International Science and Technology Journal
المجلة الدولية للعلوم والتقنية

Volume 32 العدد
April 2023 ابريل

المجلة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

Approaches to Traffic Light Systems Based on Using Machine Learning Techniques

Ramadan Amer Ali Emhamed[1], Osama Aboubaker Salem Abourodes[2], Ali Farij Kaeib[3]

1. Al Jabal Al Gharbi University, College of arts and sciences Computer department , Mizdah -libya,
2. Collage of Engineering Technology, Department of Electric and Electronic Engineering - Zwara, Libya
3. University of Sabratha, Electrical and Electronic Department Sabratha - Libya.

Email: ramadan.amer1133@gmail.com

الملخص:

نظام إشارات المرور هو أداة تنظم المركبات ذات العجلات من أجل العمل بانتظام. نظام إشارات المرور العادية الحالي ليس كافيًا للتحكم في الازدحام المروري بسبب أنه يعمل وفقًا لخطة زمنية محددة الوقت. ومع ذلك، فقد أصبح الازدحام المروري يمثل مشكلة كبيرة لمعظم البشر نظرًا لأنه يزيد من الضوضاء وتلوث الهواء وضياع الوقت. ومن ثم، فإن نظام إشارات المرور العادية لم يعد لديه القدرة على التعامل مع الازدحام الهائل، ويجب إيجاد حلول حول كيفية تنظيم المسار الذي يجب أن يسير أو يتوقف من أجل القضاء على ازدحام التقاطع بين المركبات.لذلك، تم تصميم نظام إشارات المرور الذكي المعتمد على تقنيات التعلم الآلي واكتشاف عدد المركبات تلقائيًا في كل حارة وضبط وقت الإشارة الخضراء على النحو الأمثل لتجنب وقت الانتظار الهائل والازدحام الواضح أيضًا بمعدل أسرع.أخيرًا، تم استخدام فيجول استديو و البايثون للمحاكاة, و خوارزمية (Branch and Bound) لتحديد أي منها يجب تشغيل الاشارة أو إيقاف تشغيلها، لأنه يتميز بخاصية إيجاد حل بين مجموعة من الحلول، ويتم تطبيق خوارزمية (greedy) لتحديد وقت الانتظار لكل مسار.

الكلمات المفتاحية: نظام اشارات المرور، وقت الانتظار، خوارزمية Branch and Bound, خوارزمية greedy, الازدحام المروري, الوقت الأخضرر.

Abstract:
Traffic light system is a tool that regulates wheeled vehicles to run regularly. Current ordinary traffic light system is not sufficient to control the traffic challenging congestions due to the fact they operate on a fixed-time length plan. However, the traffic congestion has grown to be a large trouble for most human beings due to the fact it increases noise, air pollution, and wasting time. Hence, ordinary traffic light system has no longer ability to deal with massive congestion, and solutions must be found on how to organize which lane must run or stop to eliminate intersection among vehicles. Therefore, the intelligent traffic light system based on machine learning techniques designed and automatically discover the quantity of vehicles in every lane and set the green sign time optimally to avoid massive waiting time and additionally clear congestion at quicker rate. Finally, the proposed system used Visual studio and Python for simulation and an algorithm of Branch and Bound used to determine which one the lamp should turn on and turn off, because it has a characteristic to find out a solution among some set of solutions, and greedy algorithm applied to determine waiting time for each line.
Keywords: traffic light system, waiting time, Branch and Bound, Greedy, traffic congestion, BFS, green time.

INTRODUCTION

Traffic congestion is one of the largest worries in nearly each country in the world, with roads designed to handle a certain amount of traffic, with the improvement of urbanization, the trouble of traffic congestion has attracted more and more attention, and the traffic congestion has grown to be a major trouble restricting city development.

However, the ordinary traffic lights are no longer that proper and it can also cause additional traffic delays, in traffic congestion problem, we have two major issues minimizing traveling time and maximizing safety [1].

This paper presented a sensor as eyes, the algorithms (B&B) Branch and Bound and greedy as brain that process the street situation, Branch and Bound algorithm it finds all of correct combination in every traffic light so that there are no collisions of traffic lanes[2].

