salesman problem is NP-complete due to the fact that we have to check that a city was not visited more than once. Then we can compute the total time of each route and check if the time is minimum, this takes polynomial time [17, 4].

2. Proof

Assume $G = (V, E)$ we can create $G^* = (V, E^*)$ where $E^* = (i, j) : i, j \in V$ and $i \neq j$. Thus the time is define as:

$$t(i, j) \Rightarrow 0 \Rightarrow (i, j) \in E \tag{1}$$

$$t(i, j) \Rightarrow 1 \Rightarrow (i, j) \notin E \tag{2}$$

When the set of vertices $x, y, z \in V$ it is obvious that $t(x, y) + t(y, z) + t(z, x) \geq 1$ where t satisfies the triangle inequality. First, it must create a Minimum Spanning Tree (MST) as shown in Figure 4. The MST problem could be formulated as follows, given a connected undirected $G = (V, E, w)$ with edge weight $w \geq 0$, find a sub-graph Tree g in which the total sum of weight w links all vertices v .

2.1. Rational of the research

Given a Metropolitan city such as Johannesburg, find a route which will take less time to arrive at the destination by crossing relevant locations (cities) once before arriving at the destination. We would like to optimize time involved in this scenario.

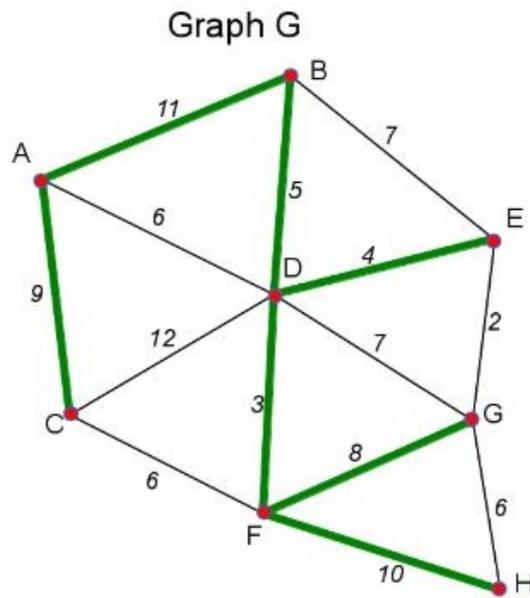


Figure 4. MST

3. Related Works

In a paper by Paruchuri et al. [12], the authors considered a minimalistic four-Branch traffic intersection as shown in Figure 5. Paruchuri et al. [12] used different traffic process as shown in arrows according to National Electrical Manufacturers Association (NEMA) convention. Their research revealed that arrival time at^* can be used in optimizing gains in sparse load scenarios. They reduced stopped delays by over 40,000 hours daily by implementing a computational framework in order to assess the arrival times to enhance traffic signal efficiency.

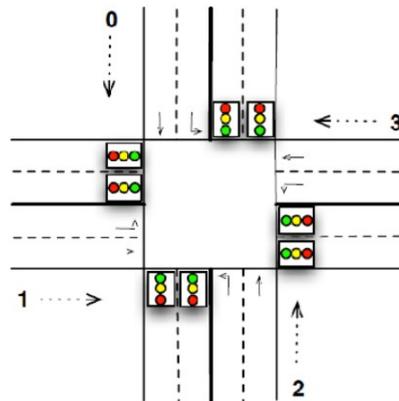


Figure 5. Traffic signal intersection [12]

3.1 Optimization Problem

Paruchuri et al. [12] presented their optimization problem in which $b = [b^{i \leftarrow j}]$ $0 \leq k \leq n^{i \rightarrow j}$ represented the arrival times of vehicles $v^{i \leftarrow j}$ in the direction j towards i . The set of green phases was sorted in ascending order:

$$g = \cup [g^{\rho}]_{\gamma} \tag{3}$$

Where $0 \leq \gamma < m^p$ and M be $|g|$. They then assume that for four green phases:

$$[g_m^{0 \Rightarrow 2}, g_m^{1 \Rightarrow 3}, g_m^{0 \Rightarrow 3, 2 \Rightarrow 1}, g_m^{1 \Rightarrow 0, 3 \Rightarrow 2}] \quad (4)$$

Given g , the time of green phase z can be computed as $g^{d\rho} = (gz) = g^{z+1} - gz - \Sigma_r$ where σ_r is the time it takes to shift from green phase to red phase. The interested reader should refer to [11, 5, 15] for more details. Similarly, Calvet et al. [3] discussed the MDVRP (Multi-Depot Vehicle Routing Problem) using a binary approach:

- 0: Assigning Customers to Depot
- 1: Assigning corresponding Road

Calvet et al. [3], considered heterogeneous depots by developing a simplistic meta-heuristic based method including market clustering in order to optimize expected benefits. The experiment enable the quantification of performance gap.

