INTELLIGENT TRAFFIC LIGHT MANAGEMENT SYSTEM (ITLMS)

An smart, dynamic management system for a smoother ride.

MR-RS Group 14
Ang Boon Yew A0096966E
Kartik Chopra A0198483L
Karamjot Singh A0198470U

Contents

Executive Summary

Transportation is a core area of a country's economy and functions, providing the means of quick and efficient movement of their population and goods through the country. Traffic lights are a critical component in road networks, directing the flow of vehicular and pedestrian traffic and ensure that transportation runs smoothly. Poor design and management of such traffic control systems may lead to traffic jams as well as road accidents, hampering economic activity and more importantly, the loss of citizens' lives. In addition, wait times are known to induce stress and affect psychological well-being for commuters, which may have adverse effects on health and work productivity.

Hence, there is a need for efficient and robust traffic control systems globally, including Singapore. Singapore as a geographically dense city-state, relies heavily on such systems to drive road transportation. In this project, we propose an intelligent traffic light management system which provides recommended green-light timings for different phases and directions in a single traffic junction. The system consists of an inference engine for assessing traffic conditions and processing traffic volume, a genetic optimization model to provide recommended green-light timings and a user interface that can be extended to include future road traffic sensor data.

1. Background & Introduction

Singapore has a huge on-road vehicle count. It ranks among one of the world's most vehicle populous cities of the world. According to the latest LTA statistics, there were 957,006 registered vehicles by the end of last year and this number is projected to increase by & capped at least 0.25% next year [1]. This large number of vehicles on the island is supposed to come with a downside - wait-times at the traffic signals.

Currently, the traffic lights in the city operate on historical data where at any given signal today, most of the predictions on the wait-times are based on past observations of vehicles that were present at the traffic signal at a distant past day. This approach of calculating the wait-times has a serious shortcoming - since this "time to green" is not based on real-time number of vehicles at a junction, vehicles on the opposite side of the junction need to wait for a static number of seconds, even when this side of the signal has no car left.

This leads to 2 issues:

1. Every 'green' second wasted on one side of the signal gets added as a 'red' second to every vehicle on other sides.
2. If the number of cars on the opposite side is huge and increasing, there is a strong chance, that many cars at the end of the signal will have to wait at the signal for the second time, as provided green time might not be sufficient - as it is provided statically.

These wait-times increase at peak times of the day throughout the city, the toll of which every passenger at any given signal has to bear.

As it is a well-known fact that commute wait-times can induce stress and affect psychological well-being [2], it is apt to design intelligent systems that can help mitigate this issue in any way possible.

Hence, there is a need for efficient and robust traffic control systems globally, including Singapore. In a traditional traffic light control system, the timings given to each traffic direction is often provided by a pre-set timing at different times of the day and does not consider the current traffic flow demand. In Singapore, the Green Link Determining (GLIDE system) accounts for this demand by determining traffic flow using road sensors to detect vehicular movement [3]. However, the system does not account for other factors which may influence the future traffic demand, such as arrival rates and queue lengths behind a junction.

2. Existing Literature

Methods to optimize and improve existing traffic light systems have been studied extensively, covering traditional methodologies such as rule-based systems, more advanced techniques, complex models such as using reinforcement learning and neural networks [4]. Queuing theory, which is widely used to model service processes has also been used in traffic light optimization [5]. These methods often aim to improve the traffic volume flow, measured by the number of vehicles passing through a selected junction, or to reduce the incidences of queues and traffic jams occurring within a road network by minimizing delays at the intersection. To support the use of these methods, sensors to detect road traffic have also been examined and deployed [6].

In rule-based systems, traffic light timings are controlled pre-determined rules based on traffic information gathered using road sensors or other means. Fuzzy logic controllers are more advanced examples of such systems and determines an output from input parameters using membership functions. Kulkarni et al. introduced a fuzzy logic controller to determine the state of current traffic demand and changes the traffic light status if necessary [7]. Alam et al. also proposed a fuzzy traffic light control system which determines the traffic demand using membership functions for queue and arrival rates [8]. Azimirad et al. introduced different fuzzy sets for vehicle queue length and expected waiting time for controlling traffic light states in order to reduce overall traffic waiting time [9].