Moreover, it is essential arrangement to determine which traffic lane that has the longer and the shorter waiting time, so that can prevent traffic jam and make function of these traffic lights effectively. For this reason, the most appropriate approach is using greedy algorithm, because this algorithm can make decision based on the most benefit condition, in this case is making decision about the shorter queue lane so that has shorter waiting time.[3]

In this paper, the B&B and greedy algorithms have combined to solve the problem we mentioned before, and hardware with a special specification has been customized to apply the theorems of logarithms that mentioned above.

LITERATURE REVIEW

There have been a lot of research addressed this problem here is a list of papers that related to this topic.

-(S.Mohanaselvi & B.Shanpriya, 2019) Developed an intelligent traffic light controller using fuzzy logic approach and demonstrated a control process using if – then rules, they developed a fuzzy logic controller to enhance the performance of the traffic signal controller. The number of arrived vehicles, queuing vehicles during different weather conditions have been considered for estimating the green light extension time is made by using the proposed fuzzy logic controller [4].

-(Luis Ramirez-Polo, 2021)Proposed intelligent traffic lights system that allow estimating the flow and finding what the best cycle times are according to the current state, being very useful in actuals moments in which due to confinement the rallying of people has altered at a veritably high rate. The developed models in this work, enhance the dwell time of the holding on vehicles at the different traffic lights of the two analyzed intersections. The simulation model has an enhancement of approximately 10 that could be found only by changing the times that the red and green colors of the traffic lights lasted, keeping the total cycle time (return to its initial color) the same for all traffic light classes at each interception.

-(Mamatha, 2018)They used image processing and machine learning approaches to automatically identify the number of vehicles in each lane to set the green signal time optimally to avoid large waiting time and clear congestion at faster rate. The YOLO darknet weights and labels are used with grouped classes to identify and determine number of vehicles in the frame of the captured image. They implemented the system by Raspberry pi 3B integrated with rotatable webcam, also they used IOT and Machine learning algorithm to determine the green traffic light duration based on traffic density. The image was captured using webcam for each lane and the image was subjected to YOLO detection algorithm to identify the number of vehicles based on which the green time was set. The detection was performed using the CNN with pre-trained YOLO coco weights [5].

-(Faraj & Boskany, 2020)designed an intelligent, low cost, and effective microcontroller circuit- based system for controlling automobiles in traffic light, it's able to dynamically accommodate timings of traffic signal. It can rapidly respond to traffic conditions to reduce traffic congestion. For applying this system, a server, microcontroller board, cameras, as hardware and wireless network between traffic lights as structure for communication are used. The system used machine learning approach (i.e., Yolov3 model and Open CV) for decision depending on existence of emergency cars and number of cars[6].

In the above papers the solution contains two main parts: eyes to watch the real-time street situation and a brain to process it.

RESULTS

In the searching by the B&B algorithm there is a useful function as a determinant of cost in each branch. For this case, the cost function is selection function whether in every x element there is another x which causes collision traffic lane. For example, if $x_1, x_2, x_3$are turned on (1), then onlyx_4 (1) may be turned on while another x is red (0).

Form of the tree that will be constructed is as follow, (1, 2):

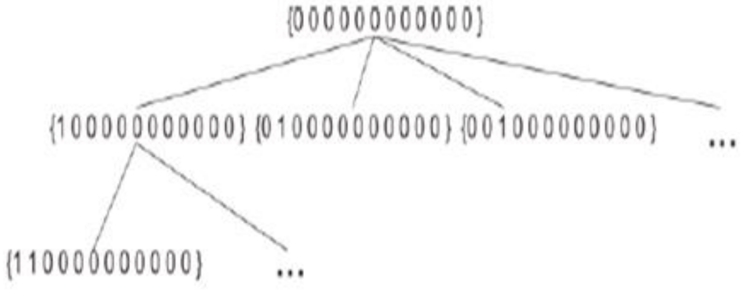

Figure 1. Image of the BFS tree road intersection

It can be seen in figure (1), searching solution on the tree will be begin with the condition that all light traffic must be turned to red.