3.2. Nomenclature

Calvet et al. [3] used r to denote a route of order o which is a finite loop of length $o+2$, here the departure location and the destination is a node o .

$$\rho: \lambda_0 \Rightarrow \lambda_1 \Rightarrow \lambda_2 \Rightarrow \dots \Rightarrow \lambda_{o-1} \Rightarrow \lambda_o \quad (5)$$

Where λ_0 is the departure location and λ_o is the destination. Usually, the cost of a route, C_r and its distance d_ρ can be modeled as:

$$C_\rho := C_{\lambda_o, \lambda_0} + \sum_{k=1}^r C_{k-1, \lambda_k}, d_\rho := d_{\lambda_r, \lambda_0} + \sum_{k=1}^r d_{\lambda_{k-1}, \lambda_k} \quad (6)$$

Hence the optimization for heterogeneous case is:

$$\text{Subject to: } \quad \text{Min } C_r := \sum_{i=1}^m \sum_{\rho \in S_i} (d_{\rho_i} - C_\rho) \quad (7)$$

$$d_{\rho_i} \leq Q_{min}, \rho \in S_i, i = 1 \dots m \quad (8)$$

$$d_\rho \leq Q_{max}, \rho \in S_i, i = 1 \dots m \quad (9)$$

$$\text{3.3. Contribution} \quad S_i \in C_\rho, i = 1 \dots m \quad (10)$$

We present a different approach from Calvet et al. [3] and Paruchuri et al. [12] since we incorporated Decision Tree algorithm in the TSP via the MST. Our approach is not only focusing on routing problems but consider the implication of Machine Learning with regards to TTM.

4. Decision Tree Algorithm

Determining the computational complexity of an algorithm is NP hard (P Vs NP). Computer scientist have tried so hard to study DT because of its simplicity. With DT algorithm, we have a Boolean function f_b which takes $w \in [b]^n \Rightarrow w[0, 1]^n$.

DT algorithms are a kind of automated supervised learning. In this algorithm data is classified according to certain parameters $w [0, 1]^n$. This tree can be explained through binaryzation , namely decision nodes and leafs. Decision nodes are nodes in which data is classified while leafs are terminal results [8]. Recently, Ghasemzadeh et al. [8] implemented DT algorithm to better help drivers in speed selection convenient in adverse weather conditions. The results show that weather conditions, and velocities are factors influencing TTM. Ghasemzadeh et al. [8] implemented DT algorithm for helping drivers to select the appropriate speed when the weather conditions are bad. Each node produces a number representing the probable percentage of choice.

5. Research Methodology

The main objective of this research can be formulated as follows:

- Implementing a platform capable of optimizing traffic jams in the city of Johannesburg
- Incorporating DT algorithm in the platform to solve the MST problem with regards to TDO
- Evaluate the system using *Entropy* and Information gain

The implementation of such a platform raises major difficulties:

- Algorithmic training
- Extracting GPS coordinates in real-time form Google map
- Visualizing extracted coordinates via a graphical interface

