IMPROVING TRAFFIC CONTROL THROUGH SIMULATION AND OPERATIONS RESEARCH

by

ROBERT JOHN TAYLOR

Bachelor of Arts and Science, University of Lethbridge, 1971

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department

of

Mathematics

ACCEPTED
FACULTY OF GRADUATE STUDIES

[redacted]

DEAN

DATE 7 Apr 75

We accept this thesis as conforming
to the required standard

[redacted]

UNIVERSITY OF VICTORIA

FEBRUARY 1975

Supervisor: Dr. S. R. Clark

ABSTRACT

There are two major problems involved in traffic control in an urban area. The first is the adjustment of the individual signals to maximize the number of vehicles passing through the intersections in a given period of time. The second problem concerns the synchronization of the lights so that traffic can flow with minimal delay.

The problems of improving traffic control are dealt with in three stages. First it is shown that with only minor assumptions being made the problem of maximizing the traffic flow through the individual intersections can be treated as a linear programming problem. In the second stage, the problem of synchronizing the signals with each other to minimize delay is examined through the use of a computer model to simulate traffic flow along a roadway between two intersections. In the final stage several methods of synchronizing the signals are evaluated. The procedures developed are applied to data for the City of Victoria.

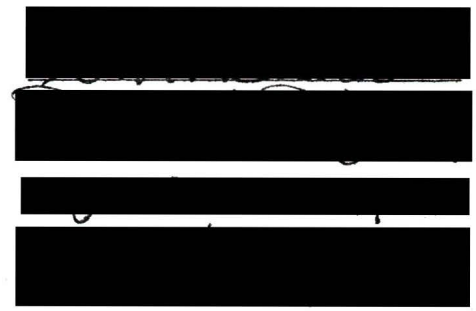


TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would especially like to thank Mr. L. Roberts and Mr. D. Wild of the City of Victoria for their cooperation in providing the necessary data on traffic flow. In addition, they were generous enough to allow the use of the methods that developed from this research in adjusting the traffic signals for the off-peak hours. As a result of the success of this test, the City is currently in the process of modifying the signals for the peak periods and it is hoped that this will produce a noticeable improvement in the traffic flow.

In addition, I would like to express my gratitude to Dr. S. R. Clark, chairman of my advisory committee, for his guidance and encouragement throughout the stages of this study. Thanks are also due to the other members of my committee for their helpful suggestions and comments, the staff of the University of Victoria Computing Centre for their assistance, and Mrs. B. Hames for her careful typing of the manuscript.

CHAPTER 1

INTRODUCTION

Recent growth of urban areas has placed tremendous strains on the transportation systems that serve them. There are two ways that the capacity of any such system can be increased: either the present system can be expanded; or, the efficiency can be improved. In the case of the network of roads in the core of a city expansion usually involves expensive and disruptive street widening. An alternative is to improve the efficiency by adjusting the signals that control the flow of traffic. We shall show that even minor adjustments to these signals can substantially effect the efficiency of the traffic system and so postpone the need for physical expansion.

There are essentially two major problems involved in traffic control in an urban area. The first is the adjustment of the individual signals to maximize the number of vehicles passing through the intersections in a given period of time. The second problem concerns the synchronization of the lights with each other so that traffic can flow smoothly and with minimal delay.

A signal controlling the flow of traffic at an intersection repeatedly goes through a set pattern of light changes. Green lights are displayed to some lanes of traffic allowing them to flow freely while other lanes are held back by red lights. Then the first lanes are halted and others allowed to pass through the intersection. This continues until all lanes have had a turn. Each separate combination of lights displayed to the traffic at an intersection is referred to as a phase of the signal. Thus, one phase may consist of green lights to both northbound and southbound traffic lanes while eastbound and

westbound vehicles are halted. This may be followed by a phase in which amber lights are displayed to northbound and southbound lanes so as to stem the flow of traffic from them. Hence, even the simplest of signals goes through a number of different phases. As the intersection becomes more complex (for example, by the inclusion of left turn only lanes and advanced or extended greens) the light pattern of the signal becomes much more complicated and the number of phases increase.

The complete set of phases of a signal constitutes a cycle of that signal. Thus, a given signal repeats a cycle of phases over and over. The time taken to pass through a set of phases is referred to as the cycle length of the signal.

A major problem facing traffic engineers is to determine the division of a cycle among the various phases to allow the maximum flow of traffic through the intersection. Traditionally this split was based upon the percentage of traffic utilizing the various lanes [13]. Thus, if northbound and southbound traffic constituted sixty per cent of the flow through an intersection they would be allotted a green phase for sixty per cent of the cycle length. This method is fairly crude and does not take into consideration either the number of lanes of traffic in each direction or the delays incurred if, say, numerous left turns are being made by the vehicles.

A second problem involving the traffic control signals is their synchronization by a master control so that traffic flows smoothly and with minimum delay through the entire street system. This master control regulates the time at which each signal begins its cycle. Thus, the signal at one intersection can start its cycle a number of seconds before the signal at another intersection. The difference

between the beginnings of the two cycles is referred to as the relative offset between the two signals. It is the objective of the traffic engineers to adjust these relative offsets so that traffic over the entire system is minimally delayed.

In the past, relative offsets have been set so that a hypothetical vehicle travelling at a constant rate of speed could progress through the signals without delay [14,11]. However, this method does not take into consideration the acceleration time required by the vehicle nor the influence of other traffic upon the time taken to traverse a block. Thus, while the signals may be adjusted so that a single vehicle travelling at a fixed rate of speed would have perfect progression through the signals, under more realistic traffic situations undue delays could result.

In this thesis we approach the problem of optimizing traffic signals in three stages. First, we show that with only a few assumptions being made, the problem of allotting a cycle among its phases can be treated as a linear programming problem. In the second stage, the problem of synchronizing the signals is examined through the use of a computer model to simulate traffic flow along a roadway between two intersections. In this way, large amounts of information about the delay that can be expected under various conditions can be gathered rapidly and conveniently without disturbing the actual system itself. Utilizing the model, we are able to describe the overall delay of the traffic in terms of the relative offsets of the signals involved. In the final stage several methods of synchronizing the signals are evaluated.

Chapter 2 is devoted to the explicit formulation of the problem of regulating the individual lights to maximize the traffic flow and the problem of synchronizing the lights to reduce delay. As the terminology of graph theory is particularly useful for formulating the problems, a brief introduction to the relevant terms is provided in the first section. The second section is devoted to the terminology of the signal system itself. The third and fourth sections deal with formal statements of the regulation of individual lights to maximize traffic flow (the cycle split problem) and the synchronization of the system to minimize the delay (the best offset problem). Historical background is provided in the final section.

The third chapter deals with reductions that are sometimes possible in the best offset problem. The first section shows how any signal in the system can be forced to have a fixed offset without changing the overall delay of the system. In the second section an algorithm is developed for totally solving the best offset problem under strict conditions. The third and fourth sections show how (under certain circumstances) the best offset problem can be simplified by decomposing it into a series of smaller problems. The final section of this chapter discusses some of the possible methods that could be used if no reductions to the problem are possible.

In order to investigate the nature of the delay in traversing a block between two signals, a computer model was developed. This model is described in the first section of Chapter 4 and supporting validation is given in the second section. The third section describes the typical pattern of delay that can be expected.

Chapter 5 applies the results of the previous chapters to the downtown signal system of The City of Victoria. The first section deals with the cycle split problem as it relates to Victoria while the best offset problem is examined in the second section.

Conclusions of this study and possible directions for further research are detailed in the final chapter. Program documentation appears in the appendices.

CHAPTER 2

FORMULATION OF THE PROBLEM

The terminology from graph theory provides an easy and concise way of formulating the problem of determining an optimal pattern of signals and so will be used extensively in this chapter. Most of the graph theoretic definitions are based on those of Frank Harary in his book Graph Theory [8]. The definitions used to describe the signal system itself are based on those used in a survey article by D.W. Ross [13].

2.1 Introduction to the Terminology of Graph Theory

By a graph G we mean a non-empty finite set $V(G)$ of elements called vertices together with a finite set $E(G)$ of unordered pairs of distinct vertices referred to as edges. We will usually denote a vertex by v_i $(i = 1,2,3,\ldots,n)$ and an edge by $[v_i,v_j]$. We say two vertices v_i and v_j are adjacent if $[v_i,v_j]$ is an edge of G. Two edges are said to be incident if they have a vertex in common. An isolated vertex is one which is adjacent to no other vertex.

Traditionally a graph is represented by dots for the vertices and by joining two dots by a line if they are incident.

Hence, if G is a graph with:

$$V(G) = \{v_1,v_2,v_3,v_4\} \text{ and}$$

$$E(G) = \{[v_1,v_2],[v_2,v_3]\}$$

G can be represented pictorially as in Figure 1.

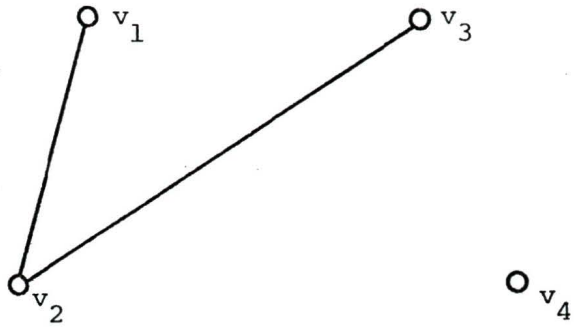
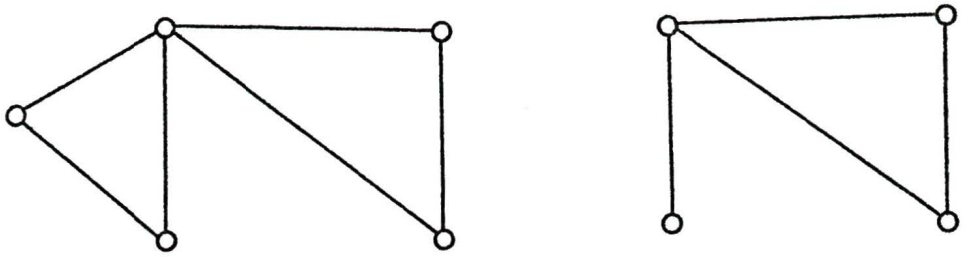


FIGURE 1. Pictorial representation of the graph G.

A subgraph H of a graph G is a graph with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. G is referred to as a supergraph of H. Hence the graph represented in Figure 2b is a subgraph of the graph represented in Figure 2a.



(a) (b)

FIGURE 2. A graph and one of its subgraphs.

For a graph G we define the graph $G' = G-\{v\}$ by:

$V(G') = V(G) - \{v\}$ and

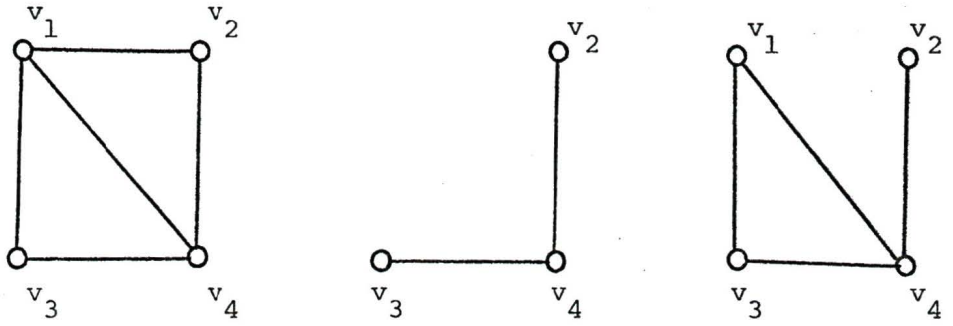
$[u_1,u_2] \in E(G')$ iff $[u_1,u_2] \in E(G)$, $u_1 \neq v$ and $u_2 \neq v$

For a graph G we define the graph $G' = G - [v_i,v_j]$ by:

$V(G') = V(G)$ and

$E(G') = E(G) - \{[v_i,v_j]\}$

Hence if G is represented as in Figure 3a then $G - \{v_1\}$ and $G - [v_1,v_2]$ can be represented by Figures 3b and 3c respectively.



(a) (b) (c)

FIGURE 3. A graph G, $G-\{v_1\}$, and $G-[v_1,v_2]$

A path P in a graph G is a subgraph of G with $V(P) = \{u_1,u_2,u_3,\ldots,u_k\}$ and $E(P) = \{[u_1,u_2],[u_2,u_3],\ldots,[u_{k-1},u_k]\}$ for some distinct u_i's in $V(G)$. We say the path P joins u_1 and u_k.

Thus if G can be represented as in Figure 4a the subgraphs represented in 4b and 4c are both paths while those in 4d and 4e are not.

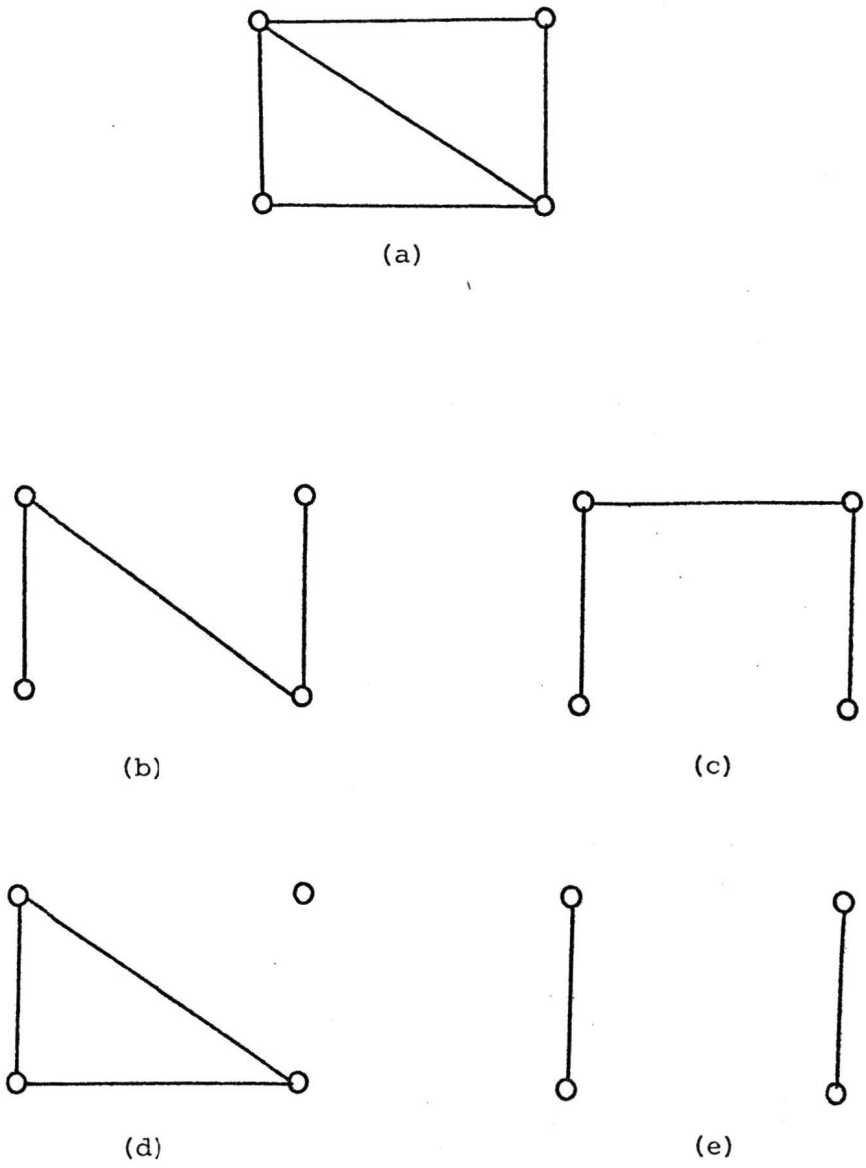
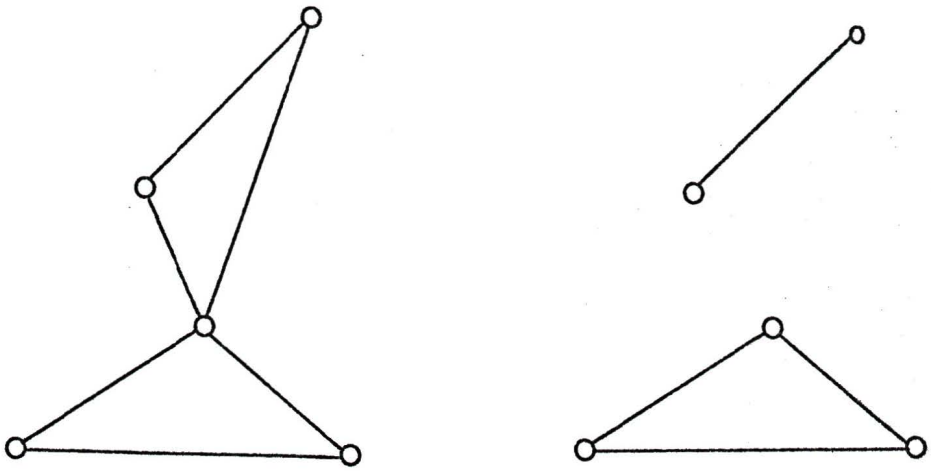


FIGURE 4. A graph G, two paths in G, and two subgraphs which are not paths.

A graph G is said to be connected if for any two distinct vertices v_i and v_j of G there exists at least one path joining v_i and v_j. Thus the graph represented in Figure 5a is connected while that in Figure 5b is not.

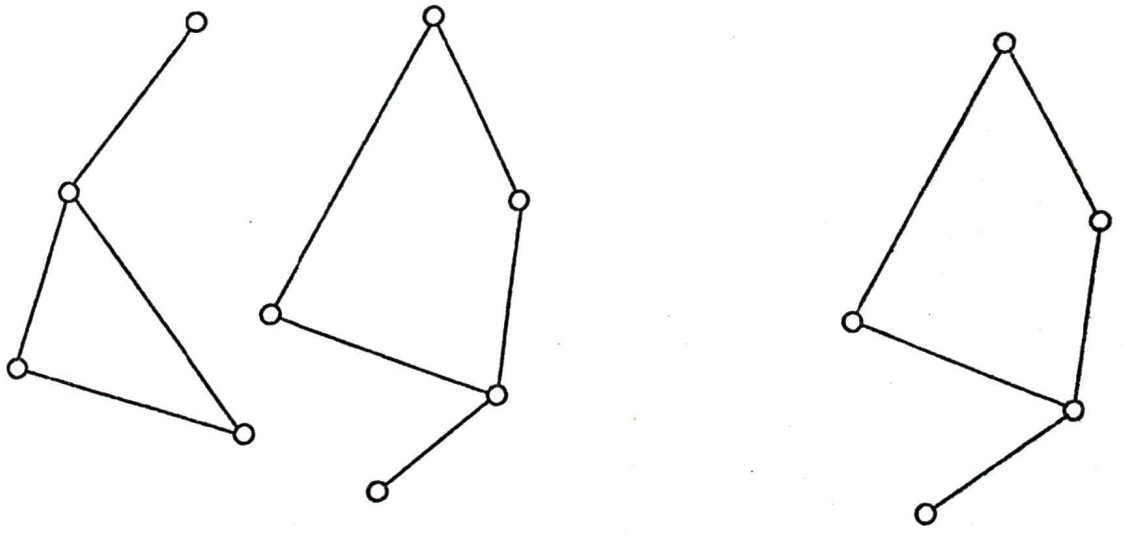


(a) (b)

FIGURE 5. A connected graph and a graph which is not connected.

A graph G is said to be a tree if for any two vertices v_i and v_j (v_i and v_j distinct) there exists one and only one path joining v_i and v_j.

A component C of a graph G is a connected subgraph of G such that there exists no vertex in $V(G)$ which is adjacent in the graph G to a vertex of C and yet is not in $V(C)$. That is, C is a maximal (with respect to the number of vertices) connected subgraph of G. The graph represented in Figure 6b is a component of the graph in Figure 6a.



(a) (b)

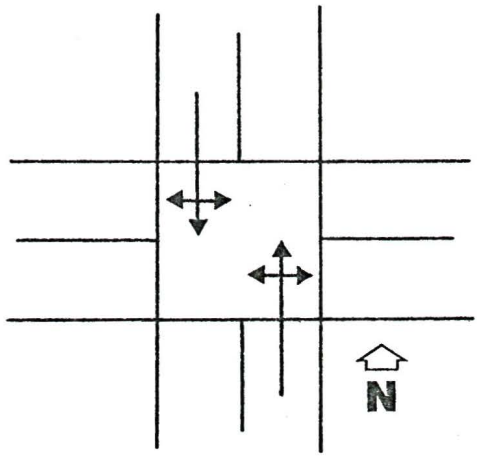
FIGURE 6. A graph and one of its components.

An edge $[v_i,v_j]$ of a graph G is called a separating edge if $G - [v_i,v_j]$ has one more component than G. If G contains a separating edge then we call G a separable graph.

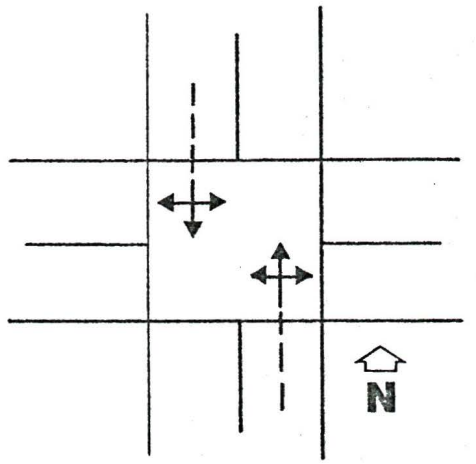
2.2 Introduction to the Terminology of the Signal System

In the type of street system with which we will be dealing each signal repeatedly goes through a sequence of phases during which certain traffic movements are allowed and others prohibited by displaying different coloured lights to the various lanes of vehicles.

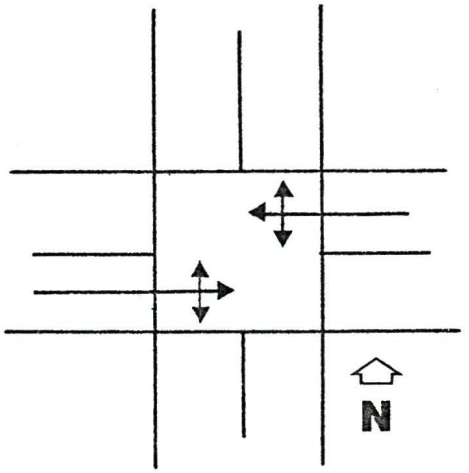
In Figure 7 a signal is represented which has four distinct phases (P_1,P_2,P_3, and P_4). If we number the lanes as in Figure 8 the phases may be summarized as in Table I.



