AN INTELLIGENT TRANSPORT SYSTEM FOR CONTROLLING TRAFFIC LIGHTS ON BUS-RAPID-TRANSIT (BRT) ROUTES IN JOHANNESBURG

MATTHEW BERNATH*, JONATHAN COUNIHAN* and REX VAN OLST[#]

*Final Year students (2007);
[#]Associate Professor and Project Supervisor;
School of Electrical & Information Engineering,
University of the Witwatersrand, 1 Jan Smuts Avenue, Johannesburg, 2000

ABSTRACT

An intelligent traffic light control algorithm that works off GPS and other data inputs from buses, which form a part of the planned (Bus Rapid Transit) BRT network in the city of Johannesburg, is created for a single four-way intersection. This algorithm minimises the time buses wait at intersections, advantaging buses at any intersection which lies on the bus route. A graphical simulator is developed to compare the bus priority algorithm to the normal case, with only synchronised lights and no oversight control. The system is designed to be modular, to allow further extension of the concept and to implement different algorithms and test cases. The results of this proof-of-concept are presented and compared to a single real-life case study. Recommendations for further work are also made.

1. INTRODUCTION

This paper presents and evaluates a design of an intelligent traffic light control algorithm to minimise the delay of BRT buses at intersections which fall on the designated BRT routes. In addition, the design of a simulator to assess the effects of such an algorithm is presented. The simulator allows one to assess the effect on the waiting times of the buses as well as the effect on cars nearby the intersection. The algorithm is deemed to be successful if the waiting times of buses can be substantially reduced with a minimum effect on private vehicles. A brief background of the problem can be found in Section 2, a determination of the problem specification can be found in Section 3, the tools used are described in Section 4 and the methodology employed is described in Section 5. Section 6 contains the test results and an analysis of the program developed, and Sections 7 and 8 deal with the use of the algorithm in real-world scenarios and ideas for improvement.

2. BACKGROUND

Mobility in Johannesburg

Mobility in Johannesburg is defined as the ability to move people and goods throughout the city [1]. In Johannesburg there is a current trend of a growing use of private cars. In 2003 alone, the use of private vehicles rather than public transport grew from 40% to 53% of total person trips [2]. One of the explanations for this is that by using private rather than public transport, citizens were saving up to 15 minutes per trip [2]. In addition, 24% of people using buses were dissatisfied due to the time it took the bus to complete their trip [2]. Thus the most important roads in Johannesburg were being overused. The need for efficient and quick public transport is critical in an attempt to reduce congestion on some busy arterials.

Current Traffic Solutions

The City of Johannesburg (CoJ) implemented the Strategic Public Transport Network (SPTN) which divided the city into a node-based public transport network [2]. The SPTN planned to direct routes into specific ‘corridors’, while providing links between these nodes. The aim was to enable public transport to be “clearly focussed” and connect the city more efficiently [2-3]. As part of the SPTN, the CoJ plans to implement a Bus Rapid Transit (BRT) network which will place 85% of Johannesburg residents within 500 metres of a bus route. Additionally, the CoJ has stated that it would like the traffic lights in Johannesburg to be optimised by 2008 [2].

Bus Rapid Transit

Bus Rapid Transit (BRT) is an optimised form of bus transport, often using new infrastructure to provide a service equivalent to that of rail transport [4]. Passengers using BRTs enjoy the advantages of special lanes reserved for BRT buses, stations well equipped to minimise the time spent buying tickets and boarding buses, and often improved access for people with disabilities [5]. BRTs are considered to be part of an Intelligent Transportation System (ITS) that tracks buses and provides vehicle arrival times [5].

The CoJ plans to have Bus Rapid Transport (BRT) systems implemented by the 2009 FIFA Confederations Cup and has already allocated R2-billion to the project [6]. The system is intended to also make an impact on the 2010 FIFA Soccer World Cup when the CoJ plans to have a 70%/30% split between public transport and private vehicles being used [7].