Then, it is certain that the solution result from the tree will be reach on all of branches. Algorithm calculation is as follow:

Get number of cars in every lane from sensors: the algorithm has not implemented, just simulated, so The number of cars are entered manually as shown below in figure(2).

Figure 2. Get number of cars in every lane

Ordering the lanes according to the number of cars: After we have got the number of cars in each lane, we gave them rank and order them according to rank as shown in Table (1).

Table 1. Ordering the lanes

Rank	Number of Cars
E	65
A	52
G	12
C	8

Calculate the green time to every lane according to its rank: in this step we determined the green time and wait time for each rank, the lane that has high rank gets longer green time and shorter wait time as shown in table (2).

Table 2 Calculate the green time.

Rank	Number of Cars	Green Time
E	65	43 Sec
A	52	32 Sec
G	12	25 Sec
C	8	17 Sec

It can be seen from table (2) the priority of lane depends on its rank, this means that the order of switching the green light between lanes is not always fixed, it depends on lane rank as well.

Select green lanes (lanes that do not cause collide): in this step at first take the highest rank and select the lanes that do not cause accident as illustrated in figure (3).

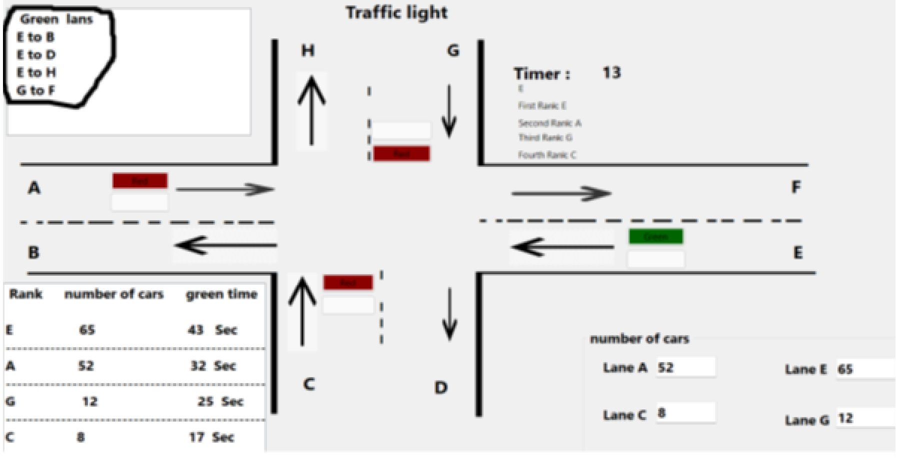

Figure 3. Select green lanes

Note that from the figure 3, lane (E) has the highest rank. Therefore, the green lanes are (E to B, E to H, E to D, G to F). Table 3 and Table 4 gave different inputs and showed system outputs.

Table 3. Experiment 1

Lane	Number of vehicles	System output	Green time	Wait time
A	52	E	43 sec	3sec
C	8	A	32sec	46sec
E	65	G	25sec	78sec
G	12	C	17sec	103sec

Table 4. Experiment 2

Lane	Number of vehicles	System output	Green time	Wait time
A	5	G	43 sec	3sec
C	15	E	32sec	46sec
E	25	C	25sec	78sec
G	35	A	17sec	103sec

Discussion

The analysis of the provided Figure 3, reveals a certain number of traffic lanes at the intersection, accessible to vehicles travelling in lanes labeled A, B, C, D, and others. The quantity of traffic lanes can be determined by employing the following formula:

$n = p \times (p - 1)$

Where:

n represents the quantity of traffic lanes.

p signifies the number of intersections.

Upon applying the formula to the given scenario, it becomes evident that there are a total of 12 traffic lanes resulting from the intersection depicted in the image[7]. This is calculated as follows:

$n = 4 \times (4 - 1) = 12$

Hence, there are 12 traffic lanes at the intersection, as derived from the calculation outlined above.