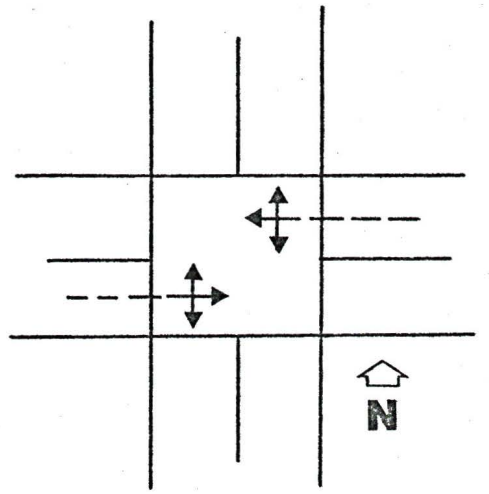
P_1: Green for northbound and southbound traffic.



P_2: Amber for northbound and southbound traffic.



P_3: Green for eastbound and westbound traffic.



P_4: Amber for eastbound and westbound traffic.

FIGURE 7. A simple intersection.

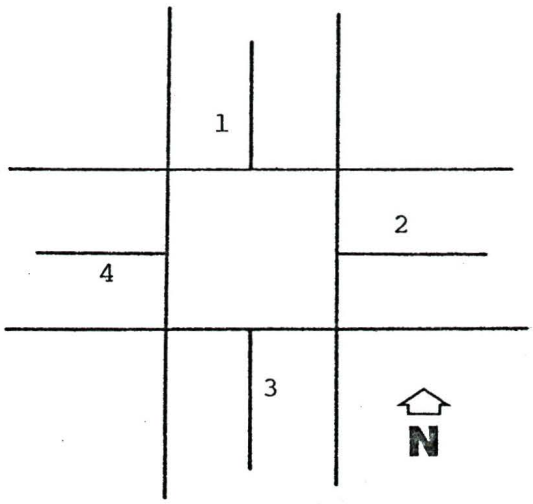


FIGURE 8. The numbering of the lanes of a simple intersection.

PHASE	LIGHT DISPLAYED TO LANE			
	1	2	3	4
P_1	G	R	G	R
P_2	A	R	A	R
P_3	R	G	R	G
P_4	R	A	R	A

G: Green

A: Amber

R: Red

TABLE I. Summary of the light pattern of a simple intersection.

Figure 9 represents a more complicated intersection involving left turn only lanes. Ten possible phases for this signal are summarized in Table II.

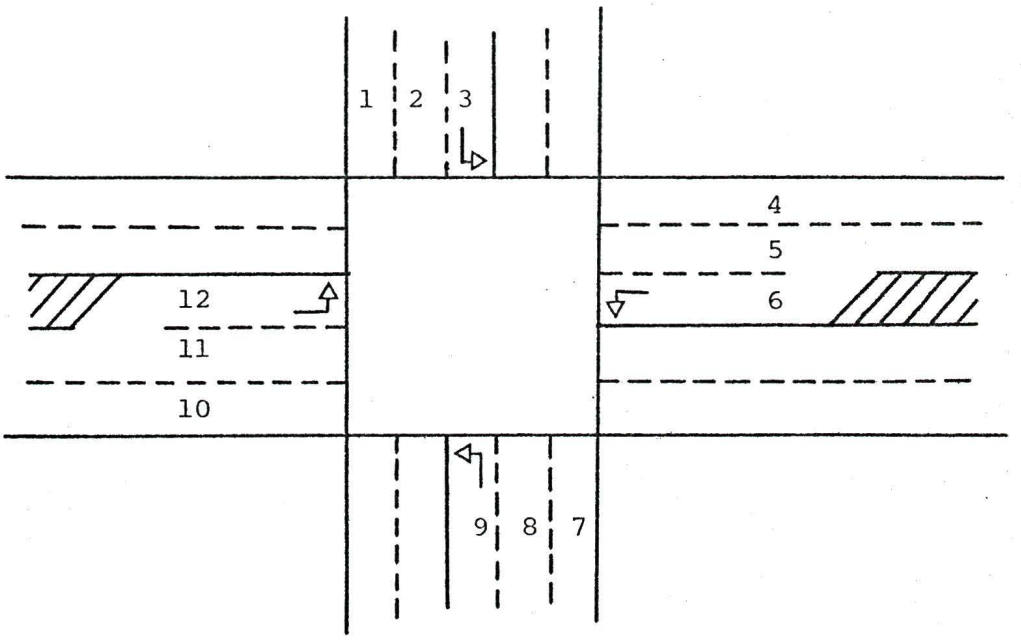


FIGURE 9. The numbering of the lanes of a more complex intersection.

PHASE	LIGHT DISPLAYED TO LANE											
	1	2	3	4	5	6	7	8	9	10	11	12
P_1	R	R	G	R	R	R	R	R	G	R	R	R
P_2	R	R	A	R	R	R	R	R	A	R	R	R
P_3	G	G	R	R	R	R	G	G	R	R	R	R
P_4	A	A	R	R	R	R	A	A	R	R	R	R
P_5	R	R	R	R	R	R	R	R	R	G	G	G
P_6	R	R	R	R	R	R	R	R	R	G	G	A
P_7	R	R	R	G	G	R	R	R	R	G	G	R
P_8	R	R	R	G	G	R	R	R	R	A	A	R
P_9	R	R	R	G	G	G	R	R	R	R	R	R
P_{10}	R	R	R	A	A	A	R	R	R	R	R	R

TABLE II. Summary of the light pattern of a more complex intersection.

We denote the length of time allocated to phase P_k by T_k. A repetition of a full sequence of phases is called a cycle while the time required to go through a cycle is referred to as the cycle length. The problem of determining how best to allocate the cycle length among the phases is the Cycle Split Problem. The allocation to the various phases can be easily represented as in Figure 10.

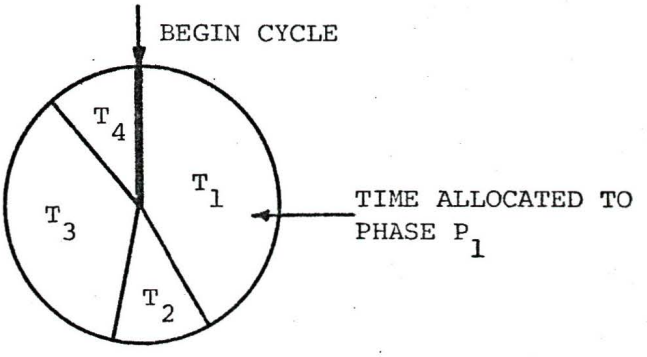


FIGURE 10. Allocation of the cycle to the phase.

In the system we will be dealing with, the cycle length of each signal is the same. Hence, we may talk about the cycle length C of a system of signals.

The signals of a light system are coordinated with respect to one another by reference to a master cycle. The offset of a signal is the difference in time between the beginning of the master cycle and the beginning of the signal cycle. Thus a signal with an offset of ten seconds begins its first phase ten seconds into the master cycle. This is illustrated in Figure 11.

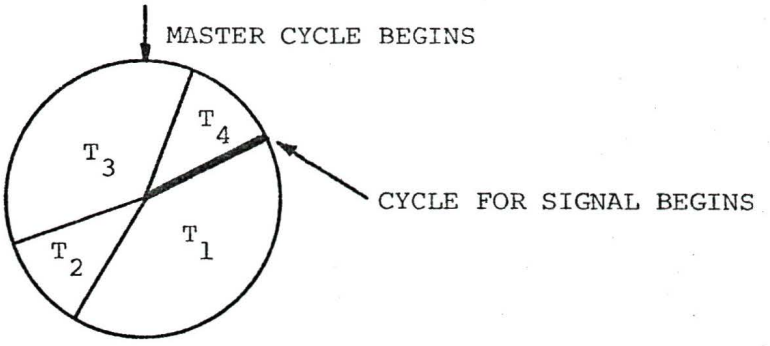


FIGURE 11. The offset of a signal.

A signal with offset equal to zero will be called a master signal. The problem of determining the set of offsets which will best serve to improve traffic flow is referred to as the Best Offset Problem (BOP).

We say two intersections are adjacent if there is a major roadway between the two. This roadway is sometimes referred to as a leg joining the two intersections.

We have now defined the terminology needed to formally state both the Cycle Split Problem and the Best Offset Problem. The Cycle Split Problem is formulated in Section 2.3 as a standard linear programming problem. The Best Offset Problem is formally stated in Section 2.4 and is dealt with in some detail in succeeding chapters.

2.3 Formal Statement of the Cycle Split Problem

The Cycle Split Problem may be formulated as a standard linear programming problem if certain assumptions about vehicle movements are made. We assume there is always a queue of vehicles waiting at an intersection (in effect assuming the intersection is working at capacity) and attempt to maximize the number of possible vehicle starts (S) through the intersection during a cycle.

Suppose the signal passes through phases $P_k (k=1,2,3,\ldots,m)$. Let T_k denote the length of time allocated to phase P_k. If we assume for each phase P_k, for each leg j $(j=1,2,3,\ldots,n)$ going into the intersection, and for each type of vehicle movement i $(i=1,2,3,\ldots,r)$; there exists a constant c_{ijk} corresponding to the largest possible number of vehicle starts during a second of type i from leg j during phase P_k we may formulate the Cycle Split Problem as follows:

$$\text{Maximize} \qquad S = \sum_{k=1}^{m} \sum_{j=1}^{n} \sum_{i=1}^{r} c_{ijk} \; T_k$$

subject to the constraints:

i) The sum of the times allocated to the phases must equal the cycle length, that is:

$$\sum_{k=1}^{m} T_k = C$$

ii) Each phase must receive a minimum allotment of time (for example: sufficient amber to clear the intersection or sufficient green to allow pedestrians to cross), that is:

$$T_k \geq a_k \qquad k=1,2,3,\ldots,m$$

iii) There must be a minimum amount of traffic flow of a given type passing through the intersection during a cycle from a given leg (for example: at least 25 vehicles per cycle must be able to turn left off the southbound leg), that is:

$$\sum_{k=1}^{m} c_{ijk} T_k \geq d_{ij} \qquad i=1,2,\ldots,r; \quad j=1,2,\ldots,n$$

2.4 Formal Statement of the Best Offset Problem

For a given street system, we construct an associated graph G as follows:

Let $V(G)$ be the set of all controlled intersections (labelled $v_1, v_2, v_3, \ldots, v_n$) and let $[v_i, v_j]$ be in $E(G)$ if and only if there is a uninterrupted leg between the two intersections. Figure 12 illustrates this construction.

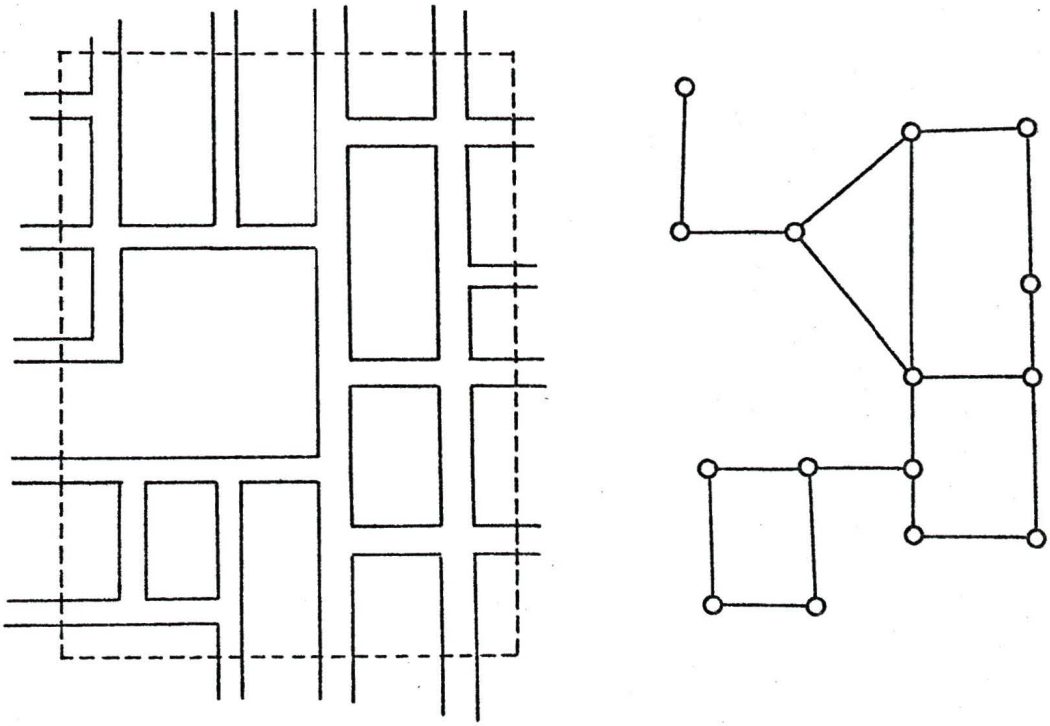


FIGURE 12. A street system and its associated graph.

As recent researchers in the field have done [13,1,9], we make the major assumption that with every edge $[v_i, v_j]$ of the graph G we can associate a function f_{ij} ($f_{ij}: R \to R^+ \cup 0$) which is a measure of the overall delay of vehicles using the leg as a function of the

difference between the offsets of the signals at either end of the leg.

We will also make certain assumptions about the nature of each of the functions f_{ij}. Because of the cyclic nature of the phases of each light, we assume f_{ij} is a periodic function with period equal to the cycle length of the system of lights. We will assume that the delay is always non-negative and that each f_{ij} obtains a minimum (denoted m_{ij}).

For notational convenience let $t_{ij} \in R$ such that $f_{ij}(t_{ij}) = m_{ij}$. Let θ_i denote the offset of signal v_i.

The Best Offset Problem may be formally stated as:

> Given a street system with associated graph G $(V(G) = \{v_1, v_2, v_3, \ldots, v_n\})$ and delay functions f_{ij}, determine offsets $(\theta_1, \theta_2, \theta_3, \ldots, \theta_n)$ which will minimize
>
> $$Z_G(\theta_1, \theta_2, \theta_3, \ldots, \theta_n) = \sum_{\substack{i,j \\ [v_i, v_j] \in E(G)}} f_{ij}(\theta_i - \theta_j)$$

As will be shown, Z_G is typically a very difficult function to minimize using standard methods. The next chapter is devoted to reducing the Best Offset Problem for a graph G to a series of smaller problems.

2.5 Historical Background

In the past, the allocation of time to the various phases that a signal repeatedly goes through has been in simple proportion to the traffic flow through the intersection [13]. In developments of late this has been augmented by simple simulation and "hill climbing" techniques designed to improve traffic flow through the intersections.[5]

The selection of offsets so as to synchronize the signals is somewhat more complicated. Hillier and Rothery [9] were among the first to consider the delay over a roadway between two signals as a function of the relative offsets of the signals.

A method developed under the auspices of the U.S. Federal Highway Administration [14] involves approximating the delay function by a series of parabolas. The optimization requires setting the derivatives of the overall delay function to zero and solving the resultant set of equations for a candidate minimum. This is repeated until one is "sufficiently certain that one has the global minimum or is at least close to it."

The first step in a method developed by the Road Research Laboratory [12] is to calculate a performance index based on the overall traffic delay and the number of vehicles stops on the leg for an initial set of signal offsets. These are altered by a fixed amount and the performance index is recalculated. This is done repeatedly as long as the performance index decreases. If it increases at any point, the search is made in another direction. This continues until a local minimum is obtained. The order in which the signals are to be adjusted must be stated. The City of Vancouver relies on a method very similar to that of the Road Research Laboratory written by J.C. Clapham [5].

The City of Victoria presently utilizes the more traditional method of selecting a number of roadways from the entire grid and setting the offsets of the signals so that a vehicle travelling at a fixed rate will have perfect progression through the lights. This method has the obvious advantage of not requiring extensive calculations, but the disadvantage that the overall delay may not be minimized.

This thesis approaches the problem of optimizing the signal system for the downtown core of the City of Victoria in three phases. In the first the problem of determining the optimal settings for individual lights is formulated as a linear programming problem. This approach was taken at the suggestion of J.C. Clapham [5].

In the second phase of the project a computer simulation of the traffic flow between two intersections is used to determine the characteristics of the delay functions. This model differs from those used previously in the field as it is somewhat more comprehensive. As suggested by Ross in his survey article, each delay function is then approximated by a sinusoidal function [13].

The third phase involves applying various function minimization techniques to the overall delay function to compare their effectiveness. Results show that the concept behind the traditional method used by the city (i.e. selecting a number of legs from the entire grid and optimizing over them) is satisfactory but the method of setting the offsets so that a hypothetical vehicle travelling at a fixed rate of speed would have perfect progression is shown to be unsatisfactory. By utilizing knowledge of the delay functions obtained through the computer simulation it is possible to adjust the offsets so that the delay is substantially reduced.

CHAPTER 3

REDUCTIONS TO THE BEST OFFSET PROBLEM

In this chapter we formalize some of the reductions known to traffic engineers and which can sometimes be applied to the Best Offset Problem. An algorithm for completely solving the problem when the graph involved consists of a set of components each of which is a tree or an isolated vertex is presented. A procedure for breaking the problem up into a series of smaller problems is also given.

Throughout this chapter we will assume (unless otherwise stated) that G is a graph with $V(G) = \{v_1, v_2, v_3, \ldots, v_n\}$. We also assume a delay function f_{ij} ($f_{ij}: R \to R^+ \cup 0$) is associated with every edge $[v_i, v_j]$ of G.

3.1 The Selection of the Master Signal

The first proposition allows us to add a constant to the offset of every light without changing the overall delay. This enables us to make any one of the signals a master signal (i.e. have a zero offset).

Proposition 1:

If R is any constant then

$$Z_G(\theta_1+R, \theta_2+R, \ldots, \theta_n+R) = Z_G(\theta_1, \theta_2, \ldots, \theta_n).$$

PROOF:

$$Z_G(\theta_1+R, \theta_2+R, \ldots, \theta_n+R) = \sum_{\substack{i,j \\ [v_i,v_j] \in E(G)}} f_{ij}((\theta_i+R)-(\theta_j+R))$$

$$= \sum_{\substack{i,j \\ [v_i,v_j] \in E(G)}} f_{ij}(\theta_i-\theta_j)$$

$$= Z_G(\theta_1, \theta_2, \theta_3, \ldots, \theta_n)$$

3.2 The Solution of the BOP if G is a Tree

The next two propositions enable us to solve the BOP if G is a tree (or a tree combined with isolated vertices). Proposition 2 gives us a method of extending (under certain circumstances) a solution of a subgraph of G to a larger subgraph of G while increasing the delay minimally. Proposition 3 justifies assigning offsets arbitrarily to isolated vertices.

Proposition 2:

If G' is a supergraph of G such that:

$$V(G') = V(G) \cup \{v_{n+1}\} \text{ and}$$

$$E(G') = E(G) \cup \{[v_k, v_{n+1}]\} \text{ for some } v_k \text{ in } V(G)$$

then $Z_{G'}(\theta_1,\theta_2,\theta_3,\ldots,\theta_n,\theta_k-t_{k,n+1}) = Z_G(\theta_1,\theta_2,\ldots,\theta_n) + m_{k,n+1}$ for any offsets $(\theta_1,\theta_2,\ldots,\theta_n)$.

PROOF:

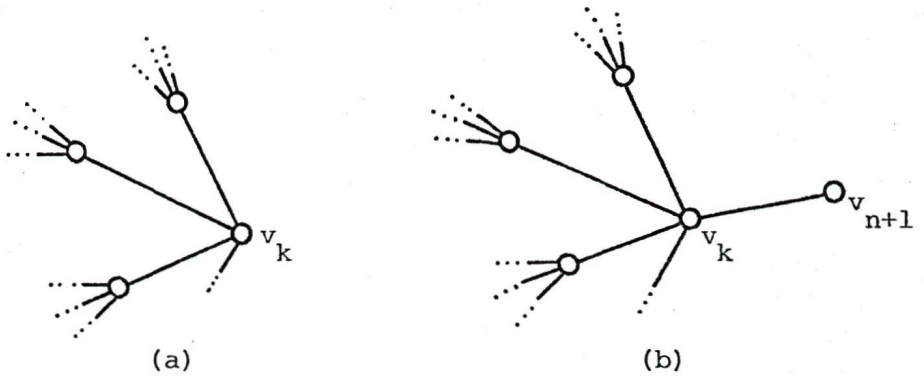


FIGURE 13. The graph G and the graph G'.

Let $\theta_i^+ = \begin{cases} \theta_i & \text{for } i = 1,2,3,\ldots,n \\ \theta_k - t_{k,n+1} & \text{for } i = n+1 \end{cases}$

$$\therefore Z_{G'}(\theta_1,\theta_2,\ldots,\theta_n,\theta_k-t_{k,n+1}) = Z_{G'}(\theta_1^+,\theta_2^+,\ldots,\theta_n^+,\theta_{n+1}^+)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G')}} f_{ij}(\theta_i^+-\theta_j^+)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G')-[v_k,v_{n+1}]}} f_{ij}(\theta_i^+-\theta_j^+) + f_{k,n+1}(\theta_k^+-\theta_{n+1}^+)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i-\theta_j) + f_{k,n+1}(\theta_k^+-\theta_{n+1}^+)$$

$$= Z_G(\theta_1,\theta_2,\ldots,\theta_n) + f_{k,n+1}(t_{k,n+1})$$

$$= Z_G(\theta_1,\theta_2,\ldots,\theta_n) + m_{k,n+1}$$

Proposition 3:

The offsets of isolated vertices may be chosen arbitrarily without affecting the overall delay.

PROOF:

Without loss of generality assume $v_1,v_2,\ldots,v_k$ are the isolated vertices. Let $H = G - \{v_1,v_2,\ldots,v_k\}$. We will show for any choice of θ_i $(i=1,2,\ldots,n)$ and θ_j' $(j=1,2,\ldots,k)$, $Z_G(\theta_1,\theta_2,\ldots,\theta_k,\theta_{k+1},\ldots,\theta_n) = Z_G(\theta_1',\theta_2',\ldots,\theta_k',\theta_{k+1},\ldots,\theta_n)$ by showing both are equal to $Z_H(\theta_{k+1},\theta_{k+2},\ldots,\theta_n)$.

Note that since $v_1,v_2,\ldots,v_k$ are isolated vertices $E(H) = E(G)$.

$$\therefore Z_G(\theta_1,\theta_2,\ldots,\theta_n) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i-\theta_j)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(H)}} f_{ij}(\theta_i-\theta_j)$$

$$= Z_H(\theta_k,\theta_{k+1},\ldots,\theta_n)$$

Let $\theta_i^+ = \begin{cases} \theta_i' & \text{for } i = 1,2,\ldots,k \\ \theta_i & \text{for } i = k+1,k+2,\ldots,n \end{cases}$

$$\therefore Z_G(\theta_1^+,\theta_2^+,\ldots,\theta_n^+) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i^+-\theta_j^+)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(H)}} f_{ij}(\theta_i^+-\theta_j^+)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(H)}} f_{ij}(\theta_i-\theta_j)$$

$$= Z_H(\theta_k,\theta_{k+1},\ldots,\theta_n)$$

In the case where G is a tree, or a tree combined with isolated vertices, Propositions 2 and 3 enable us to give an algorithm for determining $(\theta_1,\theta_2,\ldots,\theta_n)$ so that $Z_G(\theta_1,\theta_2,\ldots,\theta_n)$ is minimal, that is

$$Z_G(\theta_1,\theta_2,\ldots,\theta_n) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} m_{ij} .$$

We will assume G has at least one edge, otherwise G consists of isolated vertices and from Proposition 3 we know that the offsets can be chosen arbitrarily.