Intersections and Waiting Times

There are three dominant reasons for wanting to reduce the waiting times of buses at traffic lights. Firstly, by reducing the waiting times of buses over a number of intersections on the bus route, a substantial amount of time can be saved. Passengers may thus arrive at their destination early, while passengers who will experience difficulty in boarding the buses may have a sufficient amount of time to do so safely. This time will also reinforce the idea of buses being an optimised form of transport. Furthermore, should a bus be running late, intelligent traffic light control could help the bus maintain its original schedule. As stated, 24% of people using buses are dissatisfied with the trip times [2]. Thus, by reducing intersection waiting times, trip times can be significantly reduced.

Secondly, it has been found that fuel economy is best at cruising speeds, and acceleration uses the most fuel [8]. In addition, fuel is wasted when vehicles are stationary as the engine is still running. By using intelligent traffic light control, fuel economy can be improved in two ways. Firstly, by allowing buses to travel through green lights without stopping the buses do not need to decelerate, sit idling at traffic lights and then accelerate back to cruising speeds. Secondly, by reducing the amount of time spent by buses at red traffic lights, the fuel wasted while the bus is stationary is also decreased. By allowing buses to cross intersections without being caught in congested traffic, up to 3.78 billion litres of fuel per year can be saved [9].

Thirdly, advantaging buses at intersections illustrates that the buses have the priority over private vehicles. As the CoJ is planning to spend over R2-billion on implementing the BRT system, it is vital that the public make use of the buses. Allowing buses through busy intersections reinforces the idea that they are quick and efficient and could convince the public to use the available buses.

3. PROJECT SPECIFICATION

Problem Identification

The research problem was arrived upon in a meeting with the transportation management of the City of Joburg (CoJ) [10]. It was decided that with the introduction of highly efficient and 'intelligent' buses, an intelligent traffic light control algorithm was needed to ensure their efficiency. While the buses will be able to travel well using the normal features of BRT buses as defined by the Federal Transit Administration [11], they will be merged with normal traffic at intersections. It was found that if the waiting times at intersections could at least be minimised, the BRT buses could operate at maximum capacity. Finally it was decided that an algorithm was ineffective if it could not be adequately tested, and thus a simulator would be built to aid algorithm analysis. The project was given the name Intelligent Traffic Light Algorithm Simulator (ITLAS) to illustrate the combination of the algorithm and the simulator.

Project Constraint

Although the final simulator and algorithm would prove the concept using real-world data, thorough real-world testing would still be needed in order to verify the results gained from computer simulation. In addition, as the hardware to be installed in the buses was not available, the algorithm merely provides the code to be used in a communications module but does not interface with any hardware, be it GPS systems or traffic lights.

Project Assumptions

In order to develop Intelligent Traffic Light Algorithm Simulator (ITLAS), assumptions were made regarding the traffic light arrangements, hardware available and non-BRT traffic. Based on meetings with the CoJ, it was assumed that the BRT buses will be equipped with GPS and transmitters capable of transmitting their co-ordinates and speed. It was also assumed for the sake of private vehicles that any algorithm designed to benefit the BRT buses should have limits as to how much the other traffic at the intersections is affected. Finally, it was assumed that the algorithm could take complete control of the traffic lights at the intersection.

Current Intelligent Traffic Light Solutions

Although some forms of intelligent traffic light control do exist, none specific to BRTs have been implemented in a South African context [12]. Such solutions use motion sensors or loop detectors to sense weight [13]. In addition, these solutions were not designed to interface with a simulator and are stand-alone solutions. It was thus decided that the project would be based on such systems, but would not use any of the available code. Another solution is a 'queue jumper' lane which allows buses to jump delays at intersections [11]. These can cost up to R1.9-million per intersection [11].

4. TOOLS USED

Hardware

The BRT buses will be fitted with a GPS system [10]. The GPS will provide the speed and direction of the bus. Most GPS units are capable of calculating the speed and direction of a vehicle [14] and thus the GPS and transmitter would be able to operate independently of the bus speedometer, requiring only power from the bus.

A receiver is also required to receive the signals from the buses. The receiver runs software which constantly listens for bus signals. The receiver requires the exact position of the intersection to make decisions, and thus the code in each receiver is relatively unique (the intersection stored would be different, while the rest of the code would remain the same).