5.1. System Architecture

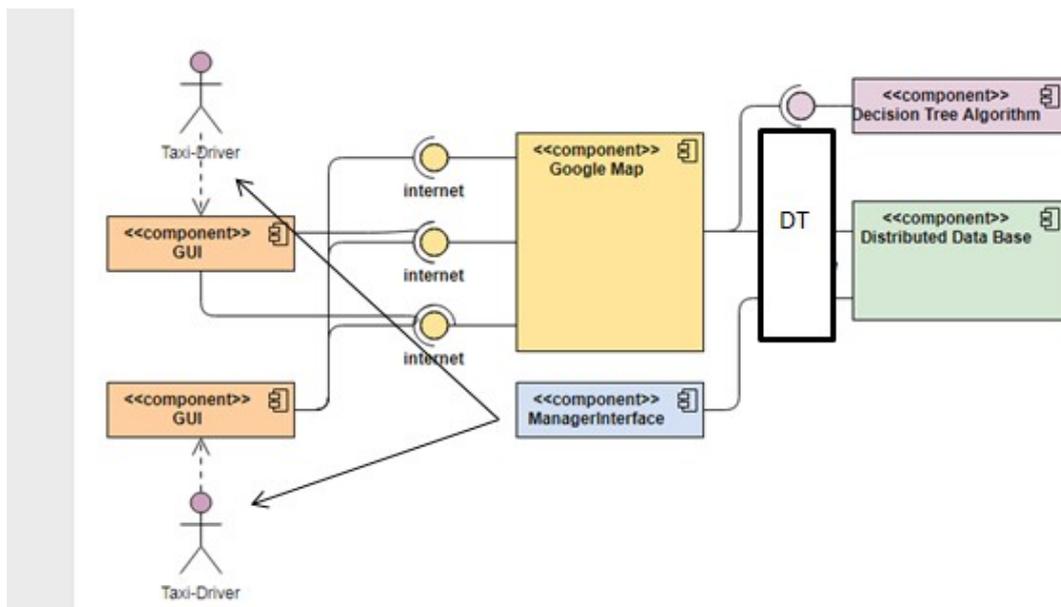


Figure 6. Schematic Description of the System

In a typical case, a driver interacts with the system which relies on GPS coordinates. Then, the driver must specify his/her destination. The system aims to extract, analyze, aggregate and classify GPS coordinates based on DT algorithm, which is very important for classification (supervised Learning). It is therefore imperative to emphasize that the system is totally independent of Google Map so that it can predict the optimal route based on the training data. The architecture of the system is explained in Figure 7, and a small description of each part of the system is provided.

- GUI: graphical User Interface used by the Taxi Driver to find the optimal route. The taxi driver must specify the destination using this GUI
- Internet: crucial component of the computational framework which support Google Map and GPS coordinates
- Google Map: a real-time web platform which produced geographic coordinates covering locations around the world. Google Map produces road maps, offering satellite views of many places around the world. Google Maps can also offer street views including photos taken from vehicles
- Manager Interface: this component is the platform’s controller managing input and output
- DT (Decision Tree): selected algorithm which predict the optimal route. The prediction should be different from Google Map
- Distributed Database: collection of data/information. The ground Truth used during training stages was recorded in the database for templates matching

5.2. Quality Appraisal of the data set

In this section, we present a sample of the data set used (see Figure 7). The coordinates was manually created by the authors and utilized as training data set. We used a survey in which 100 taxi drivers provided relevant information regarding optimal routes based on routine experiences. These coordinates were recorded in a text file for training DT algorithm. To test the system, comparisons were made between Google Map and the proposed system predictions. The 100 taxi drivers were tasked to used Google Map and the propose system in real time. Lastly, drivers were requested to comment on which system is suitable in providing optimal routes in the city of Johannesburg.

Braamfontein To Sandton		
Departure	Cross-Place	Time in sec
Braamfontein	De Villiers	15
Braamfontein	M9	20
Braamfontein	Jan Smuts Ave	22
De Villiers	De Villiers Graaff Motor Way/M1	30
De Villiers	M9	28
De Villiers	Katherine St/M85	31
...		
...		

Figure 7. Optimal routes from routine experience

Table 1. Optimal Routes from Google Map.

Departure	Cross-Place	Time in sec
Braamfontein	De Villiers	22
Braamfontein	M9	22
Braamfontein De	Jan Smuts Ave	28
Villiers De	Rivonia Rd	16
Villiers	M9	19
...		

Table 2. Difference Between Optimal Routes predicted by Google Map and routine Experience.

Departure	Cross-Place	Time in sec
Braamfontein	De Villiers	7
Braamfontein	M9	2
Braamfontein	Jan Smuts Ave	6
...		

5.3. Why Decision Tree

We can express complex decisions using a series of questions; for instance
What to do this Weekend?

- If my parents are visiting: We will go to the cinema
- If not, I will play tennis if it is sunny
- I will go shopping if it is windy and I am rich,
- I will go to the cinema if it is windy and I am poor
- I will stay in if it is rainy

We can use DT algorithm to express these decisions (see Figure 8):

We can also use *Horn – Clauses* in first order logic to implement DT algorithm, for instance in the above example: $NOT (parents) \wedge sunny\ day \Rightarrow Play\ Tennis$.