The algorithm is as follows:

STEP 1. Choose any edge $[v_r,v_s]$ in $E(G)$ and let $\theta_r = 0$ and $\theta_s = -t_{rs}$.

STEP 2. If all the edges have been chosen, set the offsets of the remaining (isolated) vertices arbitrarily to complete the algorithm, otherwise continue to STEP 3.

STEP 3. Choose any edge not already chosen but which is incident with a previously chosen one. Determine the offset of the vertex just added as in Proposition 2. Then go back to STEP 2.

Justification of the Algorithm:

STEP 1. If $[v_r,v_s]$ is in $E(G)$, $\theta_r = 0$, and $\theta_s = -t_{rs}$ then $f_{rs}(\theta_r - \theta_s) = f_{rs}(t_{rs}) = m_{rs}$ which is minimal.

STEP 2. If all the edges have been chosen, then the offsets of those vertices incident with at least one edge have been determined. The offsets of the remaining isolated vertices can be chosen arbitrarily by virtue of Proposition 3.

STEP 3. Any edge chosen from a tree so as to be incident with an already chosen edge must join a vertex which has not already been chosen to a vertex which has been chosen. The requirements of Proposition 2 are met and so the delay incurred by adding the new edge is minimal. Since the delay over each edge is minimal the overall delay is minimal. Hence if $(\theta_1,\theta_2,\ldots,\theta_n)$ are the offsets determined in this manner $Z_G(\theta_1,\theta_2,\ldots,\theta_n) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} m_{ij}$

We illustrate the algorithm by application to the graph G illustrated in Figure 14a. As per step 1 we choose any edge (say $[v_2,v_3]$) and set $\theta_2 = 0$ and $\theta_3 = -t_{23}$ yielding the graph in 14b. Since not all the edges have been chosen we apply step 3 choosing edge $[v_1,v_2]$ and setting $\theta_1 = +t_{12}$ thus yielding the graph in Figure 14c. Choosing edge $[v_3,v_4]$ and setting $\theta_4 = \theta_3 - t_{34} = -t_{23} - t_{34}$ we obtain the graph in Figure 14d. We apply step 3 once more setting $\theta_5 = -t_{23} - t_{35}$. Applying step 2 and setting $\theta_6 = 0$ arbitrarily, we finish with the graph illustrated in Figure 14e.

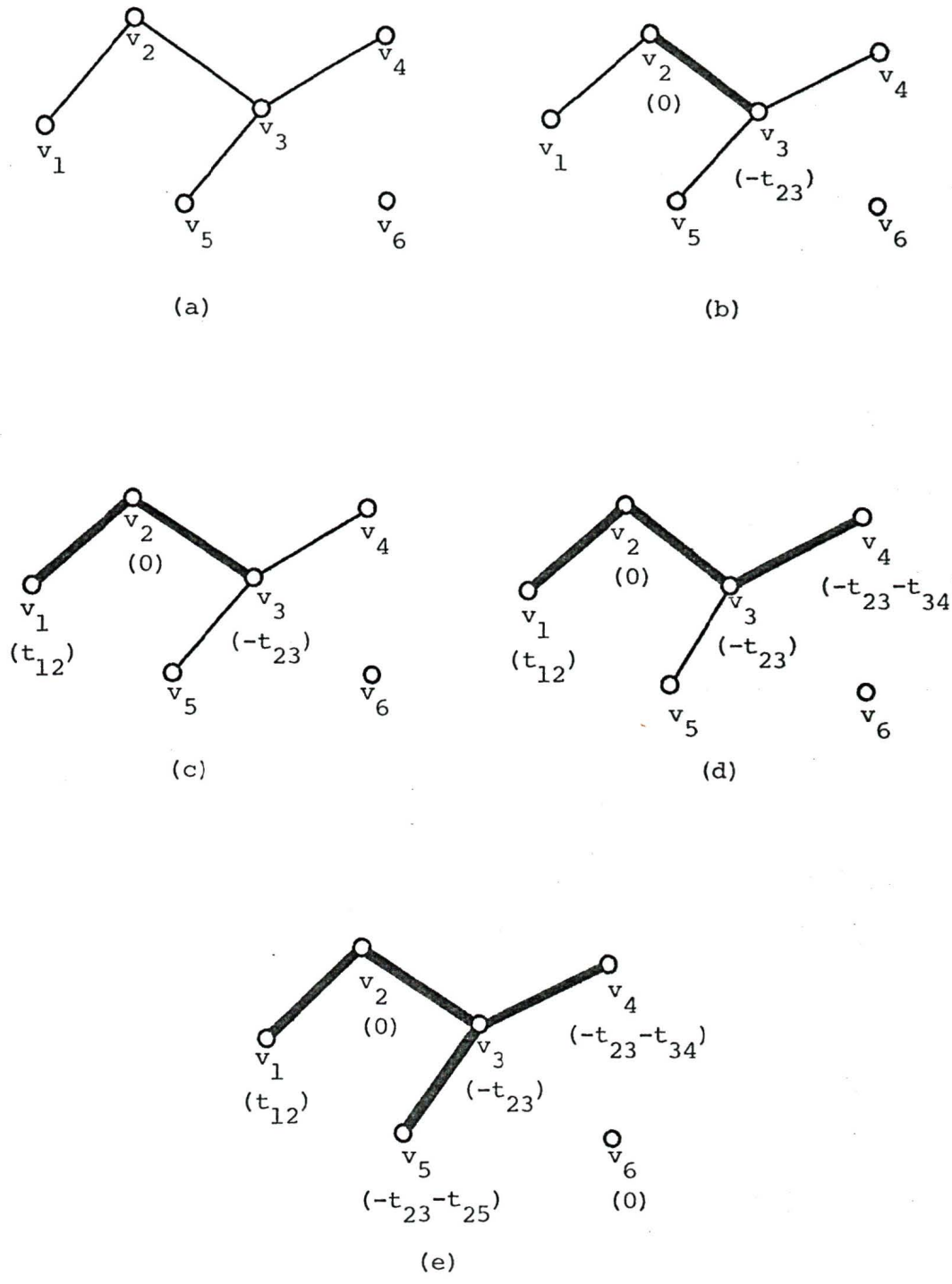


FIGURE 14. An example of the application of the algorithm for solving the BOP if the graph is a tree.

Hence,

$$Z_G(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i - \theta_j)$$

$$= f_{12}(\theta_1-\theta_2) + f_{23}(\theta_2-\theta_3) + f_{34}(\theta_3-\theta_4) + f_{35}(\theta_3-\theta_5)$$

$$= f_{12}(t_{12}) + f_{23}(-(-t_{23})) + f_{34}(-t_{23}-(-t_{23}-t_{34}))$$

$$+ f_{35}(-t_{23}-(-t_{23}-t_{35}))$$

$$= f_{12}(t_{12}) + f_{23}(t_{23}) + f_{34}(t_{34}) + f_{35}(t_{35})$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} m_{ij}$$

3.3 The BOP On Graphs With More Than One Component

The following proposition allows us to solve the BOP on a graph with a number of components by solving it for each component separately. Hence if G consists of a number of components which are trees we may optimally solve the BOP on G by solving it for each of the trees separately as in the previous section.

Proposition 4:

If G is a graph with components $C_1,C_2,\ldots,C_m$ then the overall delay in G is the sum of the delays in each of the components.

PROOF:

Without loss of generality let $V(C_k) = \{v_{i_{k-1}+1},\ldots,v_{i_k}\}$ for $k = 1,2,\ldots,m$.

$$\therefore Z_G(\theta_1,\theta_2,\ldots,\theta_n) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i - \theta_j)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in \bigcup_{k=1}^{m} E(C_k)}} f_{ij}(\theta_i - \theta_j)$$

$$= \sum_{k=1}^{m} \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_k)}} f_{ij}(\theta_i - \theta_j)$$

$$= \sum_{k=1}^{m} Z_{C_k}(\theta_{i_{k-1}+1},\ldots,\theta_{i_k})$$

The next proposition is a variation of Proposition 1 which allows us to make one signal in each component a master signal.

Proposition 5:

If G is a graph with components $C_1, C_2, C_3,\ldots,C_m$ then for each of the components C_k we can add any constant R_k to the offsets of the vertices of C_k without changing the value of $Z_G(\theta_1,\theta_2,\ldots,\theta_n)$.

PROOF:

Let $(\theta_1,\theta_2,\ldots,\theta_n)$ be a set of offsets and let $(\theta_1',\theta_2',\ldots,\theta_n')$ be the set of offsets obtained by adding R_k to θ_i if v_i is in $V(C_k)$ $(k = 1,2,\ldots,m)$.

$$\therefore Z_G(\theta_1,\theta_2,\ldots,\theta_n) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i - \theta_j)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in \bigcup_{k=1}^{m} E(C_k)}} f_{ij}(\theta_i - \theta_j)$$

$$= \sum_{k=1}^{m} \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_k)}} f_{ij}(\theta_i - \theta_j)$$

$$= \sum_{k=1}^{m} \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_k)}} f_{ij}(\theta_i+R_k) - (\theta_j+R_k))$$

$$= \sum_{k=1}^{m} \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_k)}} f_{ij}(\theta_i' - \theta_j')$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in \bigcup_{k=1}^{m} E(C_k)}} f_{ij}(\theta_i' - \theta_j')$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i' - \theta_j')$$

$$= Z_G(\theta_1',\theta_2',\ldots,\theta_n')$$

3.4 The BOP on Separable Graphs

Proposition 6 allows us to reduce the BOP on a separable graph G to two smaller problems and provides a method of combining the solutions of the two smaller problems together in a minimal manner thus forming a solution to the original problem.

Proposition 6:

If G is a separable connected graph with $[v_r,v_s]$ being the separating edge and C_1,C_2 the two components of $G - [v_r,v_s]$ then there exists a constant R which can be added to the offsets of one of the components to make the delay over the edge $[v_r,v_s]$ minimal.

PROOF:

Without loss of generality assume $v_r \in C_1$ and $v_s \in C_2$ and let $R = -t_{rs} + \theta_r - \theta_s$ where $(\theta_1,\theta_2,\ldots,\theta_n)$ is any set of offsets. Let $(\theta_1',\theta_2',\ldots,\theta_n')$ be the set of offsets obtained by adding R to the offsets of those vertices in C_2. We will show $Z_G(\theta_1',\theta_2',\ldots,\theta_n') = Z_{G-[v_r,v_s]}(\theta_1,\theta_2,\ldots,\theta_n) + m_{rs}$.

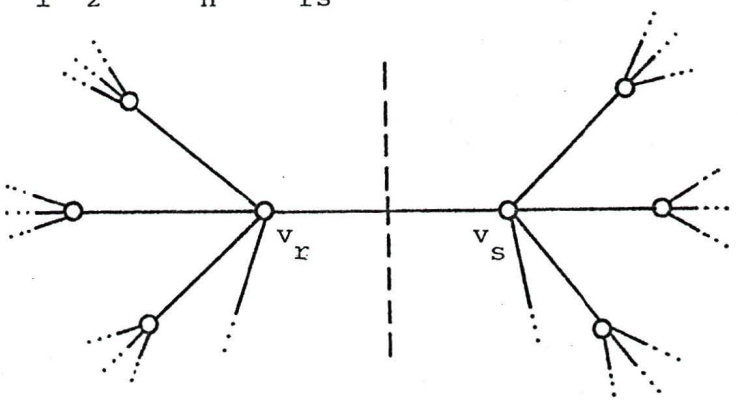


FIGURE 15. The separable connected graph G.

$$Z_G(\theta_1',\theta_2',\ldots,\theta_n') = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i' - \theta_j')$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_1)\cup\{[v_r,v_s]\}\cup E(C_2)}} f_{ij}(\theta_i' - \theta_j')$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_1)}} f_{ij}(\theta_i' - \theta_j') + \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_2)}} f_{ij}(\theta_i' - \theta_j')$$

$$+ f_{rs}(\theta_r' - \theta_s')$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_1)}} f_{ij}(\theta_i - \theta_j) + \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_2)}} f_{ij}((\theta_i+R)-(\theta_j+R))$$

$$+ f_{rs}(\theta_r - (\theta_s + R))$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(C_1)\cup E(C_2)}} f_{ij}(\theta_i - \theta_j) + f_{rs}(\theta_r-(\theta_s+R))$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)-[v_r,v_s]}} f_{ij}(\theta_i - \theta_j) + f_{rs}(\theta_r-(\theta_s+R))$$

$$= Z_{G-[v_r,v_s]}(\theta_1,\theta_2,\ldots,\theta_n) + f_{rs}(\theta_r-(\theta_s+R))$$

$$= Z_{G-[v_r,v_s]}(\theta_1,\theta_2,\ldots,\theta_n) + f_{rs}(t_{rs})$$

$$= Z_{G-[v_r,v_s]}(\theta_1,\theta_2,\ldots,\theta_n) + m_{rs}$$

Proposition 6 allows us to state the following algorithm for solving the BOP on a graph G in terms of solving it for a series of subgraphs of G.

> For a graph G, remove separating edges until the components are either: i) isolated vertices, ii) trees or iii) non-separable graphs. Solve the problem for each of the components, then reconstruct the original graph G by replacing the separating edges and making the delay over each of these minimal as in Proposition 6.

For example, given the graph G as in Figure 16 we may remove separating edges until we obtain the graph in Figure 17. The components of this graph can be solved separately and then the original graph G reconstructed as in Proposition 6.

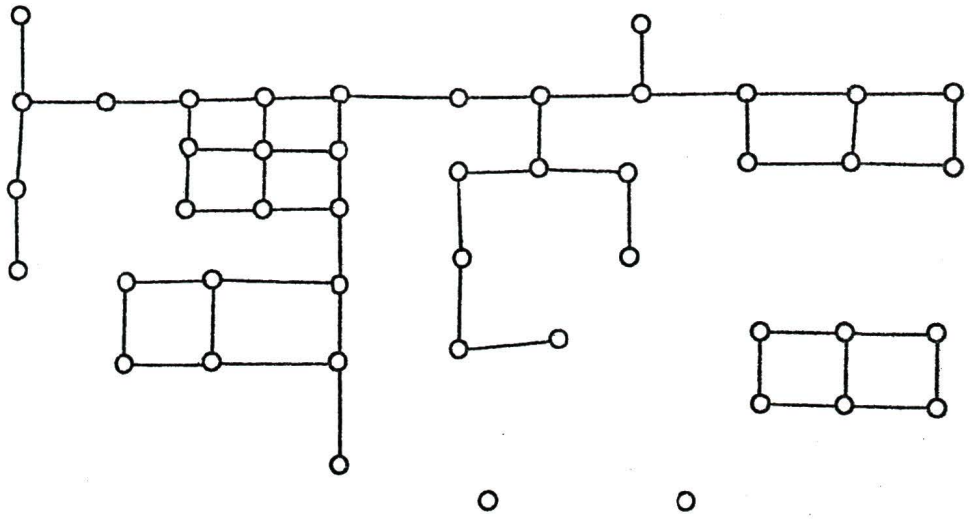


FIGURE 16. Graph of a street system.

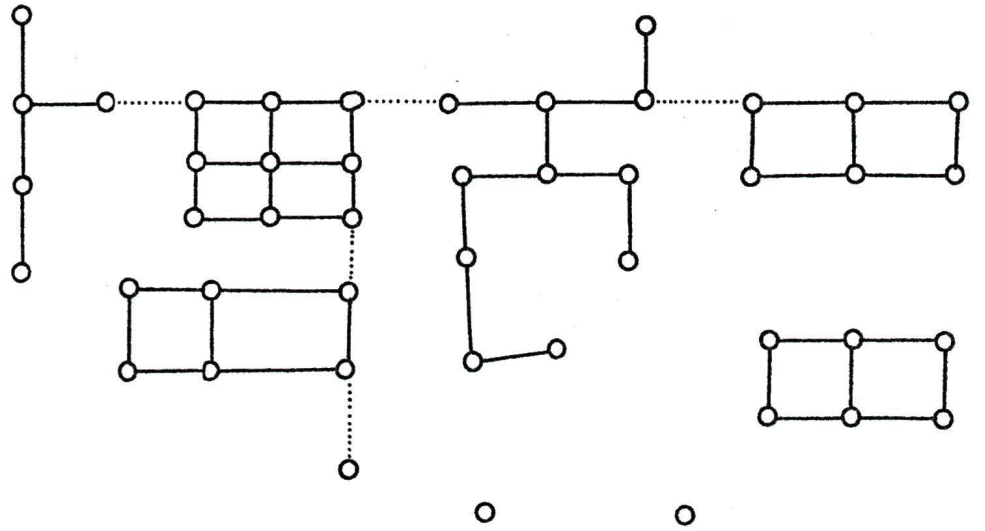


FIGURE 17. Graph of a street system with separating edges removed.

3.5 The BOP on Non-reducible Graphs

The problem remains of minimizing the overall delay function on graphs which are not separable, contain no isolated vertices and are not trees. The following is a brief discussion of possible ways of solving this problem.

a) The Davidon, Fletcher, Powell Method

This algorithm was originally proposed by W.C. Davidon [6] but did not become popular until 1963 when examined by R. Fletcher and M.J. Powell [7]. It is among the most efficient methods of function minimization currently available.

The central idea of the method is that by starting from a current point (x_i) a new point (x_{i+1}) is found by the relation

$$x_{i+1} = x_i - \alpha_i H_i g_i$$

where g_i is the gradient vector evaluated at x_i and H_i is the i^{th} approximation to the matrix of second partial derivatives. By differentiating the second order Taylor expansion of the objective function about the minimum it is possible to show that $-H_i g_i$ is a good direction in which to search while α_i prescribes the step to the minimum in this direction. The algorithm does not guarantee location of a global minimum - only a local one.

b) The Complex Method

This method is based on the "complex" method of M.J. Box [3]. The algorithm proceeds as follows:

1) An original complex of $K > N + 1$ points is formed consisting of a feasible starting point and $K - 1$ additional randomly generated points.

2) If at any time the explicit constraints are violated the point is moved a small distance inside the violated limit. If an implicit constraint is violated, the point is moved one half of the distance to the centroid of the remaining points.

3) The objective function is evaluated at each point. The point having the greatest function value is replaced by a point located further away from the centroid of the remaining points and on a line joining the rejected point and the centroid.

4) If a point repeats in giving the greatest function value on consecutive trials it is moved one half the distance to the centroid of the remaining points.

5) The new point is checked against the constraints and is adjusted as before if the constraints are violated.

6) Convergence is assumed when the values of objective function at each of the points are within a given tolerance for a number of consecutive iterations.

c) Optimizing Over Selected Trees

Another possible solution to the BOP on non-reducible graphs involves the selection of a subgraph which is a tree and which has as its vertices all the vertices of the non-reducible graph. Since this subgraph is a tree we can apply the algorithm of Section 3.2. That is:

STEP 1. Choose any edge $[v_r, v_s]$ in the tree and let $\theta_r = 0$, $\theta_s = -t_{rs}$ (Recall that the delay function f_{rs} for this leg takes on its minimum at t_{rs}). Hence,

$$f_{rs}(\theta_r - \theta_s) = f_{rs}(0-(-t_{rs}))$$

$$= f_{rs}(t_{rs})$$

$$= m_{rs} .$$

STEP 2. If all the edges of the tree have been chosen the algorithm is complete otherwise continue with STEP 3.

STEP 3. Choose any edge in the tree which has not already been chosen but which is incident with one which has. Determine the offset of the vertex just added as in Proposition 2 of Section 3.2. Go back to STEP 2.

The traditional method of approaching the BOP (used by the City of Victoria and numerous other cities) is a variation of this method. However, since no simulation is done, the traffic engineer must estimate the optimal relative offset (t_{rs}). This is done by choosing a t_{rs} so that a vehicle travelling at a fixed rate of speed would progress through the intersections with no delay. This method has the advantage of specifying the legs over which the delay is to be minimal and requires little computation to determine the offsets. However, as will be shown, the overall delay is significantly greater than it need be.

Section 2 of Chapter 5 gives some computational results from these methods as they apply to the Best Offset Problem for the downtown core of the City of Victoria.

CHAPTER 4

THE TRAFFIC DELAY MODEL

To investigate the nature of the delay function f_{ij} associated with an edge $[v_i,v_j]$ a model was developed to simulate the traffic flow along a leg between two intersections. This model is described in the first section of this chapter and supporting validation is given in the second section. The typical delay function f_{ij} is described in the last section.

4.1 Description of the Model

The model is designed to simulate traffic flow along a leg of a traffic system given various parameters about the traffic flow, physical layout of the leg, and the light patterns of the intersections at each end. By timing vehicles traversing the leg, the model yields a value of the delay function f_{ij} for a particular difference in offsets of the two intersections $(\theta_i - \theta_j)$. Stepping one of the offsets through an offset of 0 to C (the cycle length of the system) while maintaining the other at a value of 0 yields information on the nature of the function f_{ij} over its entire domain and so allows us to approximate it by a function g_{ij}.

As the system to be simulated is continuous in nature, a "time-slice" model was utilized rather than one based on critical events. During each time-slice and for each vehicle in the system, the future velocity, acceleration and position of the vehicle are calculated based upon its present status, the road system, the light patterns and the positions of the other vehicles. In addition, vehicles are added to the leg and removed as the need arises. The time taken by each vehicle to traverse

the leg is noted and when a required number have finished, the simulation for that particular relative offset ceases. The offset for one intersection is then advanced and the simulation is repeated with the new relative offset. This continues until the offset for the intersection has been stepped through the entire length of the cycle.

There are essentially six factors that determine whether a driver would slow down, increase his speed, or maintain his present speed:

i) The colour of the signal ahead. If it is red or amber and the vehicle is nearing the stop line, then the driver must slow down in order to stop at the light. If it is green then there is no need to slow down.

ii) The movement of the vehicle required at the intersection. If the vehicle is to turn in either direction then it must slow up to make the corner. If the vehicle is going straight then no reduction in speed is necessary.

iii) The distance to the vehicle directly ahead. If the distance to the vehicle ahead falls below a certain level the driver must reduce his speed so that if the vehicle ahead stops suddenly there will be no accident. If there is no vehicle ahead, or if the vehicle ahead is sufficiently distant, then there is no need for a reduction in speed.

iv) The distance to the vehicle in the left turn lane. If the driver is going to enter the left turn lane then not only must he maintain a reasonable distance to the vehicle directly ahead but also between himself and the vehicle in the left turn lane. This is also true if the vehicle in the left turn lane is blocking the lane the driver is in presently.

v) The distance to the vehicle in the right turn lane. As in the case of the left turn only lane the driver must maintain a reasonable distance to the car in the right turn lane if he is going to shift into that lane or if there is a vehicle in that lane blocking the present lane.

vi) The speed limit. If the previous five factors indicate no need for a reduction in the speed of the vehicle then the driver is free to accelerate up to the speed limit.