Software
ITLAS was programmed in Java. Java was used due to its native ability to handle threads [15], its ability to run on different platforms (should the CoJ run Unix/Linux/Windows) [16] and its similarity to C++ with which the developers were familiar. NetBeans IDE was used as the environment in which to program ITLAS. NetBeans provides Java novices with real time error checking and allows one to compile and run programs from within the Integrated Development Environment (IDE) [17].

5. METHODOLOGY EMPLOYED

ITLAS Algorithm
Overview: The BRT algorithm was designed to be used with buses which have the hardware as described in Section 4.1. The BRT algorithm implements intelligent traffic light control to minimise the delay of buses at intersections. For the detailed technical explanation of the algorithm refer to [18]. The algorithm is found in the ‘BRTAlgorithm’ class, which uses the ‘extendedGreenMode’, ‘reducedRedMode’, ‘normalMode’ and ‘Timer’ classes.

Modes of Operation: The algorithm runs in different modes depending on the presence of BRT buses and the speed and position of present buses. Threads are used throughout the algorithm and allow tasks, or code blocks to run simultaneously. They are used for three reasons. Firstly, they are used to simulate the use of a receiver which would run concurrently with the algorithm controlling the traffic lights. Secondly, they are used to avoid signals from the buses disturbing the normal operation of the traffic lights, especially when the signals will be discarded. Thirdly, threads are used in order to maintain a single object to run the traffic lights in a normal mode. When other modes are implemented, the thread in which the normal mode object runs is merely suspended and can be resumed once a special mode such as an extended green phase or reduced red phase has ended.

> *Normal Mode*: In normal mode (the code of which is in the ‘normalMode’ class), the traffic lights run through the green, orange and red cycles, maintaining each light for a predefined period of time. While in normal mode, the communications object is constantly listening for BRT interrupts – the co-ordinates, speed and direction signal that buses send. This time period between BRT interrupts can be varied, however it should send at least two interrupts per cycle. A cycle is defined as the time from when an intersection displays a green light for East/West until the next green light for East/West is shown. By sending at least two signals per cycle, the algorithm can have enough time to implement either extended green or reduced red to minimise waiting times.
>
> When a BRT interrupt has occurred, the algorithm calculates the time of arrival of the bus at the intersection. Due to the bus driving in an exclusive lane separate to other traffic, the bus is assumed to maintain a constant speed when approaching the intersection. The algorithm then assesses the current traffic light situation, and either extends a green phase, reduces the red phase at which a bus will arrive or does nothing. The algorithm will do nothing and continue in normal mode for one of two reasons: either the bus will arrive at a green light and pass through the intersection unassisted or launching extended green or reduced red phase will negatively affect other traffic beyond the limits allowed.

Extended Green Mode: Should a bus arrive within the time limit allowed for an extension of a green phase, the algorithm simply pauses the normal mode as soon as the required lights are green. Thereafter, the algorithm extends the green phase long enough for the bus to pass through the intersection without stopping and resynchronizes the light sequence. The normal mode is then resumed. The extended green mode has been decoupled from the algorithm and is found in the ‘extendedGreenMode’ class, and is thus available for reuse by other traffic engineers.

Reduced Red Mode: Should a bus arrive at a red light, the red phase can be reduced to the minimum amount of time required. The reduced red phase sequence begins when the required lights are red and the normal mode is paused. Thereafter, the bus will either wait for a reduced amount of time, or the lights will turn green in time for the bus to pass through the intersection without stopping. The lights are then resynchronised and normal mode is resumed. The reduced red mode has also been decoupled from the algorithm for reasons found above, and the code for this mode can be found in the ‘reducedRedMode’ class.

Resynchronisation: Should an intersection be currently synchronised with traffic lights at another intersection, extending green lights or reducing red lights can throw the traffic lights out of synchronisation. The traffic lights in Johannesburg are currently synchronised to ease rush-hour traffic [10]. After implementing a time-saving mode, it is critical to resynchronise the traffic lights to their original sequence. This is built into the extended green and reduced red modes. After an extended green, a reduced red is implemented for the direction of the bus to resynchronise the traffic light. After a reduced red, an extended green light is used. A flowchart illustrating the flow of the algorithm can be seen in Figure 1 below.