Figure 4. Image of traffic lanes from every car

In this system, the number of cars in each lane is entered. The algorithm then rearranges the lanes based on the number of cars and assigns a rank to each lane. This rank is used to calculate the waiting time (and green time as well) and to give priority to each lane.
For instance, a sample of an intersection can be taken. (See Figure 4) The solution can then be found using the B&B algorithm.
The B&B algorithm is a backtracking algorithm that is used to find optimal solutions to problems[8]. The algorithm works byrecursively searching the solution space, starting with the most promising node. The algorithm terminates when a node is found that is not a solution or when the entire solution space has been explored.
The B&B algorithm is a powerful tool for finding optimal solutions to problems. However, it can be computationally expensive, especially for large problems[9].
In this system, the B&B algorithm is used to find the optimal solution to the problem of finding the best way to rearrange the lanes

at an intersection. The algorithm takes into account the number of cars in each lane, the waiting time for each lane, and the green time for each lane. The algorithm then finds the solution that minimizes the total waiting time[10].

The B&B algorithm is a valuable tool for improving traffic flow at intersections. By using the B&B algorithm, it is possible to find the best way to rearrange the lanes at an intersection, which can lead to shorter waiting times for drivers.[11]

$$\{x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10} x_{11} x_{12}\ \}$$

The structure of answer through the usage of B&B algorithm is as follows:

Each element has a value that is,

0 = red traffic light

1 = green traffic light

Sequence of the solution is as follows:

x_1 = traffic light A To H

x_2 = traffic light A To F

x_3 = traffic light A To D

x_4 = traffic light C To B

x_5 = traffic light C To H

x_6 = traffic light C To H

x_7 = traffic light E To D

x_8 = traffic light E To B

x_9 = traffic light E To H

x_{10} = traffic light G To F

x_{11} = traffic light G To D

x_{12} = traffic light G To B

Table (3) shows in the system output column the order of lanes according to number of vehicles (E, A, G, C).
The structure of answer for lane E is:

{0 0 0 0 0 0 1 1 1 1 0 0}

The structure of answer for lane A is:

{1 1 1 1 0 0 0 0 0 0 0 0}

The structure of answer for lane G is:

{1 0 0 0 0 0 0 0 0 1 1 1}

The structure of answer for lane C is:

{0 0 0 0 1 1 1 1 0 0 0 0}

The system determines the green time for each lane by dividing the current cycle into time intervals. The time interval length for a cycle is assumed to be 117 seconds for simplicity. This means that the system checks the number of vehicles every 117 seconds. As shown in Table 3 and Figure 4, the lane with the highest rank receives the longest green time and the shortest waiting time. Therefore, the division of time intervals is not symmetric, as is the case with traditional traffic lights[12].
Here is a more detailed explanation of the process:

1. The system first calculates the number of vehicles in each lane.
2. The system then divides the current cycle into time intervals, based on the number of vehicles in each lane.

3. The system assigns the longest green time to the lane with the highest number of vehicles, and the shortest green time to the lane with the lowest number of vehicles.
4. The system thenselect green lanes (lanes that does not cause collide**)**.
5. The system then repeats steps 1-4 initiate a new cycle .

This process ensures that the lanes with the most vehicles receive the greenest time, which helps to reduce traffic congestion. It also ensures that the lanes with the fewest vehicles do not have to wait too long to turn, which helps to improve traffic flow[13].

The system takes into account the number of vehicles in each lane. In a traditional traffic light, each lane receives the same amount of green time, regardless of the number of vehicles. This can lead to traffic congestion, as the lanes with the most vehicles have to wait longer to turn.

As shown in Figure 4, each car has three possible destinations. For example, a car in traffic lane A can go to lanes H, F, and D, a car in traffic lane E can go to lanes B, H, and D. Therefore, it is necessary to arrange the traffic lights in a way that ensures that cars do not collide with each other. This can be done by considering which traffic lights cannot be turned on simultaneously and which traffic lights can be turned on simultaneously, as shown in Figure 3[14].

Here is a more detailed explanation of the process:

First, identify all of the possible combinations of traffic lights that cannot be turned on simultaneously. For example, in Figure 3, traffic lights A and B cannot be turned on simultaneously, because cars in lane A could collide with cars in lane B.

1. Next, identify all of the possible combinations of traffic lights that can be turned on simultaneously. For example, in Figure 3, traffic lights A and C can be turned on simultaneously, because cars in lane A will not collide with cars in lane C.