The following is a step by step description of the calculation of the future status of a vehicle given the present status of the leg and the vehicles on it.

STEP 1. Compute the velocity required by the vehicle at the next time slice due to the colour of the relevant signal (VEL1). If no speed reduction is required set VEL1 equal to the speed limit.

STEP 2. Compute the velocity required by the vehicle at the next time slice in order that it be able to perform the particular movement required at the next corner (VEL2). For example, for a right turn, the vehicle is assumed to come to a stop to allow pedestrian traffic to pass.

STEP 3. Compute the velocity required by the vehicle at the next time slice in order to maintain a reasonable distance to the vehicle ahead (VEL3).

STEP 4. Compute the velocity required by the vehicle at the next time slice in order to maintain a reasonable distance to

the last vehicle in the left turn only lane (VEL4). (This is appropriate only if the vehicle is going to shift into that lane or if there is a vehicle in the left turn only lane which is blocking the present lane.)

STEP 5. Compute the velocity required by the vehicle at the next time slice in order to maintain a reasonable distance to the last vehicle in the right turn only lane (VEL5). (This is appropriate if and only if the vehicle is going to shift into that lane or if a vehicle in that lane blocks the present lane.)

STEP 6. Compute the minimum (VW) of the values of VEL1, VEL2, VEL3, VEL4 and VEL5. This is the velocity required by the vehicle at the next time slice.

STEP 7. Compute the acceleration required (AW) to obtain the velocity of VW by the next time slice.

STEP 8. Reset (if necessary) AW to within upper and lower bounds so as to maintain realistic accelerations and deaccelerations.

STEP 9. Compute the velocity of the vehicle at the next time slice (FV) based upon the present velocity and AW.

STEP 10. Calculate the distance traveled by the vehicle.

STEP 11. Calculate the new position of the vehicle.

STEP 12. If the vehicle has passed through the leg continue with STEP 13 otherwise go to STEP 14.

STEP 13. Remove the vehicle from the system and record the time taken to traverse the leg. The iteration for this vehicle is complete.

STEP 14. If the vehicle is in a position to shift into the left or right turn only lanes and if such a shift is desired continue with STEP 15 otherwise go to STEP 16.

STEP 15. Relocate the vehicle in the left or right turn only lane.

STEP 16. If the vehicle has just entered the leg continue with STEP 17 otherwise go to STEP 18.

STEP 17. Record the time the vehicle entered the leg.

STEP 18. Record the position, velocity, acceleration, etc. of the vehicle for the next time slice.

STEP 19. End of iteration for one vehicle.

Vehicles may enter the simulation in two ways. At regular intervals (dependent upon the traffic volumes) vehicles are fed into the leg at the very ends of the lanes. This is in keeping with the assumption that a separate delay function exists for each leg. In addition, vehicles may enter the leg at one of the intersections thus simulating vehicles turning onto the leg from a side street. Whether a particular vehicle is to turn left or right, or proceed straight at the next intersection is randomly determined based on the proportion of vehicles making such maneuvers.

As described in Appendix B the model presently records the individual times required to traverse the leg. With slight modifications, the

simulation could also note the average velocity, amount of time vehicles spend not moving, the number of panic stops and similar measures of traffic flow. As the model is used to examine the delay functions f_{ij} it is sufficient for our purposes to note only the total simulated delay experienced by the vehicles traversing the leg.

As defined, the model simulates a leg with two way traffic and both left and right turn only lanes in both directions. By suitable choice of the parameters involved the model can also be used to simulate one way traffic as well as legs that do not have left or right turn only lanes.

In order to simulate one way traffic it is sufficient to specify that no vehicles enter lanes going in the other direction. To simulate a leg with no left turn only lane or no right turn only lane, the missing lane is given a length (say five feet) which always causes a vehicle in that lane to overhang into the adjacent lane thus simulating a single lane. This is illustrated in Figure 18.

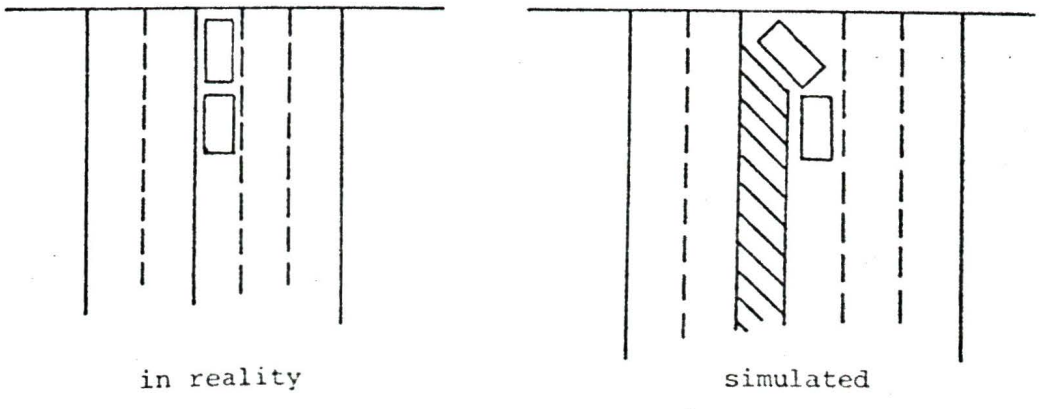


FIGURE 18. Simulation of a leg with no left turn only lane.

4.2 Validation of the Model

The most important characteristic of any model is its ability to simulate real life situations. A model may be extremely elegant and efficient but if it does not realistically portray that which it is supposed to, it is virtually useless.

As a first step in validating the FORTRAN implementation of the model described in the previous section, a portion of Hillside Street in Victoria was marked off at 50 foot intervals to the East of the signal line at Quadra. An additional checkpoint was located 25 feet from the line. For each checkpoint fifty vehicles were timed from when the light turned green until they reached the checkpoint line. The average observed timings are plotted against those predicted by the model in Figure 19.

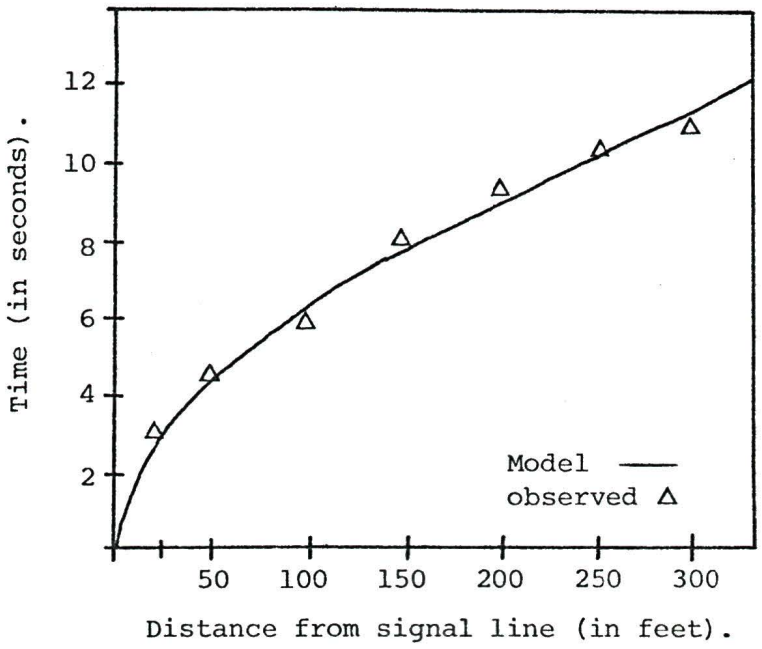


FIGURE 19. Acceleration away from a light.

In a similar fashion the times taken for vehicles to come to complete stops from given distances were also measured. The average observed timings are compared with the model in Figure 20. As can be seen, the model quite accurately simulates both acceleration from a light and braking to a stop.

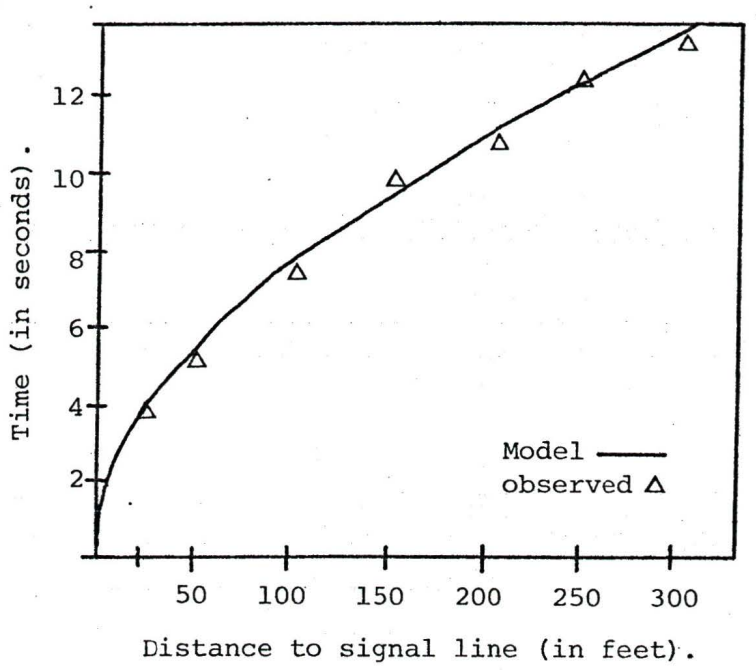


FIGURE 20. Deacceleration to a stop.

In order to validate the model under conditions including a number of vehicles and varying light patterns, fifty situations were observed and timed and were then simulated using the model. Figure 21 summarizes the observed and the model times.

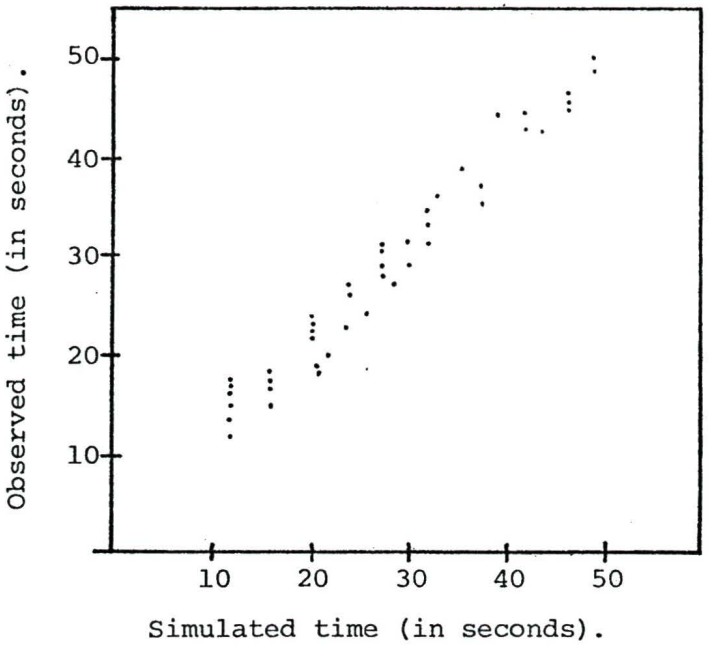


FIGURE 21. Observed vs. simulated times.

As a final check on the validity of the model a movie of the section of Douglas Street between Broughton and Fort was made. By replaying the film over and over it was possible to time all the vehicles traversing this block over a period of 10 minutes. The same block was simulated in the model with a similar flow of traffic. Figure 22 compares the two distributions of times taken to traverse the leg. A chi-squared test showed no significant difference between the two.

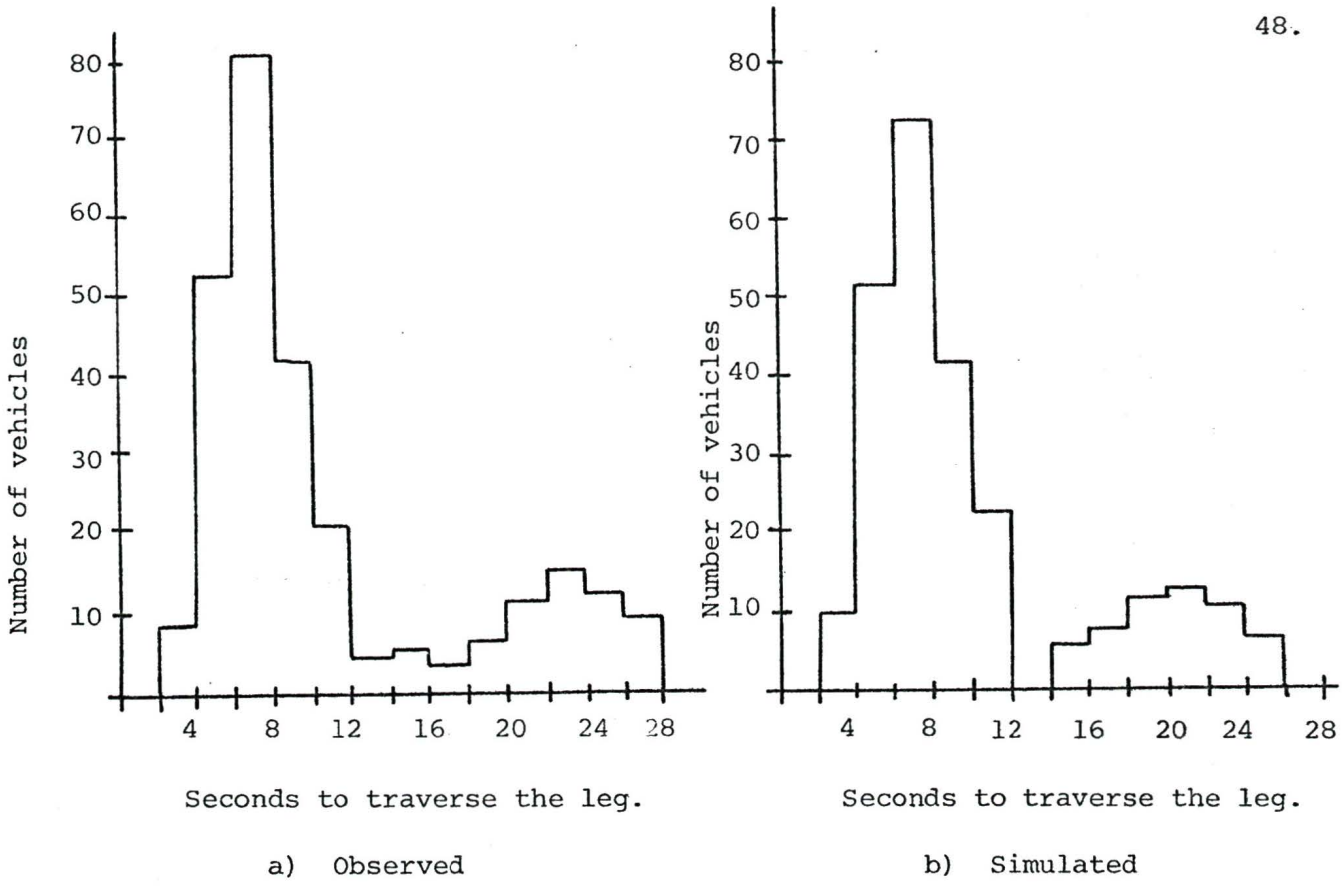


FIGURE 22. Distributions of times to traverse a leg.

4.3 The Typical Delay Function

Utilizing the computer model described it is possible to examine the delay function for the various legs in some detail. By maintaining the offset of one light of the leg at a value of 0 while stepping the other through an offset from 0 to C (the cycle length of the light) and plotting the average time taken to traverse the leg, we obtain a function similar to the one in Figure 23. To obtain the total delay function for the leg, it is necessary to multiply the average delay by the number of vehicles utilizing the leg.

The form of the delay function shown in Figure 23 is fairly sinusoidal in nature, which agrees with the delay functions observed or predicted by other authors [13,9].

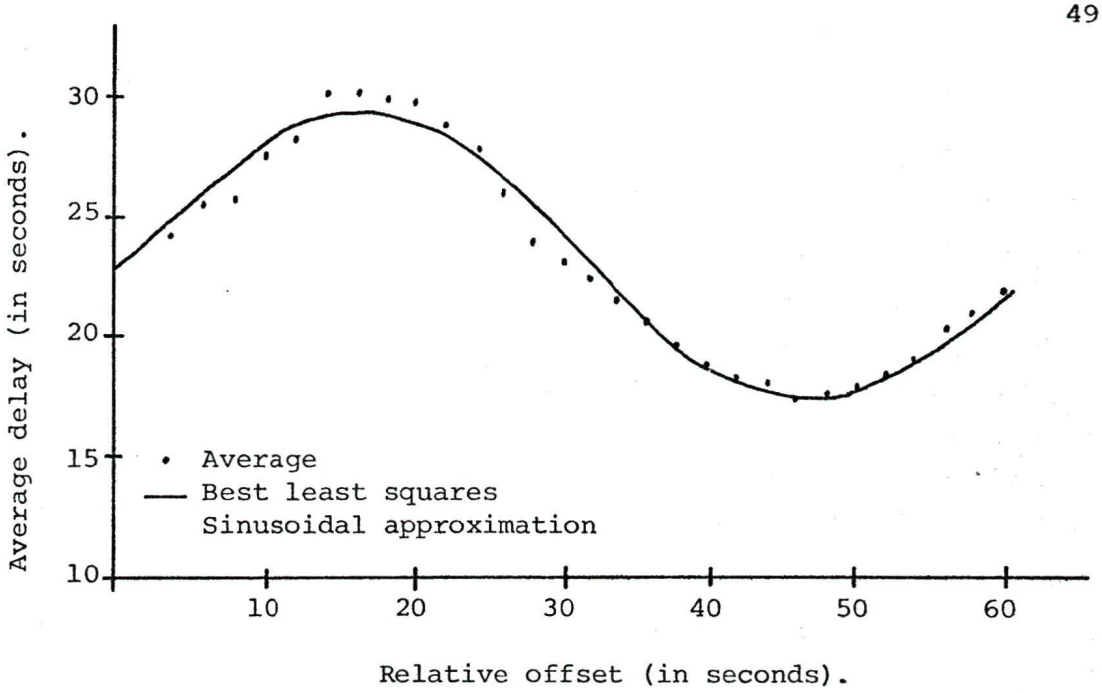


FIGURE 23. A typical average delay function.

Material in this section shows that not only is acceleration and deacceleration portrayed accurately by the model, but timings involving a number of vehicles in a variety of situations are comparable. Timings of vehicles traversing a leg in real life and the corresponding simulated timings produced distributions which were similar. On the basis of this evidence, it appears as if the model described in the first section of this chapter simulates real life situations accurately and so can be utilized in the investigation of the delay functions f_{ij}.

CHAPTER 5

APPLICATION TO THE CITY OF VICTORIA

Mr. D. Wilde and Mr. L. Roberts of the Traffic and Transportation Planning Department of the City of Victoria were very cooperative in providing traffic flow information for the downtown core of the City, thus enabling us to apply the theory described in the previous chapters to an actual urban situation. This was done in three separate stages. First, the linear programming technique described in Section 2.3 was applied to the data to optimize the allotment of time to the phases of the individual signals. Then, based on these phases times, the computer model outlined in Section 4.1 was utilized to determine the delay function for each leg of the test area. Finally, a number of different strategies for minimizing the overall traffic delay were compared.

The data provided covered most of downtown Victoria as shown in Figure 24 for the afternoon rush period. The associated graph, constructed in consultation with the Traffic and Transportation Planning Department, is illustrated in Figure 25.

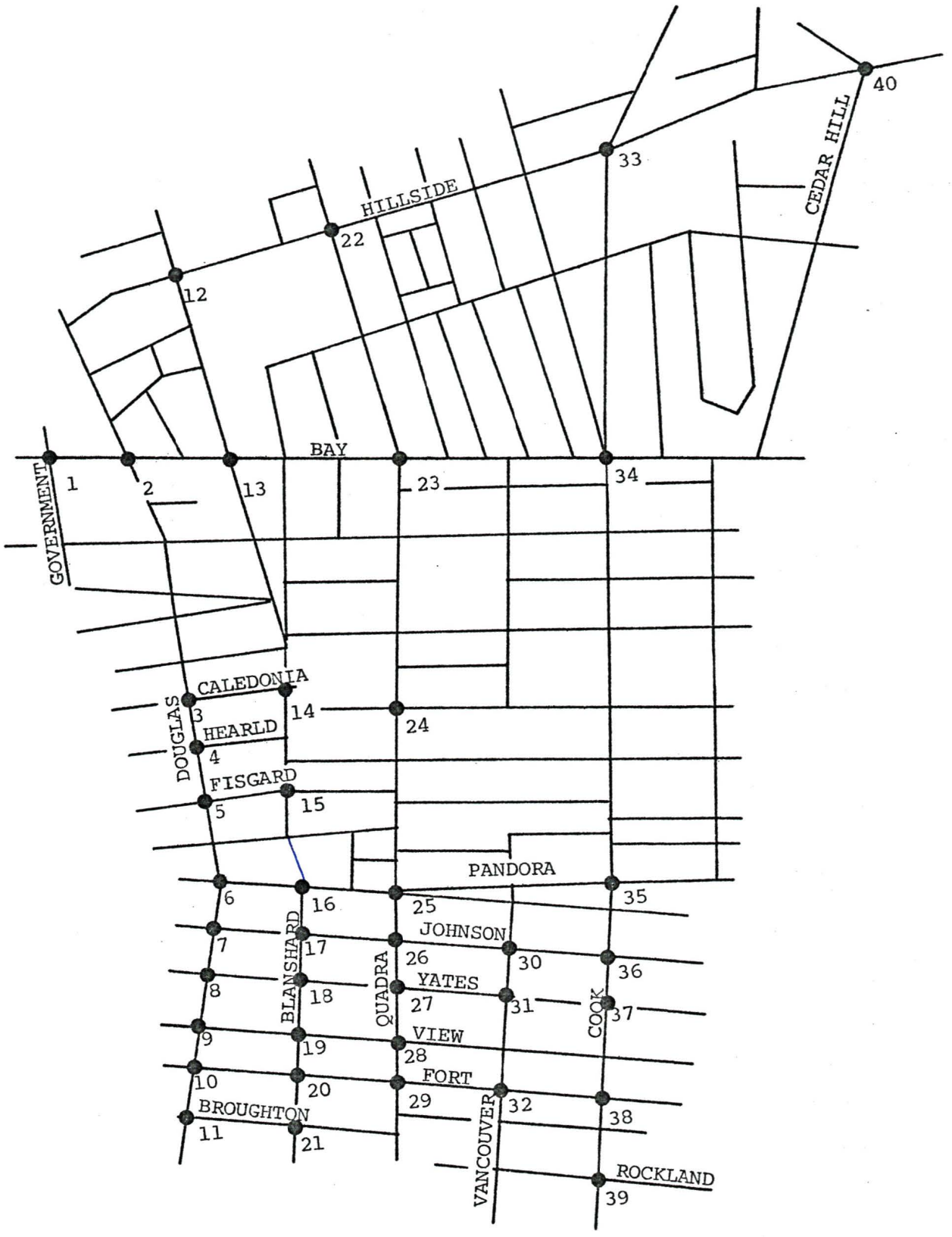


FIGURE 24. The test area.