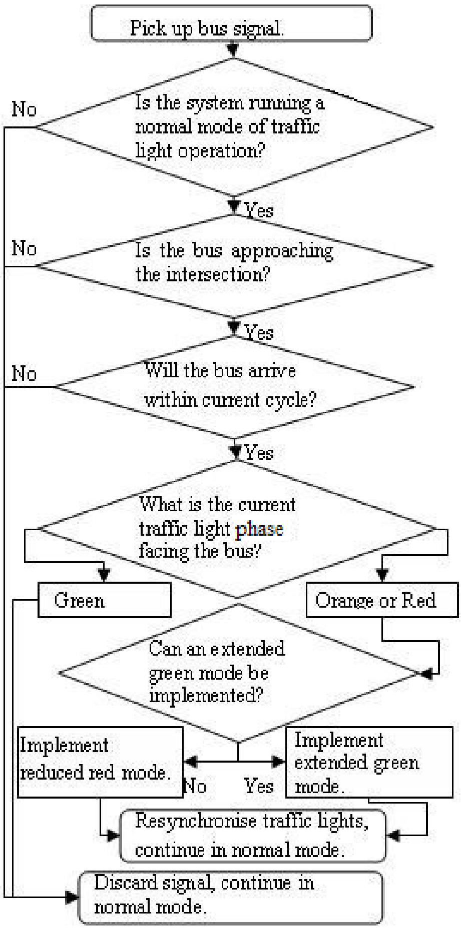

Figure 1 A flowchart illustrating the flow of the algorithm when a BRT interrupt occurs

BRT Interrupts

The signals which are emitted from the buses are referred to as BRT interrupts as they ‘interrupt’ the normal running of the traffic lights. They start the intelligent aspect of the algorithm. The interrupts contain an array of integers which transmit the data needed for the algorithm to make an intelligent decision regarding the buses and the intersection. The integers contained in the array represent the location of the bus, the speed and direction of the bus, the unique identification number of the bus and the length of the bus. The length of the bus is required as the GPS unit may be located at a certain position in the bus. A BRT bus may be up to 14 meters in length [11] and this may affect up to two seconds for traffic light time. This two second deficit may mean the difference between an extended green mode being implemented and a reduced red mode. Using the above information, up to 30 seconds (the average traffic light time in Johannesburg [19]) may be saved.

Communication

The code designed to sit in the receiver is found in the ‘Communication’ class [17]. This code receives BRT interrupts and, depending on the current mode of the algorithm, passes these interrupts to a ‘BRTInterrupt’ object to calculate which mode should be used to minimise the waiting times. From there, either the ‘extendedGreenMode’ or ‘reducedRedMode’ method is called from the ‘BRTAlgorithm’ class.

Timer

A special timer class, ‘Timer’, was created for the timing of the traffic lights. The timer can be given a specific time to count up to in seconds and can also be reset.

Limiting Effect on Other Traffic
Two limiting factors can be stored by the algorithm to avoid creating congestion at intersections. The two limiting times are the maximum amount of time by which a green light can be extended and the minimum amount of time allowed for a red light. After experimentation conducted by the authors, it was found that an extended green light should be at maximum 1.2 times the length of a normal green light. A reduced red light should be at minimum 0.7 times the length of a normal red light. These were found to be the optimum times to both allow buses through the intersection and limit the effect on other traffic.

Intersection Simulator
Overview: The intersection simulator was created to be an independent simulator to test various intelligent traffic light control algorithms. It was highly decoupled from the algorithm, and is thus capable of running any control algorithm created in Java. For use of a different algorithm refer to [20]. A more detailed description of the intersection simulator is contained in [21].

Graphical User Interface: The intersection simulator uses a graphical user interface (GUI) to create a visual depiction of the effect the algorithm has on the buses and traffic surrounding the intersection. The GUI also allows users to select which algorithm to use, and allows users to create buses on demand to test their algorithms. The time that has been saved and the current mode of the algorithm are also displayed.

Car and Bus Generation: Cars and buses are generated using a random number generator. One can alter the statistical distribution of cars and buses in this code to simulate a unique intersection. By collecting data from a specific intersection such as traffic light times, car arrival rates and bus arrival rates, one can generate a set of data closely resembling that intersection. ITLAS can thus be used to more accurately model real-world scenarios.