According to Hong et al. [9] Decision Tree can be seen as rules for performing classification and this algorithm is learning from examples not turning thought processes into decision trees. This imply that We need to put examples into clusters. We also need objects and attributes for examples. Here *attributes* describe examples (background knowledge). Each objects and attribute take only a finite set of values [9, 8]. The main issue in DT learning is the following:

- In which positions can we put a node
- How can we include the leaf and root node in the tree?

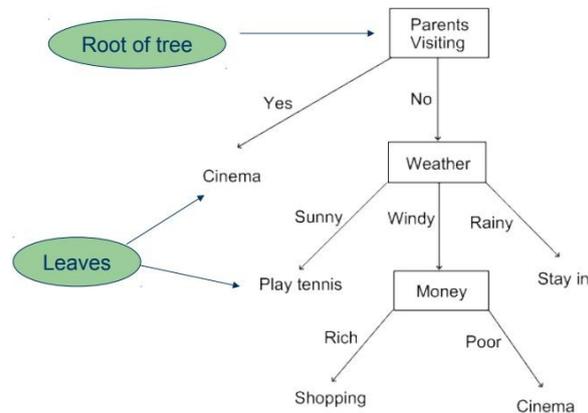


Figure 8. DT algorithm

We must use a measure called Information Gain which rely on a notion of *entropy* (Data Impurity). Node with the highest information gain is chosen and when there are no choices, a leaf node is put on. For instance, in a binary classification where p^+ is the proportion of positivity in a sample S and p^- is the proportion of negativity

$$Entropy(S) = -p^+ \log_2(p^+) - p^- \log_2(p^-) \tag{11}$$

5.4. Information Gain

Given a set of attributes A and examples S where $A = V_1 \dots V_m$. Here V represents the value. Let S_v be examples which take value v for attribute A . To compute $Gain(S, A)$ we must approximate the reduction in *entropy* to achieve this we must know the value of attribute A for the examples in S :

$$Gain(S, A) := Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{12}$$

6. Implementation

Before delving deeper into the implementation, the authors would like to present the Decision Tree for the data set given in Figure 7, Table 1 and 2. Lastly, the Graphical User Interface is presented in this section.