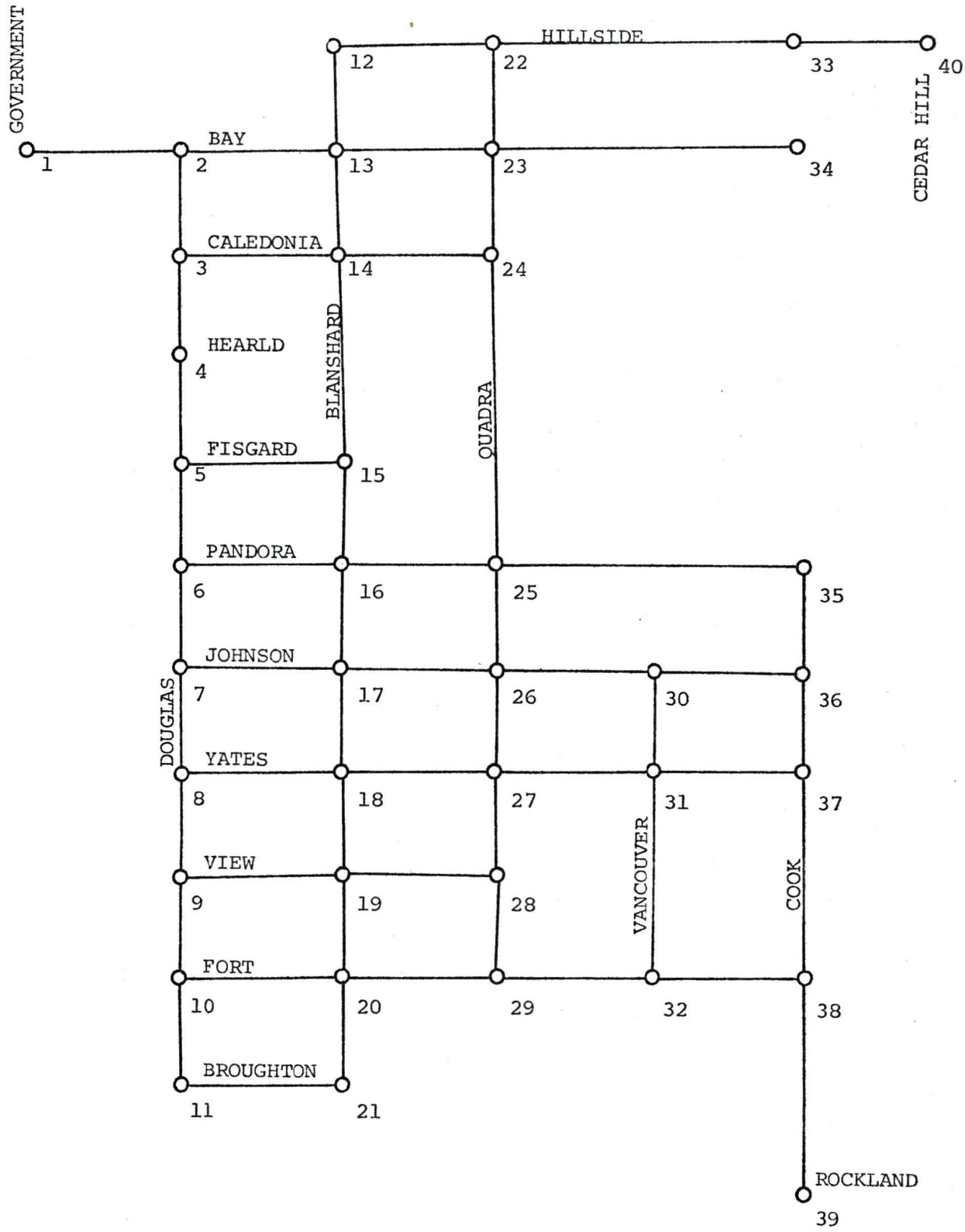


FIGURE 25. The graph associated with the test area.

5.1 The Cycle Split Problem as it Applies to the City of Victoria

In our application of the linear programming methods described in Section 2.3 to the Cycle Split Problem for the downtown area of Victoria, each intersection is assumed to have four incident legs numbered as in Figure 26.

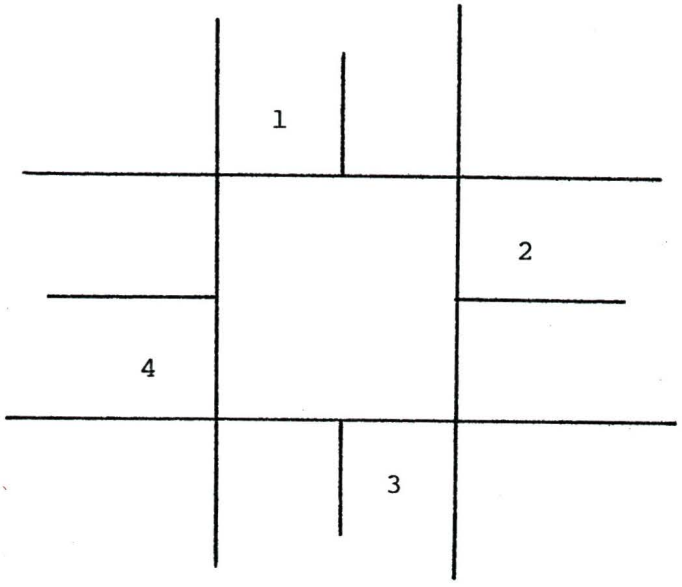


FIGURE 26. The numbering of the legs at an intersection.

The phases of all the signals in the test area fall into the 20 categories listed in Table III. Naturally no single signal goes through all 20 of these phases. For example, it would be illogical for an intersection to have an advanced green for northbound traffic followed immediately by an advance green for southbound traffic. Figure 27 illustrates the possible sequences of phases for a signal, while existing sequences are listed in Table IV.

	PHASE DESCRIPTION	COLOR OF LIGHTS			
		SOUTH-BOUND	WEST-BOUND	NORTH-BOUND	EAST-BOUND
1	Advance green to southbound	G	R	R	R
2	Advance green to northbound	R	R	G	R
3	Green to northbound and southbound	G	R	G	R
4	Amber to southbound (before delayed green to northbound traffic)	A	R	G	R
5	Delayed green to northbound (Also one way with no southbound traffic)	R	R	G	R
6	Amber to northbound (after delayed green to northbound traffic)	R	R	A	R
7	Amber to northbound (before delayed green to southbound traffic)	G	R	A	R
8	Delayed green to southbound (Also one way with no northbound traffic)	G	R	R	R
9	Amber to southbound (after delayed green to southbound traffic)	A	R	R	R
10	Amber to northbound and southbound	A	R	A	R
11	Advance green to eastbound	R	R	R	G
12	Advance green to westbound	R	G	R	R
13	Green to eastbound and westbound	R	G	R	G
14	Amber to eastbound (before delayed green to westbound traffic)	R	G	R	A
15	Delayed green to westbound (Also one way with no eastbound traffic)	R	G	R	R
16	Amber to westbound (after delayed green to westbound traffic)	R	A	R	R
17	Amber to westbound (before delayed green to eastbound traffic)	R	A	R	G
18	Delayed green to eastbound (Also one way with no westbound traffic)	R	R	R	G
19	Amber to eastbound after delayed green to eastbound traffic)	R	R	R	A
20	Amber to eastbound and westbound	R	A	R	A

TABLE III: The 20 possible phases.

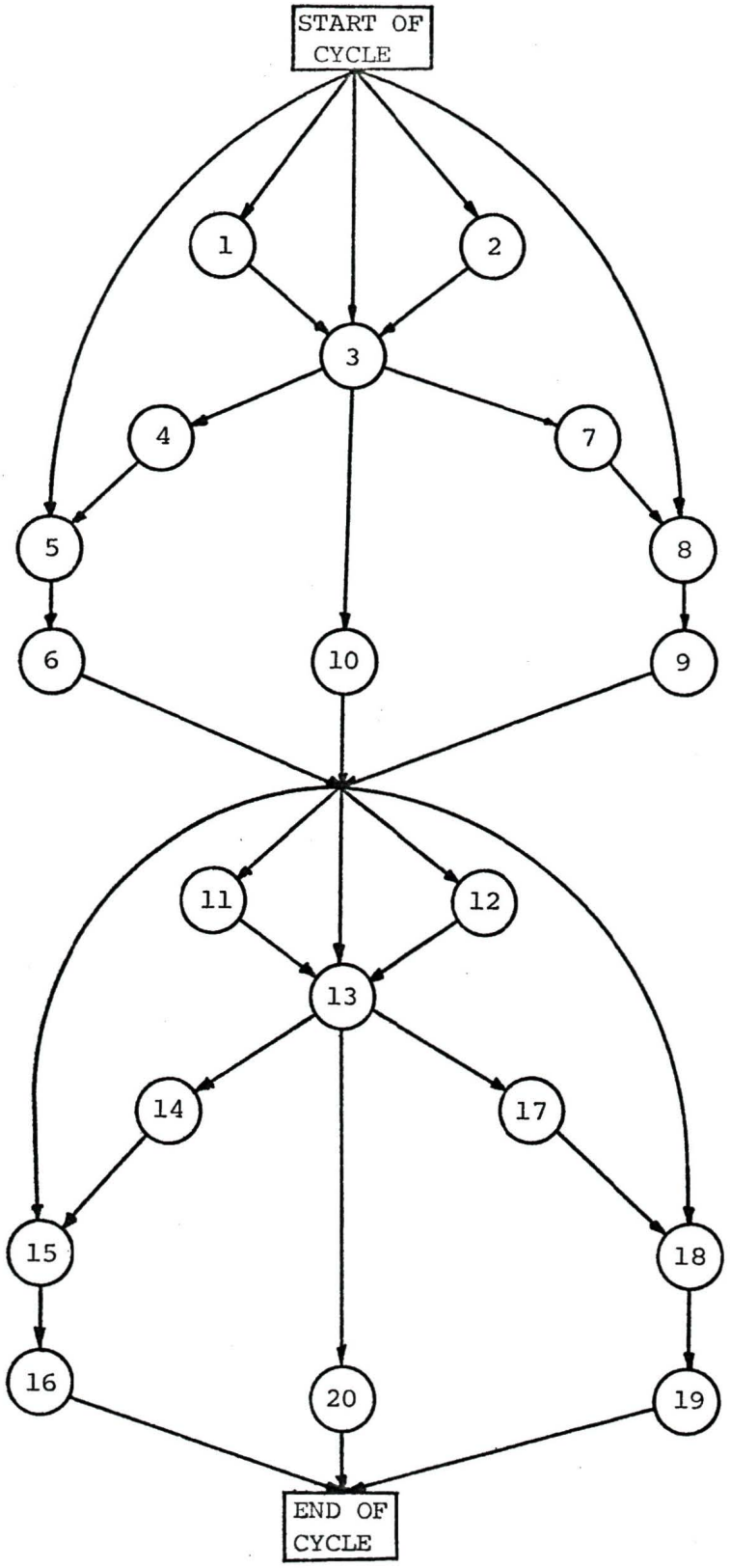


FIGURE 27. Possible sequences of phases.

INTERSECTION	PHASE SEQUENCE	INTERSECTION	PHASE SEQUENCE
1	2-3-10-13-20	21	3-10-13-20
2	3-10-13-20	22	3-10-13-20
3	3-10-13-20	23	3-10-13-20
4	3-7-8-9-13-20	24	3-10-13-20
5	3-10-13-20	25	3-10-15-16
6	3-10-15-16	26	3-10-18-19
7	3-10-18-19	27	3-10-13-20
8	3-10-15-16	28	3-10-13-20
9	3-10-13-20	29	3-10-13-20
10	1-3-10-13-20	30	3-10-18-19
11	3-10-13-20	31	3-10-13-20
12	1-3-10-12-13-20	32	3-10-13-20
13	3-10-13-20	33	3-10-11-13-20
14	3-10-13-20	34	3-10-13-20
15	2-3-10-23-20	35	3-10-15-16
16	3-10-15-16	36	3-10-18-19
17	3-7-8-9-18-19	37	3-10-13-20
18	3-10-15-16	38	3-10-13-20
19	3-10-13-20	39	3-10-13-20
20	3-7-8-9-18-19	40	3-10-13-20

Table IV: The existing sequences of phases.

Before applying the linear programming techniques discussed in Section 2.3 it is necessary to determine approximations to the constants c_{ijk}. That is, we must estimate the average number of vehicle starts from leg j (j = 1,2,3, or 4) of type i (i = 1 for left turns, 2 for right turns and 3 for straight through) during phase k (k = 1,2,3,..., or 20).

To calculate these constants we make the assumptions about traffic flow summarized in Table V. These are based on figures used by the traffic departments of both the City of Victoria and the City of Vancouver. Most are self-explanatory but it should be noted that a vehicle start rate of 1/20 per second is assumed for vehicles turning left or going straight while facing a red or amber light. This was an arbitrary constant designed to allow for the vehicles that usually pass through the intersection after the signal turns amber and before the flow of traffic abates. In Table V 'unopposed green' refers to a signal pattern in which traffic going the opposite direction is held back (an advance green for example). The term 'opposed green' refers to a pattern in which traffic does flow in the opposite direction (an ordinary green). Thus, during an opposed green fewer vehicles are able to make left turns than are able to do so during an unopposed green.

Based on the assumptions listed in Table V it is possible to approximate the constants c_{ijk} fairly easily. For example consider the leg of an intersection illustrated in Figure 28.

SITUATION	VEHICLE STARTS PER SECOND PER LANE (under heavy traffic conditions)
1) Vehicles turning left under:	
a) unopposed green	1/2
b) opposed green	1/10
c) red or amber	1/20
2) Vehicles turning right under:	
a) unopposed green	1/7
b) opposed green	1/7
c) red or amber	1/7
3) Vehicles going straight under:	
a) unopposed green	1/2
b) opposed green	1/2
c) red or amber	1/20

TABLE V. Summary of assumptions upon which the calculation of the constants c_{ijk} are based.

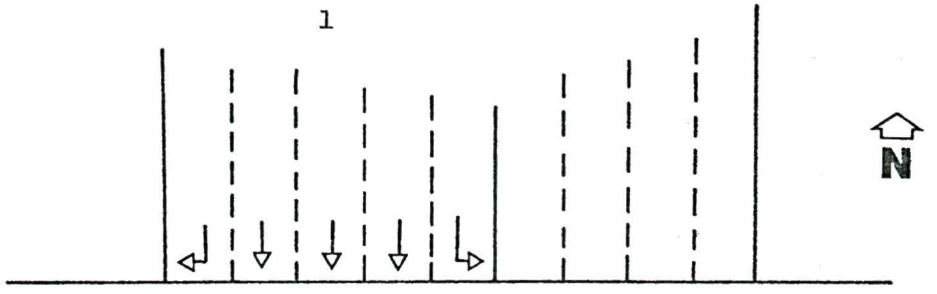


FIGURE 28. A simple leg.

To calculate the number of vehicles per second leaving leg 1 (j=1) by turning left (i=1) during an advance green for southbound traffic (k=1) we need only set c_{111} equal to 1/2 as there is an average of 1/2 vehicle per second per lane able to make a left turn during an unopposed green and there is only one left turn lane. If there had been two left turn only lanes then the constant would have the value $(1/2)\cdot 2$.

To calculate the number of vehicles per second going straight through the intersection (i=3) from leg 1 (j=1) during an advance green for southbound traffic (k=1) we need only set c_{311} equal to $(1/2)\cdot 3$ as there are three lanes reserved for traffic which does not turn and an average of 1/2 vehicle per second per lane is able to go through the intersection during an unopposed green.

A more complex situation arises when a lane is used for more than one type of vehicular movement. For example, consider the following leg of an intersection.

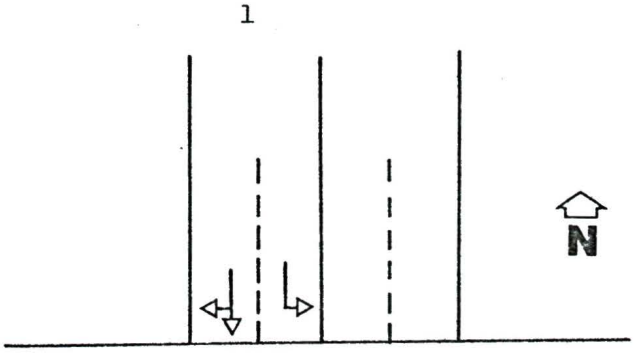


FIGURE 29. A more complex leg.

The calculation of c_{211} (that is the number of vehicles per second turning right (i=2) from leg 1 (j=1) during an advance green for southbound traffic (k=1)) is slightly more involved as we require additional information about the number of vehicles using the curb lane to turn right as compared to the total number using that lane.

For example, suppose 10 vehicles use the lane to turn right in a given period of time and 90 use it to proceed straight through the intersection. The total time required for the 100 vehicles to pass through the intersection is $7\times10 + 2\times90 = 250$ seconds (1/7 vehicle per second is able to turn right and 1/2 vehicle per second is able to go straight.) Thus, the rate at which the vehicles turn right is $c_{211} = 10/250 = 1/25$.

Using this approach to calculate all the required c_{ijk}'s for the Cycle Split Problem of a single intersection we were able to apply a standard implementation of the Simplex algorithm to the linear programming problem as stated in Section 2.3.

In a number of cases the present allotment to an amber phase is only 3.0 or 3.6 seconds. At the request of the City of Victoria this was brought up to a standard of 4.2 seconds. Hence in some cases the present allotment does not satisfy the inequalities of the problem as stated. Table VI provides a summary of the results of applying this technique to the intersections of the study area. The first column is the intersection number while the second and third columns give the estimated maximum flow before and after optimization. The fourth column indicates whether the constraints of the problem are presently being broken (by insufficient amber phases) while the last column shows the percentage improvement for those intersections where the present allotment does not break the constraints.

Intersection	Maximum Flow (vehicles per cycle) Presently	Optimized	Constraints Presently Violated?	% Improvement (if applicable)
1	186	189	No	1.66
2	135	139	No	2.41
3	154	144	Yes	
4	105	100	Yes	
5	154	155	No	0.36
6	175	171	Yes	
7	165	161	Yes	
8	163	168	No	2.66
9	122	118	Yes	
10	140	146	No	3.96
11	104	101	Yes	
12	230	239	No	3.83
13	204	209	No	2.28
14	199	187	Yes	
15	169	162	Yes	
16	234	231	Yes	
17	179	176	Yes	
18	157	162	No	3.20
19	118	122	No	3.11
20	148	153	No	3.77
21	92	93	No	1.23
22	139	142	No	1.74
23	97	101	No	4.24
24	80	83	No	4.34
25	166	167	No	0.62
26	134	142	No	5.98
27	177	185	No	4.71
28	64	72	No	11.80
29	93	100	No	7.04
30	91	109	No	20.31
31	116	126	No	9.33
32	95	105	No	10.98
33	162	160	Yes	
34	112	113	No	0.89
35	159	161	No	0.90
36	157	156	Yes	
37	135	137	No	1.37
38	114	116	No	1.53
39	83	95	No	13.59
40	143	139	Yes	

Table VI: Summary of results of applying linear programming techniques to the Cycle Split Problem.

5.2 The BOP for the City of Victoria

In the second stage of the project the phases determined in the first stage were used in the computer model described in Chapter 4 to simulate traffic along each leg of the test area. In general the average delay functions appeared similar to the one illustrated in Figure 30.

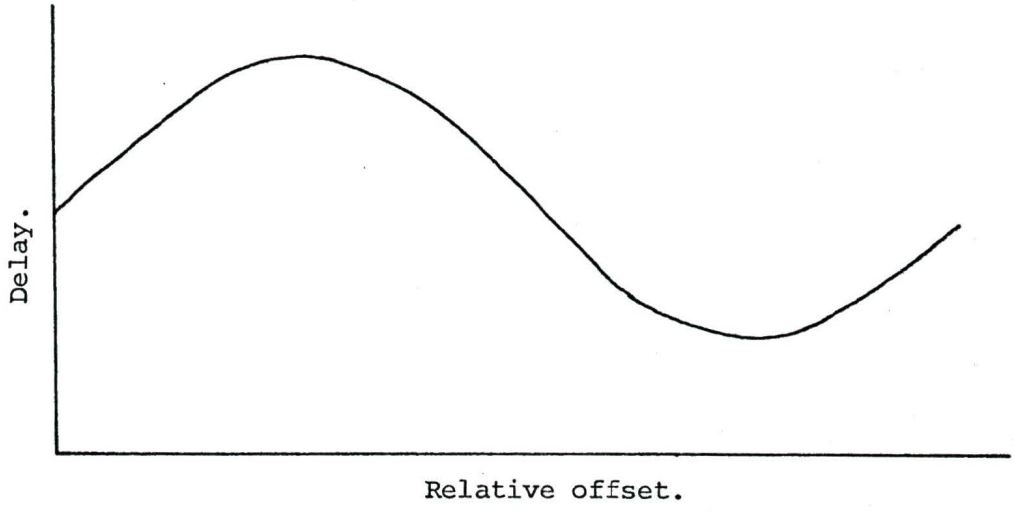


FIGURE 30. A typical average delay function.

Before applying the various minimization techniques outlined in Chapter 3 and in accord with the suggestion of Ross [1] each delay function was approximated by a sinusoidal curve of the form $g(x) = a \sin((b+x)\frac{2\pi}{60}) + c$. The graph of this function is given in Figure 31. Note that this function is periodic with period equal to the cycle length of the signal system for the City of Victoria (i.e. 60 seconds).