Advanced techniques such as genetic algorithms and simulation optimization techniques to determine traffic light timings are also widely examined in current literature. Simulation optimization techniques apply an optimization model on simulated outputs that account for stochastic processes in order to determine an solution [10]. Genetic algorithms are a family of optimziation techniques which have been inspired by the natural process of evolution and natural selection and have been widely applied in many settings, such as design optimziation, resource scheduling and solving combinatorial problems [11]. Wiering et al. use a reinforcement learning approach with a traffic simulation model to minimize the expected waiting time for vehicles at a traffic junction [12]. Kalganova and Russell proposed a model to minimize traffic cycle delay across a network of junctions using a genetic algorithm that determines the length and order of green-light timings [ref]. Teo et al. considers the queue length and traffic accumulation in red-light junction directions when allocating green light timings [13]. Sanchez et al introduced an architecture using genetic algorithms, cellular automata simulators to minimize the time spent in the junction for a vehicle [14].

3. Methodology

In this project, we develop a prototype system for recommending traffic light timings in a typical signal cycle for a single traffic junction. The system comprises of 4 main components:

1. User Interface / Traffic Volume Sensors
2. Flask Server
3. Rules Engine
4. Genetic Algorithm Model

The user interface (for demonstration) or traffic volume sensors gather required input parameters, such as the number of lanes and the number of vehicles on the road, which are then transmitted to the Flask server through a HTTP POST request.

The Flask server processes the input data into useable formats and passes the parameters to the Pyke rules engine, which determines outputs that can be used by a rules-based timing calculator or a genetic algorithm.

The timing calculator and genetic algorithm determines the estimated timings required for each direction and returns the timings to the Flask server. These timings can be then be displayed back to the user interface, or to be used in a real-time traffic light control system.

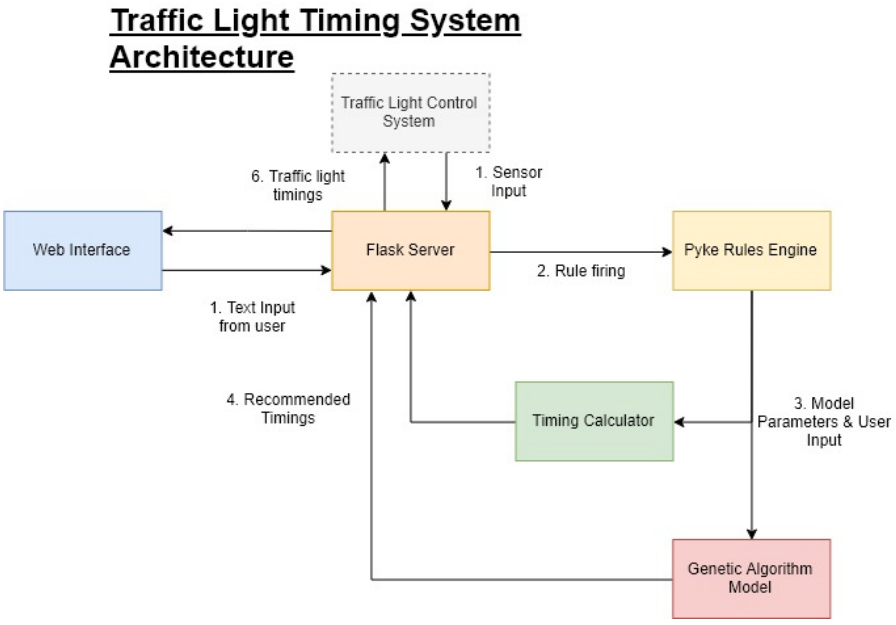

Figure 1: System Architecture

3.1 Demonstration User Interface / Traffic Volume Sensor Systems

Sensor Integration

In a real-time implementation, information on the road traffic conditions can be captured and transmitted using multiple sensors placed around the road junction. Data from these sensors are fed into a processing server such as Flask which can then be used by the Pyke Rules Engine and Genetic Algorithm model to estimate the green light timings for each traffic direction.

In order to implement the system, sensors which use two widely available road sensing techniques can be considered:

- Video Cameras: Signals from on-junction cameras can be used to estimate the number of cars through an image classification model
- Pressure Sensors: The number of cars on road will be calculated by a sensor processing model/unit which detects the movement of cars along the road.

Demonstration System

A user can access the demonstration system through a web interface that is hosted using a Python Flask server. Flask is a server microframework based in the Python that processes RESTFUL APIs and handles the routing of these requests.

The user can select the type of junction to analyze and enter in the number of lanes, the number of vehicles and each lane, the estimated arrival rate of vehicles and the number of right lanes in that direction and assess the quality of the timing results.