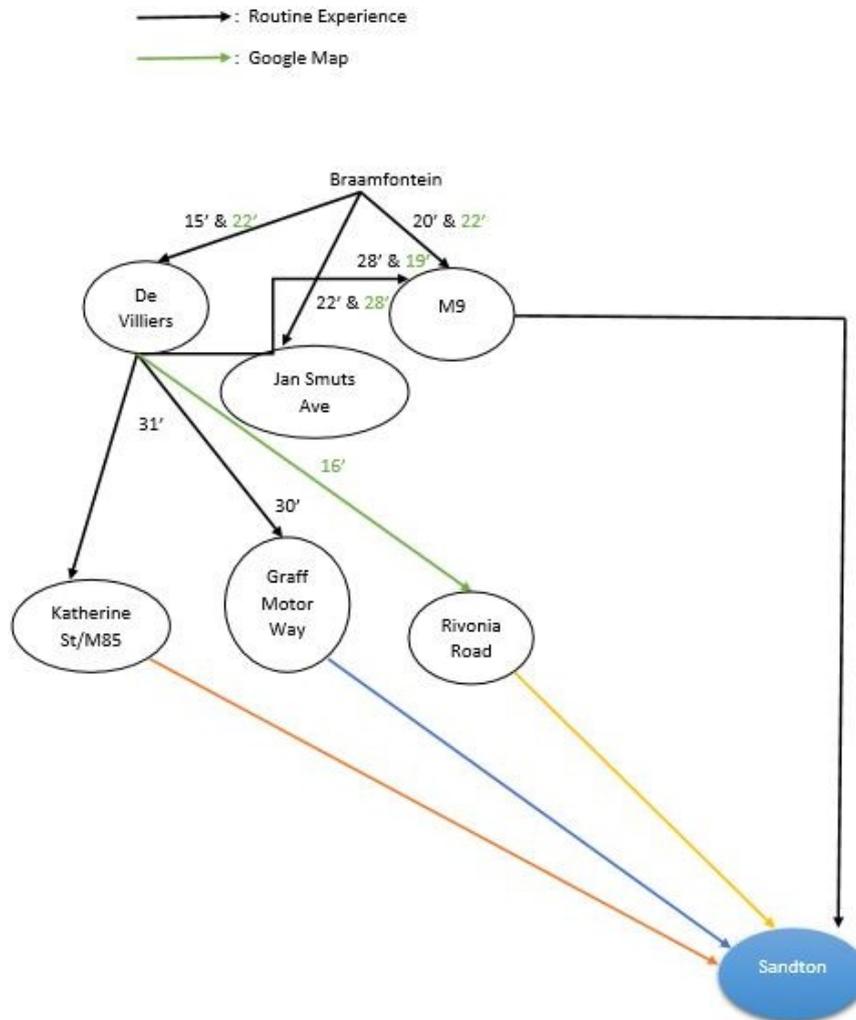


Figure 9. Decision tree for the data set

6.1. Horn clause

It all started in 1951, when an American logician named Alfred Horn emphasized Horn's clause importance in an article entitled "On sentences which are true of direct unions of algebra" [10, 18]. This marked the beginning of logic programming which is a form of horn clause. In Programming Logic or, mathematical logic, any formula having a particular rule mimicking logical properties or formal specifications is a Horn clause used in model theory [10]. In this section, we presented the Horn clause derived from Figure 9.

$$\text{Braamfontein} \quad \wedge \text{De Villiers} \wedge \text{Katherine St/M 85} \Rightarrow \text{Sandton} \quad (13)$$

$$\text{Braamfontein} \quad \vee \text{Jan Smuts Ave} \quad \vee \text{De Villiers} \wedge \text{M9} \Rightarrow \text{Sandton} \quad (14)$$

$$\text{Braamfontein} \quad \wedge \text{De Villiers} \wedge \text{Graaf Motor Way} \Rightarrow \text{Sandton} \quad (15)$$

$$\text{Braamfontein} \quad \wedge \text{De Villiers} \wedge \text{Rivonia Road} \leftarrow \text{Sandton} \quad (16)$$

$$\text{Braamfontein} \quad \wedge \quad M9 \quad \text{Sandton} \quad (17)$$

$$\text{Braamfontein} \quad \wedge \quad \text{NOT}(\text{De Villiers}) \quad \wedge \quad \text{NOT}(\text{Jan Smuts Ave}) \quad \wedge \quad \text{NOT}(M9) \Rightarrow \text{NOT}(\text{Sandton}) \quad (18)$$

6.2. Queuing

In computer science, any collection of order-data represents a queue. We can record these data according to the FIFO (First In First Out) or LIFO (Last In Last Out) approach. Qiu et al. [14] emphasized that the first item to be registered in the FIFO queue will be the first to be executed by the CPU (Control Process Unit). This means that in this approach, once a new element is inserted, all previously inserted elements must be executed before the new element can be inserted as well. The LIFO approach is only a reverse FIFO process. We can define a queue as a sequential collection of items or as a linear data structure [14]. In real life, the majority of queues are an integral part of networks. For example, passengers arriving at a seaport will have to make their way through check-in, security checks and boarding to different destinations. It is with this in mind that the study of queue networks is of great importance in TTM applications. In the following mathematical modeling, Q represents the queue and t the time while T is the total time ($T = [t_1 + t_2 + t_3 + \dots + t_n]$). In this section, we comprehensively substantiate our Queuing approach using the data set presented in Figure 7, Table 1 and 2.²

$$Q_1[\text{Braamfontein} \wedge_{t=15^J} \wedge \text{De Villiers} \wedge_{t=31^J} \text{Katherine St/M85}] T = 15^J + 31^J = 46^J \quad (19)$$

$$Q_2[\text{Braamfontein} \wedge_{t=22^J} \wedge \text{Jan Smuts Avenue} \wedge_{t=28^J} \text{DE Villiers M9}] T = 22^J + 28^J = 50^J \quad (20)$$

$$Q_3[\text{Braamfontein} \wedge_{t=15^J} \wedge \text{De Villiers} \wedge_{t=30^J} \text{Graaf Motor av}] T = 15^J + 30^J = 45^J \quad (21)$$

$$Q_4[\text{Braamfontein} \wedge_{t=15^J} \wedge \text{De Villiers} \wedge_{t=16^J} \text{Rivonia Road}] T = 15^J + 16^J = 31^J \quad (22)$$

$$Q_5[\text{Braamfontein} \wedge_{t=20^J} \wedge M9] T = 20^J \quad (23)$$

²We only consider routine experience because we used this data set to train Decision Tree algorithm

Q_5 is the optimal route because it will take only 20' to get to the destination. If there is traffic jams then Q_4 will be the optimal route and so forth. The Decision Tree algorithm will create a Queue each and every time a driver cross a location; this queue will be updated in real time according to the ground truth used in the training stage. We used a *FIFO Timing* approach whereby the First queue to be display on the Graphical Interface will be the one with minimum time. Similarly, Queues with maximal time will not be display (for instance Q_1, Q_2, Q_3)³. This is shown in section 6.3.