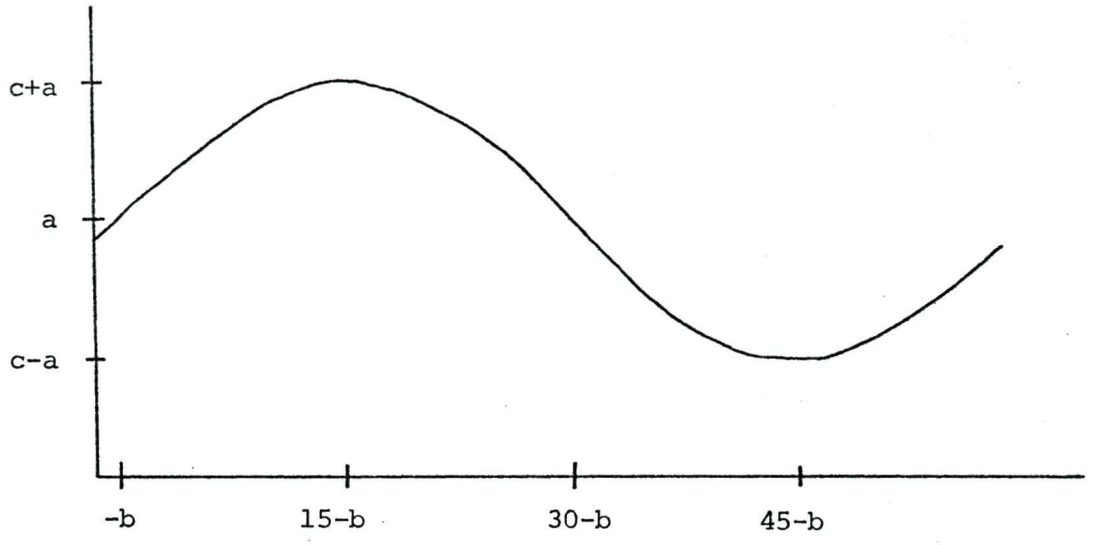


FIGURE 31. The graph of $g(x) = a \sin((b+x)\frac{2\pi}{60}) + c$

Results of this approximation are summarized in Table VII. The figures are given for the average delay function - to obtain the delay function for the leg it is necessary to multiply by the number of vehicles. The root mean square error is defined as:

$$r.m.s.e. = \sqrt{\frac{\sum_{i=1}^{N} (g(x_i) - f(x_i))^2}{N}}$$

where: $g(x)$ is the approximating function,

$f(x)$ is the approximated function, and

$x_1, x_2, \ldots, x_N$ are the data points used in the approximation.

v_i	v_j	number of vehicles traversing leg per hour (N_{ij})	a_{ij}/N_{ij}	b_{ij}	C_{ij}/N_{ij}	root mean square error/N_{ij}
1	2	1195	1.17	35.55	29.96	1.37
2	3	2085	1.44	16.96	55.56	1.11
2	13	1299	5.155	17.49	34.15	1.26
3	4	1980	5.19	43.24	21.07	1.05
3	14	291	3.43	13.35	24.88	0.68
4	5	1927	8.99	47.83	17.76	0.99
5	6	1645	5.01	47.02	21.25	0.77
5	15	517	4.14	14.26	25.81	1.68
6	7	1600	7.77	44.98	17.06	1.09
6	16	500	11.30	5.01	29.79	2.32
7	8	1588	7.08	38.68	19.54	1.34
7	17	550	10.54	24.89	31.27	1.63
8	9	1407	8.21	55.93	23.23	1.50
8	18	780	5.77	58.32	23.70	1.09
9	10	1472	7.54	45.68	17.26	0.77
9	19	837	7.61	12.37	38.11	1.44
10	11	1343	6.71	42.15	14.86	0.73
10	20	1001	11.47	22.37	29.87	1.51
11	21	531	4.56	18.34	30.36	1.62
12	13	1856	7.93	11.82	38.11	1.47
12	22	1565	8.36	4.01	36.92	1.38
13	14	2083	3.88	2.61	28.19	0.90
13	23	1252	6.97	7.08	37.21	1.93
14	15	2067	6.64	17.09	26.81	1.44
14	24	277	4.38	16.30	27.29	1.46
15	16	2059	1.65	32.89	39.61	1.36
16	17	1965	3.97	37.87	20.86	0.73
16	25	500	8.50	53.73	27.29	2.18
17	18	1609	7.97	41.11	20.11	1.50
17	26	700	6.39	20.51	22.97	1.70

v_i	v_j	number of vehicles traversing leg per hour (N_{ij})	a_{ij}/N_{ij}	b_{ij}	c_{ij}/N_{ij}	root mean square error/N_{ij}
18	19	1857	6.38	58.27	28.32	1.93
18	27	919	6.05	1.80	24.88	2.28
19	20	1362	5.23	31.57	30.98	1.87
19	28	497	6.18	16.87	35.74	2.09
20	21	1344	6.78	51.63	19.19	0.90
20	29	1326	5.72	27.52	24.74	0.90
22	33	1798	0.39	4.63	56.71	1.12
23	24	1269	4.48	26.76	40.44	2.10
23	34	1297	7.09	30.60	44.91	2.09
24	25	1410	10.00	11.11	41.67	1.37
25	26	1387	10.33	23.66	36.32	2.68
25	35	550	8.11	27.59	39.41	2.82
26	27	840	3.81	3.13	25.94	2.34
26	30	700	3.71	36.15	20.96	1.39
27	28	1009	7.23	0.66	23.71	1.55
27	31	793	6.36	0.56	23.62	1.58
28	29	1113	3.72	22.12	22.76	2.06
29	32	1552	5.27	32.35	23.88	1.04
30	31	377	5.00	55.48	21.58	1.45
30	36	750	2.33	31.80	17.39	1.76
31	32	382	8.93	5.04	33.62	1.97
31	37	792	4.26	53.18	17.49	1.97
32	38	1589	3.93	36.68	16.94	1.59
33	40	1476	3.74	11.64	54.25	0.67
35	36	1330	2.49	14.34	33.25	1.60
36	37	1403	12.09	6.56	44.55	2.01
37	38	1473	1.37	18.34	30.13	1.04
38	29	1533	3.67	47.11	18.72	0.83

TABLE VII: Summary of approximation of delay functions by sinusuidal curves.

Hence, the overall delay function was approximated by:

$$Z_G(\theta_1,\theta_2,\theta_3,\ldots,\theta_n) = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} f_{ij}(\theta_i-\theta_j)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} g_{ij}(\theta_i-\theta_j)$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} a_{ij}\ \sin((b_{ij}+\theta_i-\theta_j)\frac{2\pi}{60}) + c_{ij}$$

In the third stage the various methods of minimizing the overall delay function were evaluated. The first step was to establish lower and upper bounds on the function.

From Figure 31 it can be seen that the function $g(x) = a\ \sin((b+x)\frac{2\pi}{60}) + c$ takes on its minimum of $c - a$ when $x = 45 - b$. Therefore each function g_{ij} takes on its minimum of $c_{ij} - a_{ij}$ when $\theta_i-\theta_j = 45 - b_{ij}$.

Hence, a lower bound on the overall delay function can be determined by calculating:

$$L = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} g_{ij}(45-b_{ij})$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} a_{ij}\ \sin((b_{ij}+45-b_{ij})\frac{2\pi}{60}) + c_{ij}$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} a_{ij} \sin(\frac{3}{2}\pi) + c_{ij}$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} c_{ij} - a_{ij}$$

Similarly an upper bound for the overall delay function can be determined by calculating:

$$U = \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} g_{ij}(15-b_{ij})$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} a_{ij} \sin((b_{ij}+15-b_{ij})\frac{2\pi}{60}) + c_{ij}$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} a_{ij} \sin(\frac{\pi}{2}) + c_{ij}$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\in E(G)}} a_{ij} + c_{ij}$$

In the case of the City of Victoria, the lower bound was approximately 1.69 million seconds (about 469 hours) while the upper bound was about 2.48 million seconds (approximately 689 hours).

To provide a basis for evaluating the various possible methods of minimizing the overall delay function the mean and average of the function

were calculated. These calculations are summarized below:

i) mean:

$$\mu = E(Z_G(\theta_1,\theta_2,\ldots,\theta_{40}) \qquad \text{[the expected value of } Z_G\text{]}$$

$$= \int_0^{60}\int_0^{60}\cdots\int_0^{60} \frac{Z_G(\theta_1,\theta_2,\ldots,\theta_{40})d\theta_1 d\theta_2\ldots d\theta_{40}}{60^{40}}$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\varepsilon E(G)}} c_{ij}$$

$\approx$ 2.08 million seconds (578 hours)

ii) variance:

$$\sigma^2 = E((Z_G(\theta_1,\theta_2,\ldots,\theta_{40})-\mu)^2)$$

$$= \int_0^{60}\int_0^{60}\cdots\int_0^{60} \frac{(Z_G(\theta_1,\theta_2,\ldots,\theta_{40})-\mu)^2 d\theta_1 d\theta_2,\ldots,d\theta_{40}}{60^{40}}$$

$$= \sum_{\substack{i,j \\ [v_i,v_j]\varepsilon E(G)}} \frac{a_{ij}^2}{2}$$

$\approx$ 1,888 million seconds

[hence, the standard deviation is equal to about 0.043 million seconds]

As a further check, ten thousand sets of random offsets were generated and the overall delay function evaluated for each set. The distribution is shown in Figure 32.

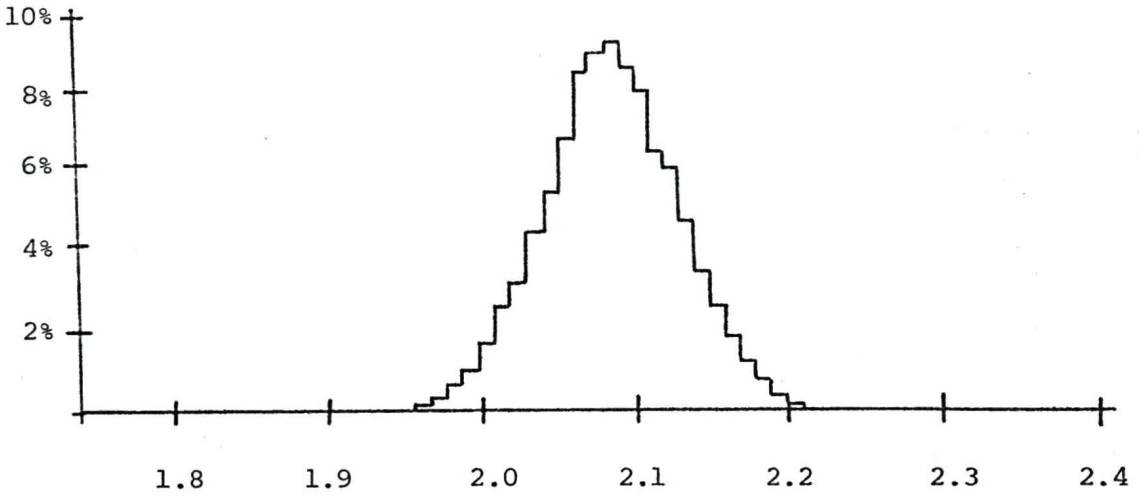


FIGURE 32. Distribution of values of the delay function (based on 10,000 sets of random offsets).

A FORTRAN implementation of the Davidon, Fletcher, Powell method of minimizing a function of several variables was applied to the problem to locate local minima [4]. Various initial points were utilized but all converged to a value of approximately 1.73 million seconds (481 hours). Convergence was within 10 iterations.

The complex method of function minimalization was also applied to the problem [11]. Numerous sets of starting offsets were tried along with various program parameters but all failed to converge. As suggested by C. Bird [2] this failure to converge was probably due to the large number of variables involved.

To check the effectiveness of selecting a subset of the edges of the graph associated with the street system which form a tree and contain all the vertices of the graph, a FORTRAN program was written to assign each of the edges a random weight and then to form a minimally weighted

spanning tree utilizing Prim's algorithm [10]. The offsets of the lights were then determined as described in Section 2 of Chapter 3. Since the delay functions for each of the legs were approximated by a function of the form

$$g_{ij}(\theta_i-\theta_j) = a_{ij} \sin((b_{ij}+(\theta_i-\theta_j))\frac{2\pi}{60}) + c_{ij}$$

the minimal delay occurs when

$b_{ij} + (\theta_i-\theta_j) = 45$. That is,

$$m_{ij} = a_{ij} \sin(b_{ij}+(\theta_i-\theta_j))\frac{2\pi}{60}) + c_{ij}$$

$$= a_{ij} \sin(45*\frac{2\pi}{60}) + c_{ij}$$

$$= a_{ij}(-1) + c_{ij}$$

$$= -a_{ij} + c_{ij}.$$

Utilizing the offsets obtained in this manner, the total delay function was evaluated. The process of selecting edges to form the tree and subsequent minimization was repeated numerous times to obtain results for a variety of trees.

A tree generating an overall delay of 1.80 million seconds (500 hours) was found within the first minute of CPU time (30 trees examined). After 5 minutes of CPU time (150 trees examined) one generating an overall delay of 1.78 seconds was found. By extending the time limit to 60 minutes (1750 trees generated) a tree yielding an overall delay of 1.77 million seconds (492 hours) was detected. This tree is illustrated in Figure 33.

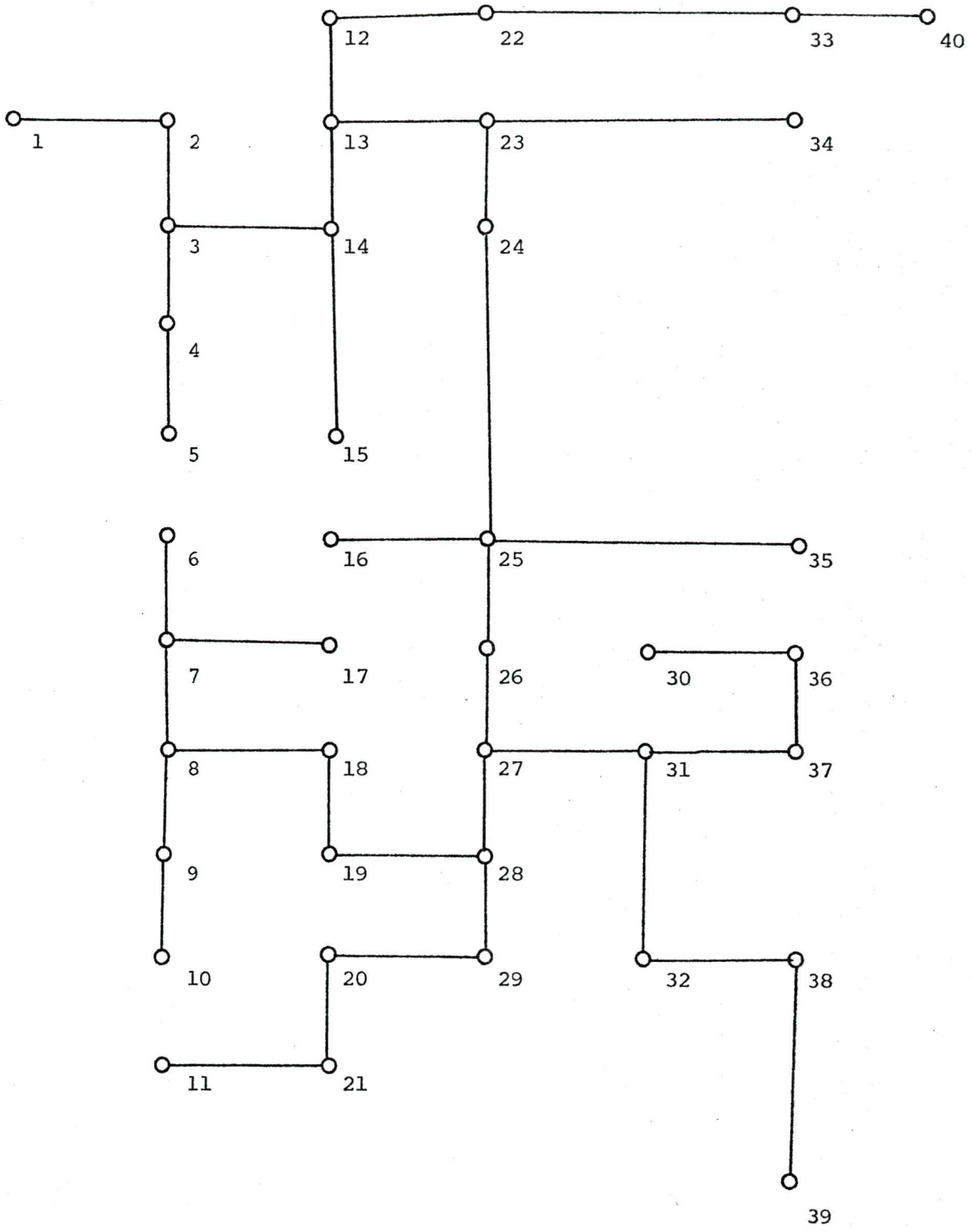


FIGURE 33. Tree yielding an overall delay of 1.77 million seconds.

The present method used by the City of Victoria for determining the offsets of the signals for the system involves selecting a tree (based on the amount of traffic utilizing the associated legs) and then setting the offsets so that a vehicle travelling at a fixed speed (22.5 m.p.h. in the case of Victoria) would have perfect progression through the lights [3]. Figure 34 illustrates the tree presently being used in the City of Victoria. Utilizing this method of setting the offsets resulted in an overall delay of 2.0 million seconds (561 hours). Utilization of the same tree but setting the offsets as described in Section 2 of Chapter 3 resulted in an overall delay of 1.80 million seconds (500 hours). This represents an improvement of about 11%.

The results of the various methods of minimizing the overall delay function are summarized in Figure 35. As can be seen, the Davidon, Fletcher, Powell method resulted in a value quite near the lower limit. However, selecting a tree in the traditional manner and then optimizing over it as in Section 3.2, resulted in an overall delay that is only slightly larger. This method has the advantage that it allows the planner to specify those legs over which the delay is minimal.

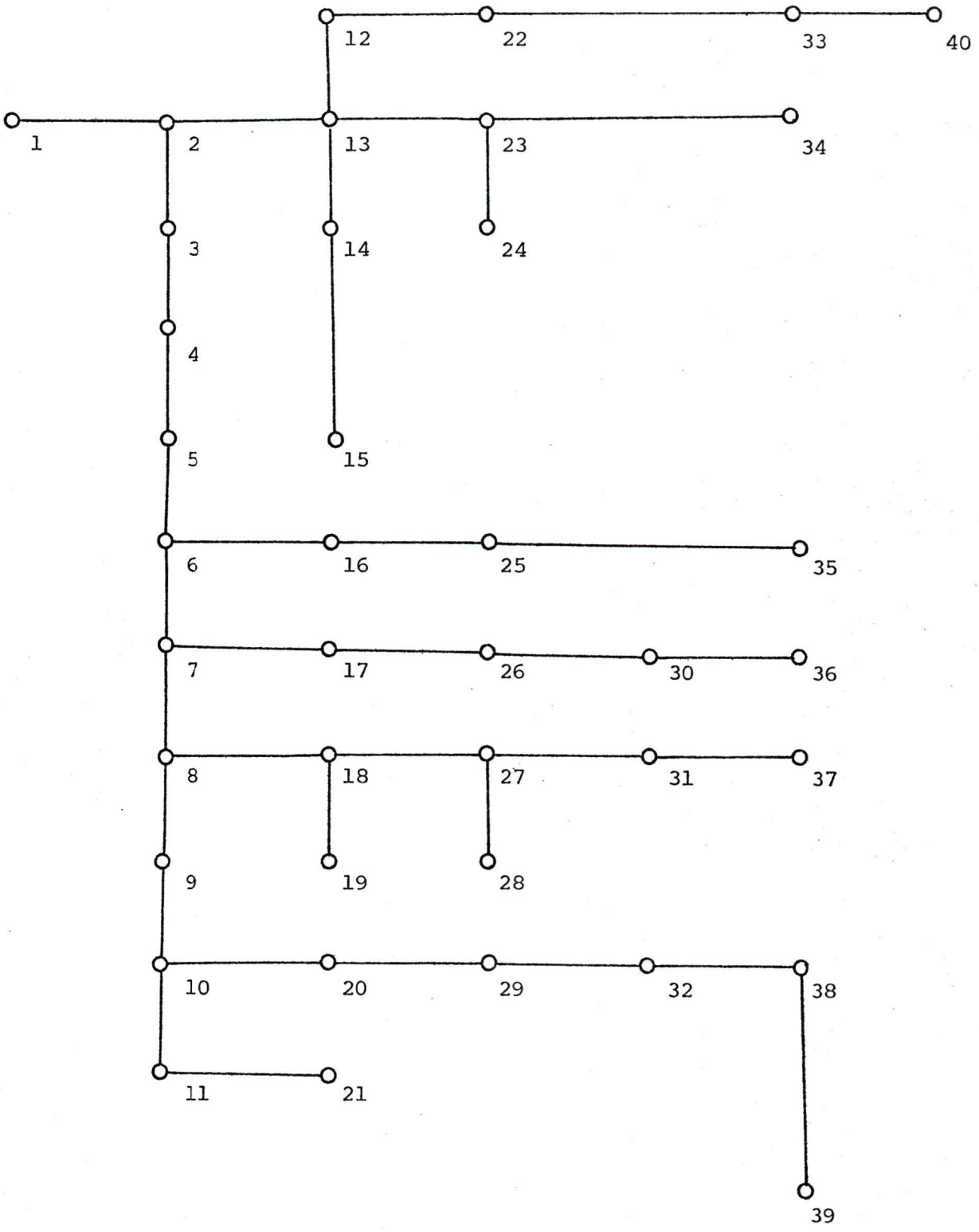


FIGURE 34. Tree presently used by City of Victoria.

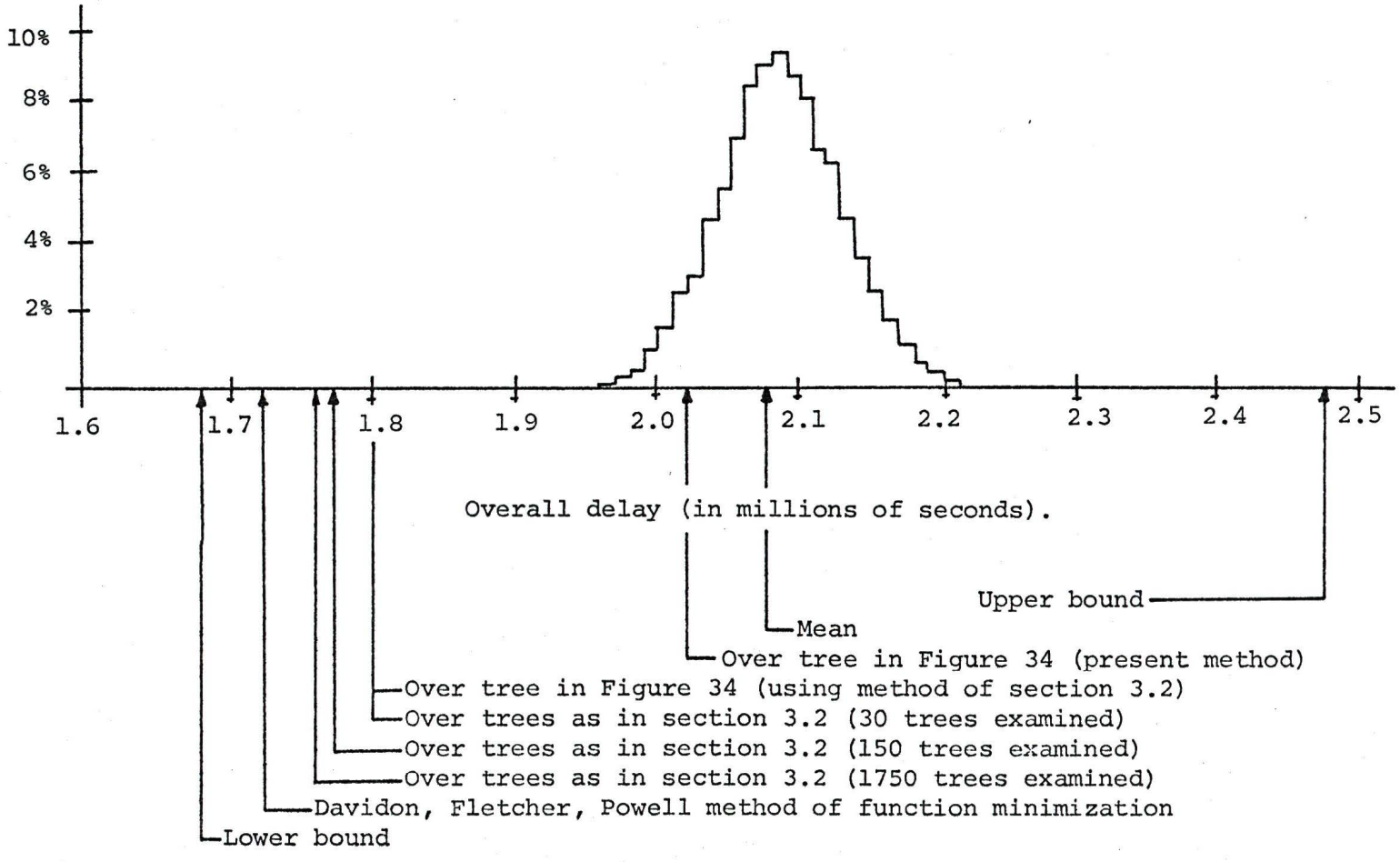


Figure 35. Summary of methods of minimizing the overall delay function.