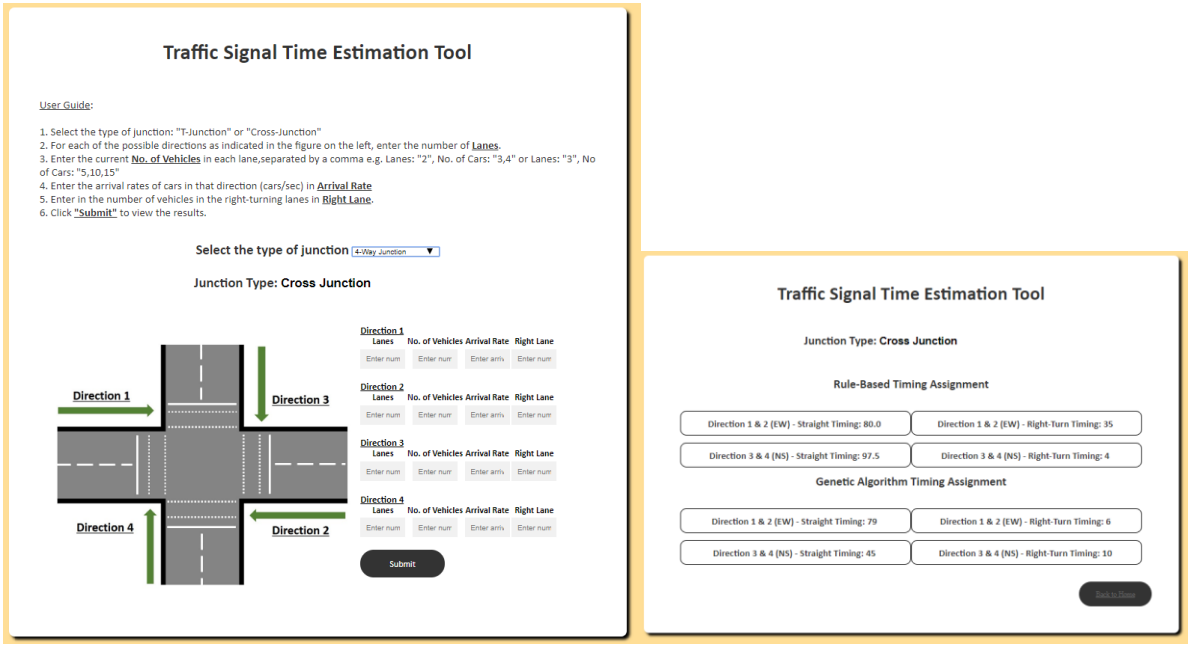

Figure 2: User Interface for Demonstration

3.2 Rules Engine

Python Knowledge Engine (PyKE)

PyKE is the Python Knowledge Engine that allows users to perform decision making based on facts, rules and questions [15]. PyKE has 3 main parts that make up its knowledge base –

1. Knowledge Fact Base – It allows the user to save their knowledge that can be represented as facts.
2. Knowledge Rule Base – It allows the user to have rule bases for their knowledge.
3. Knowledge Question Base – It allows users to store knowledge in the form of questions.

Our application leverages the Knowledge Rule Base (krb) and the Knowledge Question Base (kqb).

Knowledge Rule Base – The PyKE rule base uses forward chaining to fire rules based on facts present in the fact base (kfb). Our implementation is capable of linking junction types (big or small) based on past junction facts. For example, a four-way junction is classified as a 'Big' junction based on the past fact (present in fact base) that a three-way junction is (also) 'Big'. This implementation can be further enhanced to classify junction types into even more classes – based on not only the number of directions in a junction, but also the number of lanes in each direction.

Also, the bias used in each time calculation formula is being derived from a rule, where each junction size is allotted a set bias – which will be configured based on domain expert's knowledge, e.g.

- If *Junction Size* = <*Big*> Then
  Bias ← 25 Seconds
- If *Junction Size* = <*Small*> Then
  Bias ← 15 Seconds

Knowledge Question Base – Our PyKE engine asks the users the following questions –

1. The number of sides to the junction.
2. The number of Straights lanes on each side.
3. The number of Right lanes on each side.
4. The number of cars present in each lane.

Based on the answer to these questions and the rules from the krb system, our application feeds the extracted knowledge into our formula and calculates the time to be given on each side of the junction.

3.3 Rule-Based Timing Assignment

Singapore government is also prototyping a similar solution called GLIDE. Our approach is different from the glide system as the glide system only takes into account the current flow of traffic on one side of the Junction. Also, the glide system is only functional on the major junction across Singapore where is our system can be deployed on each individual traffic signal.