6. ANALYSIS OF METHODOLOGY AND RESULTS

Analysis of Methodology
The methodology allowed a complete algorithm and simulator to be developed. By limiting the analysis to one intersection, a simulator capable of interfacing with numerous algorithms was developed. The simulator also displays the running time of the simulation, so that the time saved may be accurately worked out. Furthermore, the simulator displays an accurate visual description of the effect of the algorithm. This has been found to be effective, especially in viewing buses in their BRT lanes.

The algorithm was developed to minimise waiting times of buses without adversely affecting other traffic. By this success criterion, the algorithm can be seen to be effective and a proof of the intelligent traffic light control concept. By proving that the concept of changing traffic lights based on the positions of buses is useful, the field of intelligent traffic light control can be further researched.

Although ITLAS has high-end requirements [22], in a real-world situation the algorithm would run by itself, which is not as process intensive as the simulator. Thus the requirements for the algorithm are far less stringent than the requirements for the entire ITLAS package.

Test Conditions
Due to the ability to create a fixed set of vehicles, the program can be subjected to numerous comparison based tests using both the algorithm developed and a normal mode of traffic light operation. However, the algorithm can access the traffic light timer and calculate how long a light will remain on its current colour. Thus, it can be calculated how long a bus would have waited at an intersection should an ‘intelligent’ mode not take over. This is the time that is saved and it is updated via the algorithm; consequently this time is never updated when the bus arrives at a green light as the algorithm is never called into operation.

ITLAS was tested using data taken at the intersection of Enoch Sontonga and Jan Smuts Avenues in Braamfontein, Johannesburg. The following figures were found after running the simulation for five hours and 46 minutes.

Table 1 Test Figures for ITLAS

Running Time:	326 minutes
Time saved by using the ITLAS algorithm:	27 minutes, 21 seconds
Buses generated:	155
Time saved per bus:	17 seconds
Bus arrival rate:	One every three minutes

Numerous other tests [19] yielded similar results, proving that the intelligent traffic light control concept does save a considerable amount of time. Assuming that there are a minimum of five intersections through which every BRT bus travels on its route and that the average time saved per bus is only 12 seconds, a minute can be saved per bus journey. These figures would be greatly exaggerated in real life; the Los Angeles BRT system includes 875 intersections on its routes [11].

7. USE OF SYSTEM

Considerations for Real-World Implementation
In Section 6, the algorithm has been proven to be successful using real-world data and thus should be implemented. Thorough testing should, however, be carried out to ensure that the algorithm functions correctly. In addition, the algorithm may cause a ‘chain effect’ whereby a bus may cause all of the intersections on the BRT route to operate either extended green or reduced red modes. This may have an adverse effect on other traffic. Such an anomaly, although highly unlikely, may hinder traffic flow.

A suggested implementation is to test the algorithm on one intersection on the BRT route. Should this prove successful, and even a small amount of time saved would indicate a success as that time would be significantly amplified over many intersections, the large scale implementation of the algorithm may be considered.

8. RECOMMENDATIONS FOR FURTHER WORK AND IMPROVEMENT

Bus Monitoring
Although unlikely, buses could change speed as they approach intersections. The current algorithm only takes a bus into account once. Future versions of the algorithm could adjust the traffic lights and then continue to monitor buses from the signals emitted. If, for

example, a bus slowed down and an extended green light was to be implemented, this could then be discarded and instead a reduced red light be implemented. Much of the code to monitor buses has been developed [22], but due to the limited time available this was not completed.

Handling Multiple Buses Simultaneously
Using the current assumption that two buses do not approach an intersection simultaneously, the algorithm has been designed to only accommodate one bus at a time. In future versions of the algorithm, traffic lights could be adjusted to accommodate two or more buses simultaneously. For example, the algorithm might find that doing a reduced red light for both buses saves more time collectively than doing an extended green light for one of the buses.

Accommodate Other Vehicles
Perhaps the largest area available for improvement is the ability of ITLAS to pick up signals from other vehicles. The data passed from buses to ITLAS contains a unique vehicle identification number. If, for example, a government assigns the numbers 001 to 100 to buses and the numbers 101 to 200 to emergency vehicles, ITLAS can make emergency vehicles a priority at any intersection. Drivers of such vehicles could merely turn the transmitter off when they do not require a priority. ITLAS could thus not only reduce bus waiting times, but also speed up the response times of emergency vehicles.