CHAPTER 6

CONCLUSIONS

In the last few years, traffic engineers have shown considerable interest in the minimization of delays at signalized intersections. In almost every urban area this type of delay is the major contributor to traffic delay. This study has emphasized the importance of determining the correct allocation of a cycle to the phases of a signal as well as the optimal synchronization of the signals.

As seen in Chapter 2, the problem of determining how best to split the cycle length among the phases of a light can be formulated as a linear programming problem if we assume that a queue of vehicles is waiting at the intersection (in effect assuming the intersection is working at capacity) and also that there exist constants which describe the possible number of vehicle starts of a particular type from a given leg during a phase of the light. Applying this technique to data for downtown Victoria resulted in projected improvements in flow of between 0.4% and 20.3%. It is noteworthy that the improvements for major intersections were typically less than those for relatively minor intersections. This may be due to the fact that the major intersections have been adjusted more than the minor ones and so were closer to the optimal setting.

The first step in examining how best to synchronize the signals to minimize traffic delay was to create a model to simulate traffic along a given leg. The model developed portrayed traffic conditions fairly accurately as was shown in Section 4.2. By fixing the offset of one light at zero while stepping the other through an offset from zero to the cycle length of the system, it was possible to examine the delay

function for that leg in some detail. In Section 2 of Chapter 5 it was seen that these delay functions could be approximated by sinusoidal curves with period equal to the cycle length.

In Chapter 3 we examined some reductions that could be made to the problem of synchronizing the signals. An algorithm for completely solving the problem was formally stated for the case when the associated graph had components which were either trees or isolated vertices. The last section of Chapter 3 examined possible methods of solving the synchronization problem when no reductions could be made. These methods were then applied to the data for downtown Victoria to evaluate their effectiveness.

The present method of synchronizing the signals for the City of Victoria involves selecting a tree containing all the intersections and then setting the lights so that a vehicle travelling at a fixed rate would have perfect progression over the legs associated with the edges of the tree. This yielded an overall delay of approximately 2.02 million seconds (561 hours). By using the same tree but minimizing as described in Section 4 of Chapter 3, this was reduced to about 1.80 million seconds (500 hours). It is clear that the fixed rate method of synchronizing can be significantly improved upon through the use of simulation to determine the exact nature of the delay functions.

To check if better results could be obtained by a different choice of trees, a number of different trees were selected by application of Prim's algorithm to weights randomly assigned to the edges of the graph. Within the first 60 seconds of CPU time a tree which generated an overall delay of 1.80 million seconds (500 hours) was found. Within 5 minutes, one generating an overall delay of 1.78 million seconds (494 hours) was

found. Increasing the time to 60 minutes of CPU resulted in a slight decrease to about 1.77 million seconds (492 hours). This shows that the present method of selecting a tree over which to minimize delays is probably satisfactory.

The Davidon, Fletcher, Powell method of minimizing a function of a number of variables was also successfully applied to the problem. This resulted in finding a set of offsets which yielded an overall delay of 1.73 million seconds or about 480 hours (only 0.04 million seconds greater than if it were possible to minimize delay over the entire graph). This method, however, has the distinct disadvantage that the planner can not specify those legs over which optimization is to occur.

One problem encountered in this study was the amount of computing time required to simulate traffic along a given leg in order to determine the characteristics of the associated delay function. An avenue of future research would be to develop better analytical methods for determining this function. The use of factor analysis and linear regression techniques upon the results of the present study might prove fruitful. Further, if one is only interested optimizing over a given tree rather than minimizing over the entire graph, one need only determine the relative offset for a leg which will yield the minimal delay over that leg rather than the characteristics of the entire delay function. This approach could lead to significant reductions in the amount of computer time required.

REFERENCES

1. Allsop, R.E. "Selection of Offsets to Minimize Delay to Traffic in a Network Controlled by Fixed-Time Signals." Transportation Science, Vol. 2 (1968), pp. 1-13.

2. Bird, C., C.M. Lee, Dennis O'Reilly. Non-linear Function Optimization. Computing Centre, The University of British Columbia, Vancouver, B.C., (October 1973).

3. Box, M.J. "A New Method of Constrained Optimization and a Comparison with other Methods." Computer Journal, Vol. 8 (1965), pp. 42-52.

4. Chan, Chorkin. Minfun: Minimization of Unconstraint Multivariable Function. Academic Systems, The University of Victoria, Victoria, B.C., (February 1973).

5. Clapham, J.C., C. Wild and Thomas A. Lambe. "Computer Calculation of Best Traffic Signal Settings in Vancouver's Downtown Network." (unpublished)

6. Davidon, W.C. "Variable Metric Method for Minimization." A.E.C. Research and Development Report, ANL-5990, (1959).

7. Fletcher, R. and M.J.D. Powell. "A Rapidly Convergent Descent Method for Minimization." Computer Journal, Vol. 6, pp. 163-168.

8. Harary, F. Graph Theory. Addison-Wesley Pub. Co., Reading, Mass., (1969).

9. Hillier, J.A. and R. Rothery. "The Synchronization of Traffic Signals for Minimum Delay." Transportation Science, Vol. 1 (1967), pp. 81-94.

10. Hillier, Frederick S., and Gerald J. Lieberman. Introduction to Operations Research. Holden-Day, Inc., (1969). pp. 222-225.

11. Richardson, Joel A., and J.L. Krester. "Algorithm 454: The Complex Method for Constrained Optimization." Communications of the ACM, Vol. 16 (1973), pp. 487-489.

12. Robertson, D.I. "TRANSYT: A Traffic Network Study Tool." Road Research Laboratory, Report LR253, (1969).

13. Ross, D.W. "Traffic Control and Highway Networks." Networks, Vol. 2, (1972), pp. 97-123.

14. SIGOP, Traffic Research Corporation, New York, (1966).

15. Wilde, D. (private communication).

APPENDIX A

DOCUMENTATION OF SUBROUTINES USED IN PHASE I

```
      SUBROUTINE SPLIT                                                      A    1
C                                                                           A    2
C                                                                           A    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR                 A    4
C     VICTORIA, BRITISH COLUMBIA                                            A    5
C     CANADA                                                                A    6
C                                                        NOV. 1974          A    7
C                                                                           A    8
C                                                                           A    9
C     PURPOSE:      TO OPTIMALLY ALLOCATE THE CYCLE LENGTH OF               A   10
C                   AN INTERSECTION TO ITS PHASES.                          A   11
C                                                                           A   12
C                                                                           A   13
C     METHOD:                                                               A   14
C                                                                           A   15
C                   THE METHOD INVOLVES SOLVING THE DUAL OF THE             A   16
C                   LINEAR PROGRAMMING PROBLEM FORMALLY STATED              A   17
C                   IN CHAPTER 2 AND DISCUSSED IN DETAIL IN                 A   18
C                   CHAPTER 5 OF THE COVERING WORK.                         A   19
C                                                                           A   20
C                                                                           A   21
C                   APPROXIMATIONS ARE MADE TO THE CONSTANTS                A   22
C                   C(I,J,K) AS OUTLINED IN CHAPTER 5.  BASED               A   23
C                   ON THESE CONSTANTS AND THE D(I,J)'S                     A   24
C                   AND A(K)'S PROVIDED, AN INITIAL SIMPLEX                 A   25
C                   TABLEAU IS GENERATED.  THIS IS SOLVED                   A   26
C                   USING AN IMPLEMENTATION OF THE SIMPLEX                  A   27
C                   ALGORITHM.  IF THE PROBLEM IS INFEASIBLE                A   28
C                   AS STATED, THE RESTRICTIONS ON THE                      A   29
C                   AMOUNT OF TRAFFIC FLOW ARE REDUCED AND                  A   30
C                   THE PROBLEM RE-FORMULATED.  THE CYCLE LENGTH            A   31
C                   IS MAINTAINED AT 60 SECONDS FOR THIS STUDY              A   32
C                                                                           A   33
C                                                                           A   34
C     VARIABLES (INCLUDING THOSE IN CALLED ROUTINES):                       A   35
C                                                                           A   36
C                                                                           A   37
C     A(I,J)        (I=1,2,...,15); (J=1,2,...,35)                        A   38
C                   SIMPLEX TABLEAU.                                        A   39
C                                                                           A   40
C     ALOT(K)       (K=1,2,...,20)                                        A   41
C                   OPTIMAL ALLOTMENT TO PHASE K.                           A   42
C                                                                           A   43
C     AMIN(K)       (K=1,2,...,20)                                        A   44
C                   MINIMUM ALLOTMENT TO PHASE K.                           A   45
C                                                                           A   46
C     C(I,J,K)      (I=1,2,3); (J=1,2,3,4); (K=1,2,...,20)                A   47
C                   NUMBER OF VEHICLE STARTS PER SECOND OF TYPE             A   48
C                   I FROM LEG J DURING PHASE K.                            A   49
C                                                                           A   50
```

```
C      D(I,J)        (I=1,2,3); (J=1,2,3,4)                                  A  51
C                    NUMBER OF VEHICLE STARTS PER HOUR OF TYPE I             A  52
C                    FROM LEG J.                                             A  53
C                                                                            A  54
C      E(K)          (K=1,2,...,20)                                          A  55
C                    NUMBER OF VEHICLE STARTS POSSIBLE DURING A              A  56
C                    SECOND OF PHASE K.                                      A  57
C                                                                            A  58
C      EBTA          TIME IN CYCLE LIGHT FOR EASTBOUND TRAFFIC TURNS         A  59
C                    AMBER.                                                  A  60
C                                                                            A  61
C      EBTG          TIME IN CYCLE LIGHT FOR EASTBOUND TRAFFIC TURNS         A  62
C                    GREEN.                                                  A  63
C                                                                            A  64
C      EBTR          TIME IN CYCLE LIGHT FOR EASTBOUND TRAFFIC TURNS RED.    A  65
C                                                                            A  66
C      FLOW          FACTOR BY WHICH ORIGINAL TRAFFIC CONSTRAINTS            A  67
C                    ARE MODIFIED TO PRODUCE A FEASIBLE PROBLEM.             A  68
C                                                                            A  69
C      I             DO-LOOP PARAMETER.                                      A  70
C                                                                            A  71
C      ICOLJ         CONDITION OF LEG J DURING PHASE K.                      A  72
C                                                                            A  73
C      ICON(K,J)     (K=1,2,...,20); (J=1,2,3,4)                             A  74
C                    CONDITION THAT LEG J IS IN DURING PHASE K.              A  75
C                    (1: UNOPPOSED GREEN; 2: OPPOSED GREEN;                  A  76
C                    3: RED OR AMBER)                                        A  77
C                                                                            A  78
C      ID            COMPLETION CODE OF THE SIMPLEX ROUTINE.                 A  79
C                    (1: OPTIMAL SOLUTION FOUND; 2: ORIGINAL                 A  80
C                    PROBLEM INFEASIBLE, 3: ORIGINAL PROBLEM UNBOUNDED)      A  81
C                                                                            A  82
C      IDENT(I)      (I=1,2,...,6)                                           A  83
C                    ALPHANUMERIC IDENTIFICATION OF INTERSECTION.            A  84
C                                                                            A  85
C      IDIR(J)       (J=1,2,3,4)                                             A  86
C                    INTERSECTION NUMBER OF THE NEXT INTERSECTION IN THE     A  87
C                    J DIRECTION.                                            A  88
C                                                                            A  89
C      IJK           TEMPORARY STORAGE LOCATION.                             A  90
C                                                                            A  91
C      IN            INTERSECTION NUMBER.                                    A  92
C                                                                            A  93
C      INT           INTERSECTION NUMBER.                                    A  94
C                                                                            A  95
C      IP            TEMPORARY LOCATION USED IN SELECTING CORRECT            A  96
C                    C(I,J,K).                                               A  97
C                                                                            A  98
C      ISUB(K)       (K=1,2,...,20)                                          A  99
C                    PHASE NUMBER OF THE KTH PHASE OF THE SIGNAL.            A 100
```

```
C                                                                       A 101
C     ITEMP(I)      (I=1,2,...,15)                                      A 102
C                   TEMPORARY STORAGE.                                  A 103
C                                                                       A 104
C     ITYP          TYPE OF LANE I.                                     A 105
C                                                                       A 106
C     ITYPE(I)      (I=1,2,...,10)                                      A 107
C                   TYPE OF LANE I.                                     A 108
C                      1: LEFT TURN ONLY.                               A 109
C                      2: RIGHT TURN ONLY.                              A 110
C                      3: STRAIGHT THROUGH TRAFFIC ONLY.                A 111
C                      4: LEFT & STRAIGHT TRAFFIC.                      A 112
C                      5: RIGHT & STRAIGHT TRAFFIC.                     A 113
C                      6: ALL TYPES OF TRAFFIC ALLOWED.                 A 114
C                                                                       A 115
C     J             TYPE OF MOVEMENT.                                   A 116
C                                                                       A 117
C     JP            TEMPORARY LOCATION USED IN SELECTING CORRECT        A 118
C                   C(I,J,K).                                           A 119
C                                                                       A 120
C                                                                       A 121
C     J1            DO-LOOP PARAMETER.                                  A 122
C                                                                       A 123
C     J2            DO-LOOP PARAMETER.                                  A 124
C                                                                       A 125
C     K             DO-LOOP PARAMETER.                                  A 126
C                                                                       A 127
C     KP            TEMPORARY LOCATION USED IN SELECTING                A 128
C                   CORRECT C(I,J,K).                                   A 129
C                                                                       A 130
C     K1            TEMPORARY STORAGE.                                  A 131
C                                                                       A 132
C     K2            DO-LOOP PARAMETER.                                  A 133
C                                                                       A 134
C     L             ROW OF VARIABLE TO ENTER BASIS.                     A 135
C                                                                       A 136
C     LABEL(I)      (I=1,2,...,15)                                      A 137
C                   LABELS OF THE INITIAL BASIS VECTORS.                A 138
C                                                                       A 139
C     LN            LEG NUMBER.                                         A 140
C                                                                       A 141
C     M             NUMBER OF EQUATIONS IN SIMPLEX TABLEAU.             A 142
C                                                                       A 143
C     MP1           M + 1                                               A 144
C                                                                       A 145
C     MP2           M + 2                                               A 146
C                                                                       A 147
C     MP3           M + 3                                               A 148
C                                                                       A 149
C     MP4           M + 4                                               A 150
```

```
C
C       N               NUMBER OF VARIABLES IN SIMPLEX TABLEAU.
C
C       NAVIB           NUMBER OF ARTIFICIAL VARIABLES IN BASIS.
C
C       NBTA            TIME IN CYCLE LIGHT FOR NORTHBOUND TRAFFIC TURNS
C                       AMBER.
C
C       NBTG            TIME IN CYCLE LIGHT FOR NORTHBOUND TURNS GREEN.
C
C       NBTR            TIME IN CYCLE LIGHT FOR NORTHBOUND TURNS RED.
C
C       NC              NUMBER OF COLUMNS IN SIMPLEX TABLEAU.
C
C       NLL             NUMBER OF LEFT TURN ONLY LANES.
C
C       NLS             NUMBER OF STRAIGHT THROUGH LANES IN LEG.
C
C       NLT             TOTAL NUMBER OF LANES IN LEG.
C
C       NLTM1           NLT - 1
C
C       NOW(K)          (K=1,2,...,20)
C                       PRESENT ALLOTMENT TO PHASE K.
C
C       NP              NUMBER OF PHASES IN A CYCLE FOR THE INTERSECTION.
C
C       NPP2            NP + 2
C
C       NPP14           NP + 14
C
C       NP1             N + 1
C
C       NP2             N + 2
C
C       NP3             N + 3
C
C       NP4             N + 4
C
C       NR              NUMBER OF ROWS IN SIMPLEX TABLEAU.
C
C       NRL             NUMBER OF RIGHT TURN ONLY LANES.
C
C       NSIC            NUMBER OF SECONDS IN A CYCLE.
C
C       OPP(I,J)        (I=1,2,...,10); (J=1,2,3)
C                       NUMBER OF VEHICLE STARTS PER SECOND OF TYPE J FROM
C                       LANE I OF THE PRESENT LEG DURING AN OPPOSED GREEN
C                       PHASE.
C
```