We calculate the time to be given on each side of the junction by taking into account the number of cars present at the junction in real time. we not only take into consideration the number of cars but we also take into consideration the number of lanes that are present on either side of the junction. Another factor that influences our time is how quickly a car crosses the junction. We also factor in a penalty which is basically the number of cars on the other side of the junction so that the time is not increasing on just one side but it gets regulated by the number of cars.

Our derived formula to calculate the straight time is as follows:

$$t_s = (max\_cars * m + bias) - penalty$$

Where,

- t_s is the time given to the opposite straight lanes.
- max_cars are the maximum cars in either of the straight lanes of a particular side of the junction.
- m is the time taken in seconds by a single car to cross the junction.
- bias is the base time given to a side.
- penalty is the time deducted based on cars on other sides.

We found m to vary from 3 seconds to 4 seconds and hence we fixed m to be 3.5 that is the average. The bias or the base time is calculated based on the number of lanes a side has. If a side has greater than 3 lanes it is considered a large junction and the bias is set to be 25 seconds, whereas a side with less than 3 lanes is considered small and the bias is set to be 15 seconds. The bias is added so that there is always some walking time for pedestrians even if no cars are present. The penalty is calculated based on the number of cars on the other sides of the junction. It is a sum of the total cars on the remaining sides divided by the number of lanes.

$$penalty = \frac{1}{lanes} \sum total\ cars$$

For the time given to the lanes that turn right, it is based on the following rules –

- If the present side has more cars than the opposite side, then the time given to the right side is the difference in number of cars along with a small bias.
- Otherwise if the other side has more cars and has no right turn lanes, then the right turn time is a bias value.

3.4 Genetic Algorithm Optimization

Using the parameters that are derived from the rules engine, we develop a genetic optimization model for allocating green-light timings to the respective phases in traffic signal cycle for a single traffic junction. In addition, the model is also able to consider the new incoming vehicle arrivals that will cause traffic build-up during the red light phase for each direction, balancing the time allocated to each phase. A traffic signal cycle refers to one completion of several traffic movement phases, where each phase has a green light activated for one direction of traffic flow to allow vehicles from that direction to move across the junction.

Phase 1: Green light is activated for direction i = 1,2 and vehicles in that direction move first. Red light is activated for direction i = 3,4

Phase 2: Red light is activated for vehicles travelling straight in direction i = 1,2. Green arrow light is activated for right turning vehicles from these directions.

Phase 3: Red arrow light is activated for right-turn vehicles in direction i = 1,2 to signal end of Phase 2 and green light is activated for direction i = 3,4.

Phase 4: Red arrow light is activated for vehicles travelling straight in direction i = 3,4 to signal end of Phase 3. Green arrow light is activated for right turning vehicles from these directions.

Model Formulation

I : Set of directions i
$\in \{1,2,3,4\}$ representing North, South, East and West directions respectively

K : Set of roads k
$\in \{1,2\}$ representing the road that allows traffic in both directions where k
$= 1$ represents traffic in opposite direction pairs $i \in \{1,2\}$ or $\{3,4\}$ and k
$= 2$ represents the opposite pair

a_{ik}^{r} : The number of cars waiting at direction i on road k, where $r =$
1 if it is a right turn lane and 0 if otherwise

λ_{ik}^{r} : The mean arrival rate of cars coming from direction i on road k, where r
$= 1$ if it is a right turn lane and 0 if otherwise

t_j :
Decision variable as the length of the green light timing given to traffic in cycle phase $j \in \{1,2,3,4\}$

m
: Mean length of time required for a vehicle to cross the junction. We set this to be 3.5 seconds

The objective of the model is balance the traffic flow across all directions by minimizing the mean absolute deviations between the traffic flows for each phase:

$$minimize\ \frac{1}{n}\sum_{i=1,j=i+1}|v_i - v_j|$$

$$\text{Maximum No. of Cars on Road } k\text{: } d_k^r = \max_i a_{ik}^r$$

$$\text{Expected Amount of Cars Arriving : } \lambda_k^r = \sum_{i\in k}\lambda_{ik}^r$$

For each phase, the estimated traffic flow is calculated:

$$\text{Phase 1 Traffic Flow : } v_1 = \frac{t_1}{m(d_1^0 + \lambda_1^0 t_1)}$$

$$\text{Phase 2 Traffic Flow : } v_2 = \frac{t_2}{m(d_1^1 + \lambda_1^1 t_1)}$$

$$\text{Phase 3 Traffic Flow : } v_3 = \frac{t_3}{m(d_2^0 + \lambda_2^0 (t_1 + t_2))}$$

$$\text{Phase 4 Traffic Flow : } v_4 = \frac{t_4}{m(d_2^1 + \lambda_2^1 (t_1 + t_2 + t_3))}$$

The genetic algorithm was implemented in Python using the *pyeasyga* package, which is a simple framework for deploying genetic algorithm optimization models in Python [16].