9. CONCLUSION

By combining information engineering and traffic engineering techniques, an intelligent algorithm to control traffic lights at intersections along BRT routes was developed. A simulator to test the algorithm and display the results graphically was also created. Both the algorithm and simulator were created in Java to enable them to run on any computer with the JRE installed.

During extensive testing of the algorithm and simulator, it was found that significant amounts of time could be saved by giving buses priority at busy intersections. This time would reinforce the idea of the new era of public transport being efficient and reliable.

Although the development of ITLAS was a success, the algorithm has yet to be tested in a real-world environment and would undoubtedly need refinement before being put into active use.

10. ACKNOWLEDGEMENTS

The authors would like to thank Peter Filbey (Joburg Roads Agency), John Kelly and Siveshni Reddy (City of Joburg: Transport Planning) for their advice and guidance throughout this student project.

11. REFERENCES

[1] Online TDM Encyclopaedia – Measuring Transportation, http://www.vtpi.org/tdm/tdm12.htm, Last Accessed 22 October 2007.

[2] City of Johannesburg, *Integrated Transport Plan 2003/2008, Volume 1*, May 2003

[3] New-look public transport system for Johannesburg, http://www.joburg.org.za/pr/2007/oct2b.stm, Last Accessed 11 October 2007.

[4] The BRT Policy Center, http://www.gobrt.org/, Last Accessed 20 October 2007.

[5] The BRT Policy Center, *Go BRT! High Quality Rapid Transit for the 21st Century,* Breakthrough Technologies Institute, Washington DC, March 2007.

[6] Joburg unveils efficient bus system, http://www.joburg.org.za/2006/nov/nov30_reavaya.stm, Last Accessed 20 October 2007.

[7] Matlhoko Consulting CC, *High Level Transport Plan for Nasrec Stadium for the 2010 World Cup*, October 2006.

[8] Fuel Economy, http://www.fueleconomy.gov, Last Accessed 21 October 2007.

[9] Study: Traffic Congestion Continues to Worsen in US, Wasting 2.9B Gallons of Fuel in 2005, http://www.greencarcongress.com/2007/09/study-traffic-c.html, Last Accessed 20 October 2007.

[10] J. Kelly, Personal Communication, CoJ, August 2006

[11] Federal Transit Administration, United States Department of Transportation, *Characteristics of Bus Rapid Transit for Decision-Making,* Office of Research, Demonstration and Innovation, August 2004.

[12] M. D. Bernath, "Appendix J: Literary Survey." 4th Year Project Report 07P29, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, October 2007.

[13] New York City Department of Transportation, *New York BRT Study,* New York State Department of Transportation, August 2006.

[14] Adaptive Technology Explained, http://www.nctec.co.za/adaptive_technology/gps_talking_navigation_systems.html, Last Accessed 23rd October 2007.

[15] Java Lessons: Concurrency, http://java.sun.com/docs/books/tutorial/essential/concurrency/, Last Accessed 23 October 2007.

[16] Java Runtime Environment, http://java.sun.com/j2se/desktopjava/jre/index.jsp, Last Accessed 25th October 2007.

[17] NetBeans IDE, www.netbeans.org/, Last Accessed 22 October 2007.

[18] M. D. Bernath, "Appendix F: Technical Details of Algorithm." 4th Year Project Report 07P29, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, October 2007.

[19] M. D. Bernath, "Appendix I: Testing of ITLAS." 4th Year Project Report 07P29, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, October 2007.

[20] M. D. Bernath, "Appendix H: Using an Alternative Algorithm with the ITLAS Simulator." 4th Year Project Report 07P29, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, October 2007.

[21] J. Counihan. "Appendix M: The Design Of An Intelligent Traffic Light Algorithm For Bus Rapid Transit (BRT) Intersections: Simulator Design." 4th Year Project Report 07G08, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, October 2007.

[22] M. D. Bernath, "Appendix K: Source Code and Attached Compact Disc." 4th Year Project Report 07P29, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, October 2007.