6.3. Graphical User Interface



Figure 10. Interaction 1: The Graphical User Interface

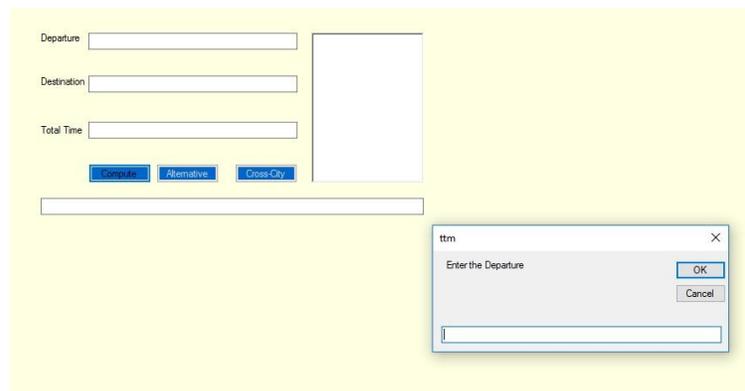


Figure 11. Interaction 2: when the driver clicks on the Compute Button

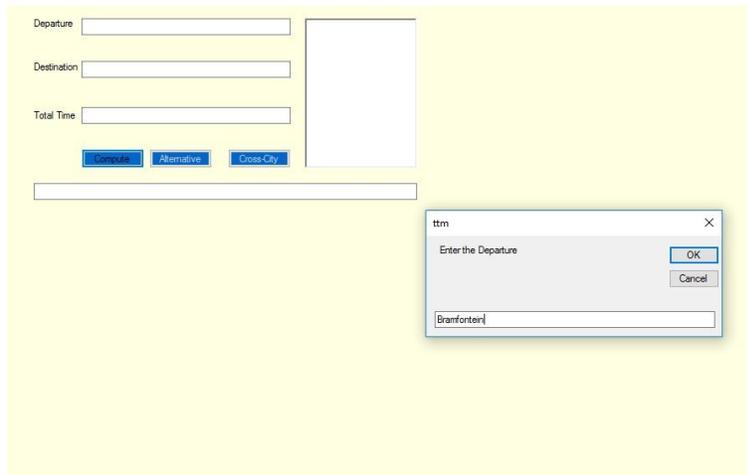


Figure 12. Interaction 3: inserting the departure location

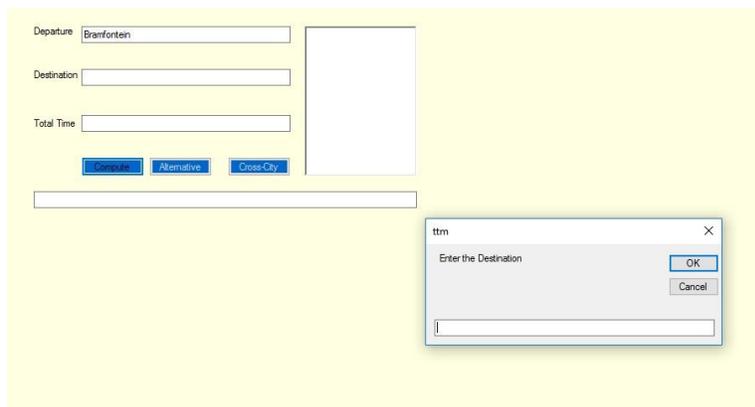


Figure 13. Interaction 4: the departure location is stored in the Queue

6.4. Conclusion & additional results

In this research we presented a simple prototype optimizing traffic jams in the city of Johannesburg. The training data has been labelled based on routine experience of taxi drivers in the afore mentioned city. The research also shows us that Google Map is not always accurate with regard to locate the optimal route. The great limitation of our prototype remains the training data especially if we would like to implement the same technique in a complex city.