We define each individual chromosome as $[t_1, t_2, t_3, t_4]$, representing the green-light timings given for each phase. At that beginning of the algorithm we generate an initial population of size 100 of chromosomes which each contain a value between [5,80], representing the upper and lower limits for the timings to be given. A generation size of 200, chromosome crossover rate of 0.8 and mutation rate of 0.05 to was used in order to reduce the computation time and ensure fast results for implementation.

4 Results & Discussion

In order to study the existing traffic system and to evaluate our proposed system's performance, a three-way junction on Clementi Street 13 was chosen as an example site. Attributes of the junction, such as the number of straight and right-turning lanes, the number of vehicles and the time taken for the vehicles to clear the junction were recorded.

Figure 3: Example Junction at Clementi Street 13

Several recordings of the actual movement of traffic at the busy three-way junction of Clementi Street 13 were taken to benchmark our system's performance. The quantification of one such recording taken on 16th September at 1630 Hrs. is shown in the table below.

The timings allocated by the current traffic light control system was recorded along with the actual time the vehicles took to clear the junction as a benchmark. Using the formula-based method and the genetic algorithm model, we generated the recommended timings for each model.

		Direction 1 - Straight	Direction 1 – Right Turn	Direction 2	Direction 3
N. Straight Lanes		4	-	3	2
N. Right Lanes		-	1	-	-
Total Vehicles		26	2	47	7
N. Vehicles per lane (max)		**7** 7 7 5	2	15 **17** 15	**4** 3
Actual Junction Crossing Time		26	7	47	12
Current System	Allotted Time	57	23	57	48
	Space-Time	31 (10*)	16	10	36
ITLMS: Rules-Based	Allotted Time	79	7	79	24
	Space-Time	43 (32*)	0	32	12
ITLMS: Genetic Algorithm	Allotted Time	78	10	78	32
	Space-Time	42 (31*)	3	31	20

*As there are opposite directions of the same road, the max actual time on that road is taken as space-time as the green light signal would remain activated for the direction taking more time.

Table 1: Summary of Junction Attributes and Results

Using the calculated timings for the rules-based and genetic algorithm, we determined the resulting space-time for each direction or phase of the traffic light cycle. Space-time refers to the amount of time where there are no vehicles travelling when a green light is given for a particular direction. Longer space-time hence results in less system efficiency, as the green light time could have been given to another direction for vehicles to pass.

Current System

$$Wait/Cycle\ Time = \sum Time\ Alotted = 57 + 23 + 48 = 128s$$

$$Waste\ Time = \sum (Time\ allotted - Actual\ Time) = (57 - 47) + (23 - 7) + (48 - 12) = 62s$$

ITLMS – Rule-based approach

$$Wait/Cycle\ Time = \sum Time\ Alotted = 79 + 7 + 24 = 110s$$

$$Waste\ Time = \sum (Time\ allotted - Actual\ Time) = (79 - 47) + (7 - 7) + (24 - 12) = 44s$$

ITLMS – GA approach

$$Wait/Cycle\ Time = \sum Time\ Alotted = 78 + 10 + 32 = 110s$$

$$Waste\ Time = \sum (Time\ allotted - Actual\ Time) = (78 - 47) + (10 - 7) + (32 - 12) = 54s$$

	Total Wait/Cycle Time (s)	Total Space-time (s)	Mean Space-time (s)
Current System	128	62	20.7
ITLMS: Rules-Based	110	44	14.7
ITLMS: Genetic Algorithm	110	54	18.0

Table 2: Summary of Junction Attributes and Results

From the results in Table 2, the total wait time in each lane as well as the space-time incurred are significantly reduced using both methods of estimating green-light timing in ITLMS. The bias allotted takes into account the pedestrian crossings on this junction and allots appropriate time both for vehicles and pedestrians to pass the junction.

5 System Limitations & Future Work

At present, the demonstration system can accept manual user input for the parameters required for determining the traffic light timings. However, smart sensors such as video cameras and traffic junction cameras that are able to capture the road traffic demand and conditions can also provide real-time information into the system. By integrating sensor information into the system, this will allow the system to function on demand at traffic light junctions in an actual implementation system. When implemented with a sensing system, such as traffic junction cameras or pressure sensors, this system will also have to include a robust image classifier or a signal processing model, which will serve as input providers to the system.