2. Finally, create a schedule for the traffic lights that ensures that no two traffic lights that cannot be turned on simultaneously are turned on at the same time. This schedule ensures that cars in lane A will not collide with cars in lane B or lane C.

This process can be used to create a schedule for any number of traffic lights. The more traffic lights there are, the more complex the schedule will be. However, the basic principles remain the same[15].

CONCLUSION

The solution of the branch-and-bound (B&B) algorithm and greedy algorithm methods can address the problem of avoiding traffic lane collisions in a traffic light arrangement, as well as how to regulate the waiting time for each traffic lane. However, the solution still has some weaknesses, such as the function of determining cost in the B&B algorithm in odd-intersection streets cannot work as well as in even-intersection streets. Additionally, there is no explanation of how to implement this method in a one-way street. Nevertheless, the basic problem of traffic light solving has been addressed in this paper with two algorithm solutions.

In this paper, we built a system that simulates a smart traffic light that can read the number of cars in each lane, order the lanes based on the number of vehicles, and find the green time and waiting time for each lane. The results of the system met the expected goals.

In the future, the system in this paper can be improved by using sensors or the YOLO software, which is the fastest object detection framework suitable for real-time applications. TheYOLO software can be used to determine the number of vehicles in each lane and then provide these numbers to the system for processing.

References

[1] Faraj, M. A., & Boskany, N. W. (2020). Intelligent Traffic Congestion Control System using Machine Learning and Wireless Network. UHD JOURNAL OF SCIENCE AND TECHNOLOGY, 123-131.

[2] PRASAD, V., ALI, U., ASSAF, M. H., DAS, S. R., & N, S. (2020). Real Time Intelligent Traffic Light System. *WSEAS TRANSACTIONS on SYSTEMS*, 20-25.

[3] Mamatha, A. (2018). Automated Traffic Light System Based on Image Processing and Machine Learning Techniques. *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, 27-32.

[4] S.Mohanaselvi, & B.Shanpriya. (2019). Application of Fuzzy Logic to Control Traffic Signals. *The 11th National Conference on Mathematical Techniques and Applications* (pp. 020045.1 - 020045.9). AIP Publishing.

[5] Luis Ramirez-Polo, M. A.-B. (2021). Simulation and Optimization of Traffic Lights for Vehicles Flow in High Traffic Areas. *International Workshop of Innovation and Technologies* (pp. 548-553). Leuven: Elsevier B.V.

[6] ‘A Greedy Branch-and-Bound Inclusion-Exclusion Algorithm for Calculating the Exact Multi-State Network Reliability’, Wei-Chang Yeh, NationalTsingHua University

[7] Egon Balas,‘BRANCH AND BOUND METHODS FOR THE TRAVELING SALESMAN PROBLEM’, Carnegie-Mellon University & Paolo Toth University

[8] Jens Clausen∗,‘Branch and Bound Algorithms - Principles and Examples’, 1999

[9] A. de Bruin, A. H. G. Rinnooy Kan and H. Trienekens, “A Simulation Tool for the Performance of Parallel Branch and Bound Algorithms”, Math. Prog. 42 (1988)

[10] J. Clausen, J. L. Tr¨aff, “Implementation of parallel Branch-and-Bound algorithms - experiences with the graph partitioning problem”, Annals of Oper

[11] J. Clausen and J. L. Tr¨aff, “Do Inherently Sequential Branch-and-Bound Algorithms Exist ?”, (1994).

[12] Bader, David A.; Hart, William E.; Phillips, Cynthia A. (2004). "Parallel Algorithm Design for Branch and Bound".

[13] David A. Bader ,William E. Hart,’PARALLEL ALGORITHM DESIGN FOR BRANCH AND BOUND’,University of New Mexico

[14] M. D. Hendy and D. Penny. Branch and bound algorithms to determine minimal evolutionary trees. Mathematical Biosciences,, 1982.

[15] Khattak, M. A. (2011). PLC Based Intelligent Traffic Control System. *International Journal of Electrical & Computer Sciences*, 69-73.