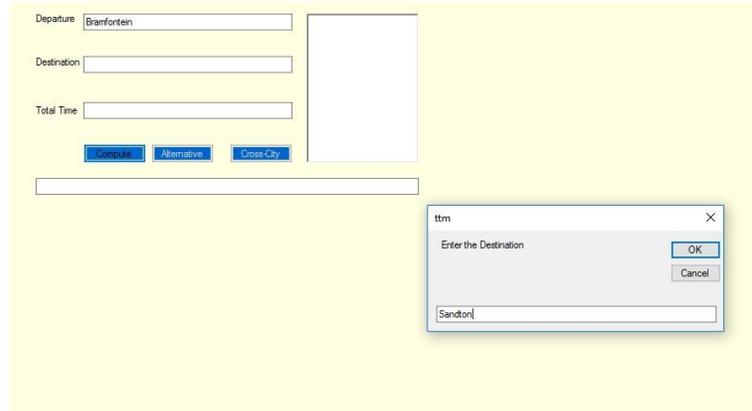


Figure 14. Interaction 5: inserting the destination



Figure 15. Interaction 6: the destination is stored in the Queue and the total time is computed by the decision tree algorithm



Figure 16. Interaction 7: when the driver click on the “Alternative Button”. This button displays an alternative of the optimal route



Figure 17. Interaction 8: when the driver click on "Cross-city" button. This button displays all the relevant locations that he must cross before reaching the destination



Figure 18: Interaction 9: When the driver click on the "Clean" button. All the text boxes are left blank. The exit button close the application

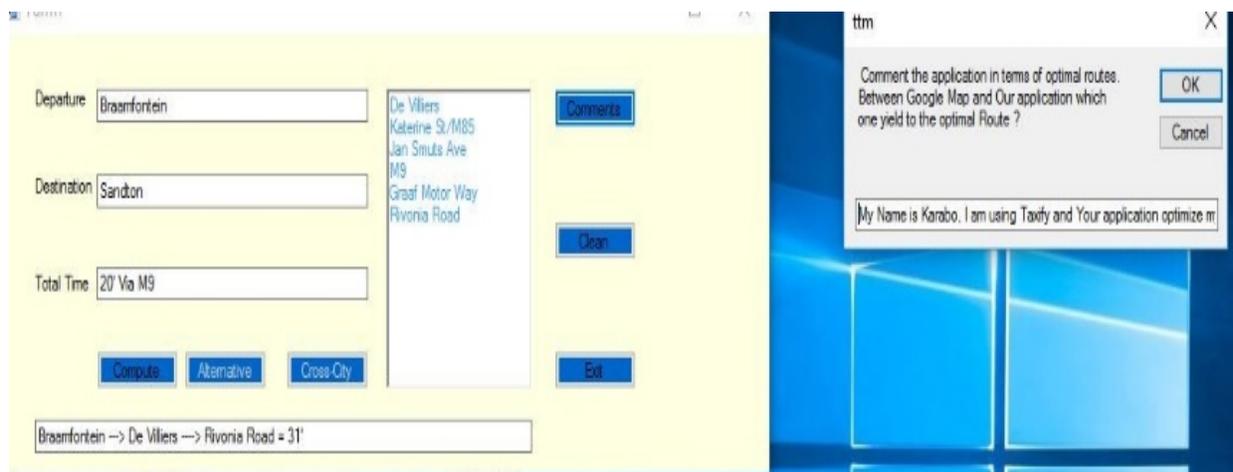


Figure 19: Interaction 10: When the driver click on the "Comments" button. An input box pop up to stored the comment in a text File