Once such a technology is integrated to the system, the system can be further extended to provide the suggestion for pedestrian wait-times. This can work in conjunction with the traffic-light time optimiser to tune the parameters like *bias* in our system.

6 Conclusion

In this project, we develop a traffic light management system, that provides green-light timings to different phases at a traffic junction. The team gained the knowledge of implementing solution to a real-world problem by constructing an intelligent system. Using the rule-based system provided us to devise this reasoning system with relative ease. We learnt how to code the knowledge of a domain expert into a knowledge base. Using evolutionary computing technique like genetic algorithm, we learnt how to find an optimal solution from a pool of available solutions.

We modelled the system for accuracy and improvement of the existing traffic-light system and got positive results. The developed system is capable of reducing the wait times – to a notable extent, which helps in alleviating the issue of unnecessary waiting at the signals, hence, improving the lives of citizens.

From this module's lectures, we grasped the important techniques used for making a robust system and implementation of these techniques in this project gave us a thorough understanding & practical knowledge of present reasoning systems concepts.

7 References

[1] Land Transport Authority (LTA) Singapore, “Annual Vehicle Statistics 2018,” 2018. [Online]. Available: https://www.lta.gov.sg/content/dam/ltaweb/corp/PublicationsResearch/files/FactsandFigures/MVP01-1_MVP_by_type.pdf.

[2] A. Frakt, “Stuck and Stressed: The Health Costs of Traffic,” *New York Times*, 2019. [Online]. Available: https://www.nytimes.com/2019/01/21/upshot/stuck-and-stressed-the-health-costs-of-traffic.html.

[3] Land Transport Authority (LTA) Singapore, “GREEN LINK DETERMINING (GLIDE) SYSTEM,” 2018. [Online]. Available: https://www.lta.gov.sg/content/ltaweb/en/roads-and-motoring/managing-traffic-and-congestion/intelligent-transport-systems/green-link-determining--glide--system-.html.

[4] S. Araghi, A. Khosravi, and D. Creighton, “A review on computational intelligence methods for controlling traffic signal timing,” *Expert Systems with Applications*. 2015.

[5] T. S. Babicheva, “The use of queuing theory at research and optimization of traffic on the signal-controlled road intersections,” in *Procedia Computer Science*, 2015.

[6] M. Collotta, T. Giuffré, G. Pau, and G. Scatà, “Smart traffic light junction management using wireless sensor networks,” *WSEAS Trans. Commun.*, 2014.

[7] G. H. Kulkarni and P. G. Waingankar, “Fuzzy logic based traffic light controller,” in *ICIIS 2007 - 2nd International Conference on Industrial and Information Systems 2007, Conference Proceedings*, 2007.

[8] J. Alam, M. K. Pandey, and H. Ahmed, “Intellegent Traffic Light Control System for Isolated Intersection Using Fuzzy Logic,” *Conf. Adv. Commun. Control Syst. 2013*, pp. 209–215, 2013.

[9] E. Azimirad, N. Pariz, and M. B. N. Sistani, “A novel fuzzy model and control of single intersection at urban traffic network,” *IEEE Syst. J.*, 2010.

[10] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, “Simulation optimization: a review of algorithms and applications,” *Ann. Oper. Res.*, 2016.

[11] C. García-Martínez, F. J. Rodriguez, and M. Lozano, “Genetic algorithms,” in *Handbook of Heuristics*, 2018.

[12] M. Wiering, J. Vreeken, J. Van Veenen, and A. Koopman, “Simulation and optimization of traffic in a city,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2004.

[13] K. T. K. Teo, W. Y. Kow, and Y. K. Chin, “Optimization of traffic flow within an urban traffic light intersection with genetic algorithm,” in *Proceedings - 2nd International Conference on Computational Intelligence, Modelling and Simulation, CIMSim 2010*, 2010.

[14] J. J. Sánchez, M. Galán, and E. Rubio, “Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization,” in *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, 2004.

[15] B. Frederiksen, “Applying Expert System Technology to Code Reuse with Pyke,” in *PyCon*, 2008, pp. 1–26.

[16] Ayodeji Remi-Omosowon, “Pyeasyga Python Package,” *GitHub*, 2014. [Online]. Available: https://github.com/remiomosowon/pyeasyga.