References

- [1] Ang, B. Urban transportation management and energy savings: the case of Singapore. *International Journal of Vehicle Design*, 17(1):1–12, 1996.
- [2] Braekers, K., Ramaekers, K., and Van Nieuwenhuysse, I. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.
- [3] Calvet, L., Ferrer, A., Gomes, M. I., Juan, A. A., and Masip, D. Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation. *Computers & Industrial Engineering*, 94:93–104, 2016.
- [4] Chan, T.-H. H. and Jiang, S. H.-C. Reducing curse of dimensionality: Improved ptas for tsp (with neighborhoods) in doubling metrics. *ACM Transactions on Algorithms (TALG)*, 14(1):9, 2018.
- [5] Clarke, G. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581.
- [6] Drexl, M. Synchronization in vehicle routing a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3):297–316.
- [7] Foumani, M., Moeini, A., Haythorpe, M., and Smith-Miles, K. A cross-entropy method for optimising robotic automated storage and retrieval systems. *International Journal of Production Research*, 56(19):6450–6472, 2018.
- [8] Ghasemzadeh, A., Hammit, B. E., Ahmed, M. M., and Young, R. K. Parametric ordinal logistic regression and non-parametric decision tree approaches for assessing the impact of weather conditions on driver speed selection using naturalistic driving data. *Transportation research record*, page 0361198118758035, 2018.
- [9] Hong, H., Liu, J., Bui, D. T., Pradhan, B., Acharya, T. D., Pham, B. T., Zhu, A.-X., Chen, W., and Ahmad, B. B. Landslide susceptibility mapping using j48 decision tree with adaboost, bagging and rotation forest ensembles in the guangchang area (china). *Catena*, 163:399–413, 2018.
- [10] Lam, H.-P. and Hashmi, M. Enabling reasoning with legal rule ml. *Theory and Practice of Logic Programming*, 19(1):1–26, 2019.
- [11] Laporte, G. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.
- [12] Paruchuri, V., Chellappan, S., and Lenin, R. B. Arrival time based traffic signal optimization for intelligent transportation systems. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 703–709, 2013. IEEE.
- [13] Perboli, G., Rosano, M., Saint-Guillain, M., and Rizzo, P. Simulation–optimisation framework for city logistics: an application on multimodal last-mile delivery. *IET Intelligent Transport Systems*, 12(4):262–269, 2018.
- [14] Qiu, T., Xia, F., Feng, L., Wu, G., and Jin, B. Queuing theory-based path delay analysis of wireless sensor networks. *Advances in Electrical and Computer Engineering*, 11(2):3–8, 2011.
- [15] Rossi, F., Zhang, R., Hindy, Y., and Pavone, M. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. *Autonomous Robots*, 42(7):1427–1442, 2018.
- [16] Thomson, J. M. (1978). *Great cities and their traffic*. 1978
- [17] Traub, V. and Vygen, J. Approaching [equation] for the st-path tsp. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1854–1864, 2018. Society for Industrial and Applied Mathematics.
- [18] Wang, W. and Jiao, L. Trace abstraction refinement for solving horn clauses. *The Computer Journal*, 59(8):1236–1251, 2016.
- [19] Wassan, N. A., Salhi, S., Nagy, G., Wassan, N., and Wade, A. C. Solving the mixed backhauling vehicle routing: Problem with ants. *International Journal of Energy Optimization and Engineering (IJEEO)*, 2(2):62–77.
- [20] William, J. Cook. in pursuit of the traveling salesman. mathematics at the limits of computation. 2012
- [21] Zhang, L., Lv, H., Tan, D., Xu, F., Chen, J., Bao, G., and Cai, S. Adaptive quantum genetic algorithm for task sequence planning of complex assembly systems. *Electronics Letters*, 54(14):870–872, 2018.

Biographies

Mike Nkongolo completed his master's degree in computer science from the Department of Computer Science and Applied Mathematics at the University of the Witwatersrand, Johannesburg, South Africa-2019. He earned a B.S. in Information Technology from Université Liberté (Freedom University), Drc, Lubumbashi-2013, Postgraduate Diploma in Computer Science from the University of the Witwatersrand, Johannesburg, South Africa-2015, Bachelor of Science (Honors) in Computer Science from the University of the Witwatersrand, Johannesburg, South Africa-2016. He has published journal and conference papers in both languages, French and English. Mr. Mike Nkongolo Wa Nkongolo has completed his Master research project under the supervision of professor Turgay Celik. His research interests include Artificial Intelligence, Modal Logics, Machine Learning, Data Science, Natural Language Processing, Combinatorics and Discrete Mathematics, and Databases. He is a member of Grin Verlag and currently working as an Information Technology lecturer for a private institution in Pretoria, South Africa.

Laby ILumbe is a professional academic with a wide-range experience of lecturing subjects related to Transport & Supply Chain Management. He completed a Bachelor Technology in Logistics with Cum Laude and holds a B. Com Honours in Logistics Management from University of Johannesburg (UJ). He is currently in the process of completing his master's degree (M. Com) in Supply Chain Management. He holds various positions within industries such as a consultant in logistics and an assistant lecturer in education for the school of Business and Economics at University of Johannesburg (UJ). Laby interest in how technology can be leveraged to augment logistics channels and network design as well as alignment of supply and demand. Further research interests include privatization of railways, humanitarian disaster relief, blockchain, supply chain difficulties in gas sector, air freight in SADC region and the logistics of international trade.