A 151 A 152 A 153 A 154 A 155 A 156 A 157 A 158 A 159 A 160 A 161 A 162 A 163 A 164 A 165 A 166 A 167 A 168 A 169 A 170 A 171 A 172 A 173 A 174 A 175 A 176 A 177 A 178 A 179 A 180 A 181 A 182 A 183 A 184 A 185 A 186 A 187 A 188 A 189 A 190 A 191 A 192 A 193 A 194 A 195 A 196 A 197 A 198 A 199 A 200

```
C     PIVOT          VALUE OF THE PIVOT ELEMENT IN THE GAUSS-JORDAN          A 201
C                    ELIMINATION.                                            A 202
C                                                                            A 203
C     RED(I,J)       (I=1,2,...,10); (J=1,2,3)                               A 204
C                    NUMBER OF VEHICLE STARTS OF TYPE J FROM LANE I          A 205
C                    OF THE PRESENT LEG DURING A RED OR AMBER PHASE.         A 206
C                                                                            A 207
C     SBTA           TIME IN CYCLE LIGHT FOR SOUTHBOUND TRAFFIC              A 208
C                    TURNS AMBER.                                            A 209
C                                                                            A 210
C     SBTG           TIME IN CYCLE LIGHT FOR SOUTHBOUND TURNS GREEN.         A 211
C                                                                            A 212
C     SBTR           TIME IN CYCLE LIGHT FOR SOUTHBOUND TURNS RED.           A 213
C                                                                            A 214
C     SUM            TEMPORARY STORAGE FOR CALCULATING VARIOUS SUMS.         A 215
C                                                                            A 216
C     SUMN           TOTAL TIME PRESENTLY ALLOTTED.                          A 217
C                                                                            A 218
C     TEM            TEMPORARY STORAGE.                                      A 219
C                                                                            A 220
C     TF(J,I,L)      (J=1,2,3,4); (I=1,2,3); (L=1,2,3)                       A 221
C                    NUMBER OF VEHICLE STARTS OF TYPE I FROM LEG J           A 222
C                    WHEN THAT LEG IS IN A PHASE OF TYPE L.                  A 223
C                                                                            A 224
C     TOLER          TOLERANCE FOR SIMPLEX ALGORITHM.                        A 225
C                                                                            A 226
C     TOPP(J)        (J=1,2,3)                                               A 227
C                    NUMBER OF VEHICLE STARTS PER SECOND OF TYPE J           A 228
C                    FROM THE PRESENT LEG DURING AN OPPOSED GREEN            A 229
C                    PHASE.                                                  A 230
C                                                                            A 231
C     TRED(J)        (J=1,2,3)                                               A 232
C                    NUMBER OF VEHICLE STARTS PER SECOND OF TYPE J           A 233
C                    FROM THE PRESENT LEG DURING A RED OR AMBER              A 234
C                    PHASE.                                                  A 235
C                                                                            A 236
C     TUNOPP(J)      (J=1,2,3)                                               A 237
C                    NUMBER OF VEHICLE STARTS PER SECOND OF TYPE J           A 238
C                    FROM THE PRESENT LEG DURING AN UNOPPOSED GREEN          A 239
C                    PHASE.                                                  A 240
C                                                                            A 241
C     UNOPP(I,J)     (I=1,2,...,10); (J=1,2,3)                               A 242
C                    NUMBER OF VEHICLE STARTS OF TYPE J FROM LANE I          A 243
C                    OF THE PRESENT LEG DURING AN UNOPPOSED GREEN            A 244
C                    PHASE.                                                  A 245
C                                                                            A 246
C     VEL(I)         (I=1,2,...,10)                                          A 247
C                    NUMBER OF VEHICLES ENTERING LANE I PER HOUR.            A 248
C                                                                            A 249
C     VGSFL(I)       (I=1,2,...,10)                                          A 250
```

```
C                NUMBER OF VEHICLES PER HOUR GOING STRAIGHT                A 251
C                FROM LANE I.                                              A 252
C                                                                          A 253
C     VGSL       VEHICLES GOING STRAIGHT FROM LEG.                         A 254
C                                                                          A 255
C     VL         VEHICLES GOING LEFT.                                      A 256
C                                                                          A 257
C     VR         VEHICLES GOING RIGHT.                                     A 258
C                                                                          A 259
C     VS         VEHICLES GOING STRAIGHT.                                  A 260
C                                                                          A 261
C     VT         TOTAL VEHICLES IN LEG.                                    A 262
C                                                                          A 263
C     VTLFL(I)   (I=1,2,...,10)                                            A 264
C                NUMBER OF VEHICLES PER HOUR TURNING LEFT                  A 265
C                FROM LANE I.                                              A 266
C                                                                          A 267
C     VTLL       VEHICLES TURNING LEFT FROM LEG.                           A 268
C                                                                          A 269
C     VTRFL(I)   (I=1,2,...,10)                                            A 270
C                NUMBER OF VEHICLES PER HOUR TURNING RIGHT.                A 271
C                                                                          A 272
C     VTRL       VEHICLES TURNING RIGHT FROM LEG.                          A 273
C                                                                          A 274
C     W          TEMPORARY STORAGE USED IN CALCULATION OF NUMBER           A 275
C                OF VEHICLES ENTERING EACH LANE.                           A 276
C                                                                          A 277
C     WBTA       TIME IN CYCLE LIGHT FOR WESTBOUND TRAFFIC TURNS           A 278
C                AMBER.                                                    A 279
C                                                                          A 280
C     WBTG       TIME IN CYCLE LIGHT FOR WESTBOUND TRAFFIC TURNS GREEN     A 281
C                                                                          A 282
C     WBTR       TIME IN CYCLE LIGHT FOR WESTBOUND TURNS RED.              A 283
C                                                                          A 284
C     WG         WHEN EAST/WEST LIGHTS TURN GREEN.                         A 285
C                                                                          A 286
C     WPS(K)     (K=1,2,...,20)                                            A 287
C                WHEN IN CYCLE PHASE K STARTS.                             A 288
C                                                                          A 289
C     Z1         TOTAL NUMBER OF VEHICLE STARTS PER CYCLE WITH             A 290
C                PRESENT ALLOTMENT.                                        A 291
C                                                                          A 292
C     Z2         TOTAL NUMBER OF VEHICLE STARTS PER CYCLE WITH             A 293
C                OPTIMAL ALLOTMENT.                                        A 294
C                                                                          A 295
C     Z3         PERCENTAGE INCREASE OF VEHICLE STARTS PER CYCLE           A 296
C                AFTER OPTIMIZATION.                                       A 297
C                                                                          A 298
C                                                                          A 299
C     INPUT:     FOR EACH INTERSECTION, FOUR CARDS DESCRIBING              A 300
```

```
C                TRAFFIC FLOW AND THE NUMBER OF LANES, FOLLOWED          A 301
C                BY TWO CARDS DESCRIBING THE PHASE SEQUENCE              A 302
C                OF THE LIGHT. (A BLANK CARD INDICATES THE END           A 303
C                OF THE DATA.)                                           A 304
C                                                                        A 305
C                                                                        A 306
C      CARD 1: (DESCRIPTION OF SOUTHBOUND TRAFFIC AT THE INTERSECTION)   A 307
C                                                                        A 308
C                                                                        A 309
C      COL.                                                              A 310
C                                                                        A 311
C      2-4             INTERSECTION NUMBER.                              A 312
C      8               CARD NUMBER (1).                                  A 313
C      9-12            INTERSECTION TO THE SOUTH.                        A 314
C      13-18           VEHICLES PER HOUR GOING STRAIGHT SOUTH THROUGH    A 315
C                      THE INTERSECTION.                                 A 316
C      19-24           VEHICLES PER HOUR TURNING LEFT FROM THE SOUTH-    A 317
C                      BOUND LEG.                                        A 318
C      25-30           VEHICLES PER HOUR TURNING RIGHT FROM THE SOUTH-   A 319
C                      BOUND LEG.                                        A 320
C      31-33           NUMBER OF SOUTHBOUND LANES (INCLUDING TURN ONLY   A 321
C                      LANES).                                           A 322
C      34-36           NUMBER OF LEFT TURN ONLY LANES FROM SOUTHBOUND LEG. A 323
C      37-39           NUMBER OF RIGHT TURN ONLY LANES FROM SOUTH-       A 324
C                      BOUND LEG.                                        A 325
C      57-80           ALPHANUMERIC IDENTIFICATION OF INTERSECTION.      A 326
C                                                                        A 327
C                                                                        A 328
C      CARD 2: (DESCRIPTION OF WESTBOUND TRAFFIC AT THE INTERSECTION)    A 329
C                                                                        A 330
C      COL.                                                              A 331
C                                                                        A 332
C      2-4             INTERSECTION NUMBER.                              A 333
C      8               CARD NUMBER (2).                                  A 334
C      9-12            INTERSECTION TO THE WEST.                         A 335
C      13-18           VEHICLES PER HOUR GOING STRAIGHT WEST THROUGH     A 336
C                      THE INTERSECTION.                                 A 337
C      19-24           VEHICLES PER HOUR TURNING LEFT FROM THE WEST-     A 338
C                      BOUND LEG.                                        A 339
C      25-30           VEHICLES PER HOUR TURNING RIGHT FROM THE WEST-    A 340
C                      BOUND LEG.                                        A 341
C      31-33           NUMBER OF WESTBOUND LANES (INCLUDING TURN ONLY    A 342
C                      LANES).                                           A 343
C      34-36           NUMBER OF LEFT TURN ONLY LANES FROM WEST-         A 344
C                      BOUND LEG.                                        A 345
C      37-39           NUMBER OF RIGHT TURN ONLY LANES FROM WEST-        A 346
C                      BOUND LEG.                                        A 347
C      57-80           ALPHANUMERIC IDENTIFICATION OF INTERSECTION.      A 348
C                                                                        A 349
C                                                                        A 350
```

```
C      CARD 3: (DESCRIPTION OF NORTHBOUND TRAFFIC AT THE INTERSECTION)      A 351
C                                                                           A 352
C      COL.                                                                 A 353
C                                                                           A 354
C      2-4            INTERSECTION NUMBER.                                 A 355
C      8              CARD NUMBER (3).                                     A 356
C      9-12           INTERSECTION TO THE NORTH.                           A 357
C      13-18          VEHICLES PER HOUR GOING STRAIGHT NORTH THROUGH       A 358
C                     THE INTERSECTION.                                    A 359
C      19-24          VEHICLES PER HOUR TURNING LEFT FROM THE NORTH-       A 360
C                     BOUND LEG.                                           A 361
C      25-30          VEHICLES PER HOUR TURNING RIGHT FROM THE NORTH-      A 362
C                     BOUND LEG.                                           A 363
C      31-33          NUMBER OF NORTHBOUND LANES (INCLUDING TURN ONLY      A 364
C                     LANES).                                              A 365
C      34-36          NUMBER OF LEFT TURN ONLY LANES FROM NORTHOBUND       A 366
C                     LEG.                                                 A 367
C      37-39          NUMBER OF RIGHT TURN ONLY LANES FROM NORTHBOUND      A 368
C                     LEG.                                                 A 369
C      57-80          ALPHANUMERIC IDENTIFICATION OF INTERSECTION.         A 370
C                                                                           A 371
C                                                                           A 372
C      CARD 4: (DESCRIPTION OF EASTBOUND TRAFFIC AT THE INTERSECTION)       A 373
C                                                                           A 374
C      COL.                                                                 A 375
C                                                                           A 376
C      2-4            INTERSECTION NUMBER.                                 A 377
C      8              CARD NUMBER (4).                                     A 378
C      9-12           INTERSECTION TO THE EAST.                            A 379
C      13-18          VEHICLES PER HOUR GOING STRAIGHT EAST THROUGH        A 380
C                     THE INTERSECTION.                                    A 381
C      19-24          VEHICLES PER HOUR TURNING LEFT FROM THE EAST-        A 382
C                     BOUND LEG.                                           A 383
C      25-30          VEHICLES PER HOUR TURNING RIGHT FROM THE EAST-       A 384
C                     BOUND LEG.                                           A 385
C      31-33          NUMBER OF EASTBOUND LANES (INCLUDING TURN ONLY       A 386
C                     LANES).                                              A 387
C      34-36          NUMBER OF LEFT TURN ONLY LANES FROM EAST-            A 388
C                     BOUND LEG.                                           A 389
C      37-39          NUMBER OF RIGHT TURN ONLY LANES FROM EAST-           A 390
C                     BOUND LEG.                                           A 391
C      57-80          ALPHANUMERIC IDENTIFICATION OF THE INTERSECTION.     A 392
C                                                                           A 393
C                                                                           A 394
C      CARD 5: (FIRST CARD DESCRIBING THE PATTERN OF THE LIGHT)             A 395
C                                                                           A 396
C      COL.                                                                 A 397
C                                                                           A 398
C      2-4            INTERSECTION NUMBER.                                 A 399
C      6              CARD NUMBER (5).                                     A 400
```

```
C       7-9             NUMBER OF THE FIRST PHASE THE LIGHT ENTERS.                A 401
C       10-12           MINIMUM ALLOTMENT TO THIS PHASE (IN TENTHS OF              A 402
C                       SECONDS WITH NO DECIMAL POINT).                            A 403
C       13-15           PRESENT ALLOTMENT TO THIS PHASE (IN TENTHS OF              A 404
C                       SECONDS WITH NO DECIMAL POINT).                            A 405
C       16-18           NUMBER OF THE SECOND PHASE THE LIGHT ENTERS.               A 406
C       19-21           MINIMUM ALLOTMENT TO THIS PHASE (IN TENTHS OF              A 407
C                       SECONDS WITH NO DECIMAL POINT).                            A 408
C       22-24           PRESENT ALLOTMENT TO THIS PHASE (IN TENTHS OF              A 409
C                       SECONDS WITH NO DECIMAL POINT).                            A 410
C       25-33           DESCRIPTION OF THIRD PHASE OF LIGHT.                       A 411
C       34-42           DESCRIPTION OF FOURTH PHASE OF LIGHT.                      A 412
C       43-51           DESCRIPTION OF FIFTH PHASE OF LIGHT.                       A 413
C       57-80           ALPHANUMERIC IDENTIFICATION OF THE INTERSECTION.           A 414
C                                                                                  A 415
C                                                                                  A 416
C       CARD 6: (SECOND CARD DESCRIBING THE PATTERN OF THE LIGHT)                  A 417
C                                                                                  A 418
C       COL.                                                                       A 419
C                                                                                  A 420
C       2-4             INTERSECTION NUMBER.                                       A 421
C       6               CARD NUMBER (6).                                           A 422
C       7-9             NUMBER OF THE SIXTH PHASE THE LIGHT ENTERS.                A 423
C       10-12           MINIMUM ALLOTMENT TO THIS PHASE (IN TENTHS                 A 424
C                       OF SECONDS WITH NO DECIMAL POINT).                         A 425
C       13-15           PRESENT ALLOTMENT TO THIS PHASE (IN TENTHS OF              A 426
C                       SECONDS WITH NO DECIMAL POINT).                            A 427
C       16-24           DESCRIPTION OF THE SEVENTH PHASE.                          A 428
C       25-33           DESCRIPTION OF THE EIGHTH PHASE.                           A 429
C       34-42           DESCRIPTION OF THE NINTH PHASE.                            A 430
C       43-51           DESCRIPTION OF THE TENTH PHASE.                            A 431
C       57-80           ALPHANUMERIC IDENTIFICATION OF THE INTERSECTION.           A 432
C                                                                                  A 433
C                                                                                  A 434
C       OUTPUT:         FOR THE INTERSECTION AN OPTIMAL ALLOCATION                 A 435
C                       (IF THE PROBLEM IS FEASIBLE) OF THE CYCLE                  A 436
C                       TO THE VARIOUS PHASES IS PRODUCED.  WARNINGS               A 437
C                       ARE GENERATED IF THE DATA APPEARS INCORRECT.               A 438
C                       CARDS SUITABLE FOR USE IN THE SUBROUTINE SIMU              A 439
C                       ARE ALSO PRODUCED.  (ONE CARD IS PRODUCED FOR EVERY        A 440
C                       EDGE INCIDENT WITH THE INTERSECTION.)  THE FORMAT          A 441
C                       OF THE CARDS IS AS FOLLOWS:                                A 442
C                                                                                  A 443
C       COL.                                                                       A 444
C                                                                                  A 445
C       2-4             INTERSECTION NUMBER (I SAY).                               A 446
C       5-7             NUMBER OF THE INTERSECTION AT THE                          A 447
C                       OTHER END OF THE LEG (J SAY).                              A 448
C       8-17            TIME IN CYCLE THAT LIGHT I TURNS GREEN                     A 449
C                       FOR VEHICLES ENTERING THE LEG (F10.4).                     A 450
```

```
C      18-27          TIME IN CYCLE THAT LIGHT I TURNS AMBER                    A 451
C                     FOR VEHICLES ENTERING THE LEG.                            A 452
C      28-37          TIME IN CYCLE THAT LIGHT I TURNS RED                      A 453
C                     FOR VEHICLES ENTERING THE LEG.                            A 454
C      38-47          TIME IN CYCLE THAT LIGHT I TURNS GREEN                    A 455
C                     FOR VEHICLES LEAVING THE LEG.                             A 456
C      48-57          TIME IN CYCLE THAT LIGHT I TURNS AMBER                    A 457
C                     FOR VEHICLES LEAVING THE LEG.                             A 458
C      58-67          TIME IN CYCLE THAT LIGHT I TURNS RED                      A 459
C                     FOR VEHICLES LEAVING THE LEG.                             A 460
C                                                                               A 461
C                                                                               A 462
C      CALLED ROUTINES:                                                         A 463
C                                                                               A 464
C                     GJELIM     (INDIRECTLY)                                   A 465
C                     SIMPLX                                                    A 466
C                     MINRAT     (INDIRECTLY)                                   A 467
C                     SWAP       (INDIRECTLY)                                   A 468
C                                                                               A 469
C                                                                               A 470
      COMMON A,TOLER,N,NP1,NP2,NP3,NP4,M,MP1,MP2,MP3,MP4                        A 471
      COMMON LABEL                                                              A 472
C                                                                               A 473
C                                                                               A 474
      REAL UNOPP(10,3),OPP(10,3),RED(10,3)                                      A 475
      REAL TUNOPP(3),TOPP(3),TRED(3)                                            A 476
C                                                                               A 477
C                                                                               A 478
      INTEGER VEL(10),ITYPE(10),VTLFL(10),VTRFL(10),VGSFL(10)                   A 479
      REAL NBTG,NBTA,NBTR,ALOT(20),WPS(20),NOW(20)                              A 480
      INTEGER ICON(20,4)/1,3,2,3,3,3,1,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3   A 481
     1,3,3,3,3,3,3,3,1,2,1,1,3,3,3,3,3,3,1,2,1,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3   A 482
     2,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,2,3,3,3,1,1,3,3/                           A 483
      REAL C(3,4,20),AMIN(20),TF(4,3,3),E(20),D(3,4)                            A 484
      INTEGER IDIR(4),ISUB(20),ITEMP(15),IDENT(6)                               A 485
      DOUBLE PRECISION A(15,35),TOLER,LABEL(30),TEM                             A 486
C                                                                               A 487
C     INITIALIZATION SECTION                                                    A 488
C                                                                               A 489
      TOLER=0.00001                                                             A 490
      NSIC=60                                                                   A 491
C                                                                               A 492
C     START OF LOOP FOR EACH INTERSECTION OF THE SYSTEM                         A 493
C                                                                               A 494
1     CONTINUE                                                                  A 495
      FLOW=0.95                                                                 A 496
      WRITE (6,70)                                                              A 497
C                                                                               A 498
C     FOR EACH LEG INTO THE INTERSECTION READ IN THE TRAFFIC FLOW INFO          A 499
C                                                                               A 500
```

```
      DO 33 J1=1,4                                                          A 501
      READ (5,71) IN,LN,IDIR(J1),VGSL,VTLL,VTRL,NLT,NLL,NRL,IDENT           A 502
      WRITE (6,72) IN,LN,IDIR(J1),VGSL,VTLL,VTRL,NLT,NLL,NRL,IDENT          A 503
C                                                                           A 504
CC    CHECK IF ALL THE INTERSECTIONS HAVE BEEN DONE                         A 505
C                                                                           A 506
      IF (IN.EQ.0) GO TO 69                                                 A 507
C                                                                           A 508
CC    CALCULATE THE MINIMUM TRAFFIC FLOW CONSTRAINTS (IE. THE DIJ'S)        A 509
C                                                                           A 510
      D(1,J1)=VTLL                                                          A 511
      D(2,J1)=VTRL                                                          A 512
      D(3,J1)=VGSL                                                          A 513
C                                                                           A 514
CC    CALCULATE THE POSSIBLE NUMBER OF VEHICLE STARTS PER SECOND OF         A 515
CC    TYPE I FROM LEG J1 DURING A PHASE OF TYPE K                           A 516
C                                                                           A 517
      IF (NLT.GT.1) GO TO 5                                                 A 518
      IF (NLT.GT.0) GO TO 4                                                 A 519
      DO 3 I=1,3                                                            A 520
      DO 2 K=1,3                                                            A 521
      TF(J1,I,K)=0.0                                                        A 522
2     CONTINUE                                                              A 523
3     CONTINUE                                                              A 524
      GO TO 33                                                              A 525
C                                                                           A 526
CC    DETERMINE THE TYPE OF EACH LANE                                       A 527
C                                                                           A 528
4     CONTINUE                                                              A 529
      ITYPE(NLT)=6                                                          A 530
      GO TO 7                                                               A 531
5     CONTINUE                                                              A 532
      DO 6 I=1,NLT                                                          A 533
      ITYPE(I)=3                                                            A 534
6     CONTINUE                                                              A 535
      ITYPE(1)=1                                                            A 536
      IF (NLL.EQ.0) ITYPE(1)=4                                              A 537
      ITYPE(NLT)=2                                                          A 538
      IF (NRL.EQ.0) ITYPE(NLT)=5                                            A 539
7     CONTINUE                                                              A 540
C                                                                           A 541
CC    CALCULATE THE NUMBER OF VEHICLES ENTERING EACH LANE                   A 542
C                                                                           A 543
      VEL(1)=0                                                              A 544
      VEL(NLT)=0                                                            A 545
      IF (NLL.EQ.1) VEL(1)=VTLL                                             A 546
      IF (NRL.EQ.1) VEL(NLT)=VTRL                                           A 547
      NLS=NLT-NLL-NRL                                                       A 548
      VT=VTLL+VTRL+VGSL-VEL(1)-VEL(NLT)                                     A 549
      W=VT/NLS                                                              A 550
```

```
      IF (NLL.NE.1) VEL(1)=W                                          A 551
      IF (NRL.NE.1) VEL(NLT)=W                                        A 552
      NLTM1=NLT-1                                                     A 553
      IF (NLTM1.LT.2) GO TO 9                                         A 554
      DO 8 I=2,NLTM1                                                  A 555
      VEL(I)=W                                                        A 556
8     CONTINUE                                                        A 557
9     CONTINUE                                                        A 558
C                                                                     A 559
C     CALCULATE THE NUMBER OF VEHICLES TURNING LEFT FOR EACH LANE     A 560
C                                                                     A 561
      DO 10 I=1,NLT                                                   A 562
      VTLFL(I)=0                                                      A 563
10    CONTINUE                                                        A 564
      VTLFL(1)=VTLL                                                   A 565
C                                                                     A 566
C                                                                     A 567
C     CALCULATE THE NUMBER OF VEHICLES TURNING RIGHT FOR EACH LANE    A 568
C                                                                     A 569
      DO 11 I=1,NLT                                                   A 570
      VTRFL(I)=0                                                      A 571
11    CONTINUE                                                        A 572
      VTRFL(NLT)=VTRL                                                 A 573
C                                                                     A 574
C     CALCULATE THE NUMBER OF VEHICLES GOING STRAIGHT FOR EACH LANE   A 575
C                                                                     A 576
      DO 12 I=1,NLT                                                   A 577
      VGSFL(I)=VEL(I)-VTLFL(I)-VTRFL(I)                               A 578
12    CONTINUE                                                        A 579
C                                                                     A 580
C     FOR EACH LANE CALCULATE THE POSSIBLE NUMBER OF VEHICLE STARTS   A 581
C     BASED ON THE TYPE OF THE LANE AND THE NUMBER OF VEHICLES        A 582
C     TURNING LEFT, RIGHT OR GOING STRAIGHT.                          A 583
C                                                                     A 584
      DO 29 I=1,NLT                                                   A 585
      VL=VTLFL(I)                                                     A 586
      VR=VTRFL(I)                                                     A 587
      VS=VGSFL(I)                                                     A 588
C                                                                     A 589
C     FOR LEFT TURNS                                                  A 590
C                                                                     A 591
      J=1                                                             A 592
      ITYP=ITYPE(I)                                                   A 593
      GO TO (13,14,14,15,14,16), ITYP                                 A 594
C                                                                     A 595
C     FOR LEFT TURN ONLY LANE                                         A 596
C                                                                     A 597
13    UNOPP(I,J)=0.5                                                  A 598
      OPP(I,J)=0.1                                                    A 599
      RED(I,J)=0.05                                                   A 600
```

```
      GO TO 17                                                          A 601
C                                                                       A 602
C     FOR ANY LANE ALLOWING RIGHT TURNS                                 A 603
C                                                                       A 604
14    UNOPP(I,J)=0.05                                                   A 605
      OPP(I,J)=0.05                                                     A 606
      RED(I,J)=0.05                                                     A 607
      GO TO 17                                                          A 608
C                                                                       A 609
C     FOR A LANE ALLOWING LEFT TURNS AND STRAIGHT THROUGH TRAFFIC       A 610
C                                                                       A 611
15    UNOPP(I,J)=VL/(2.0*VL+2.0*VS)                                     A 612
      OPP(I,J)=VL/(10*VL+2*VS)                                          A 613
      RED(I,J)=0.05                                                     A 614
      GO TO 17                                                          A 615
C                                                                       A 616
C     FOR A LANE ALLOWING ALL 3 TYPES OF MOVEMENTS                      A 617
C                                                                       A 618
16    UNOPP(I,J)=VL/(2.0*VL+2*VS+7*VR)                                  A 619
      OPP(I,J)=VL/(10.0*VL+2.0*VS+7.0*VR)                               A 620
      RED(I,J)=0.05                                                     A 621
17    CONTINUE                                                          A 622
C                                                                       A 623
C     FOR VEHICLES TURNING RIGHT                                        A 624
C                                                                       A 625
      J=2                                                               A 626
      GO TO (18,19,18,18,20,21), ITYP                                   A 627
C                                                                       A 628
C     FOR LEFT TURN ONLY LANES, STRAIGHT THROUGH ONLY LANES,            A 629
C     AND LANES ALLOWING LEFT AND STRAIGHT THROUGH TRAFFIC.             A 630
C                                                                       A 631
18    UNOPP(I,J)=0.05                                                   A 632
      OPP(I,J)=0.05                                                     A 633
      RED(I,J)=0.05                                                     A 634
      GO TO 22                                                          A 635
C                                                                       A 636
C     FOR RIGHT TURN ONLY LANES                                         A 637
C                                                                       A 638
19    UNOPP(I,J)=0.142857                                               A 639
      OPP(I,J)=0.142857                                                 A 640
      RED(I,J)=0.142857                                                 A 641
      GO TO 22                                                          A 642
C                                                                       A 643
C     FOR LANES ALLOWING RIGHT TURNS AND STRAIGHT THROUGH TRAFFIC       A 644
C                                                                       A 645
20    UNOPP(I,J)=VR/(7.0*VR+2.0*VS)                                     A 646
      OPP(I,J)=VR/(7.0*VR+2.0*VS)                                       A 647
      RED(I,J)=VR/(7*VR+20.0*VS)                                        A 648
      GO TO 22                                                          A 649
C                                                                       A 650
```

```
C       FOR LANES ALLOWING ALL TYPES OF MOVEMENTS                       A 651
C                                                                       A 652
21      UNOPP(I,J)=VR/(7.0*VR+2.0*VS+2.0*VL)                            A 653
        OPP(I,J)=VR/(7.0*VR+2.0*VS+10.0*VL)                             A 654
        RED(I,J)=VR/(7*VR+20*VS+20*VL)                                  A 655
22      CONTINUE                                                        A 656
C                                                                       A 657
C       FOR VEHICLES GOING STRAIGHT                                     A 658
C                                                                       A 659
        J=3                                                             A 660
        GO TO (23,23,24,25,26,27), ITYP                                 A 661
C                                                                       A 662
C       FOR LANES ALLOWING ONLY LEFT TURNS AND RIGHT TURNS              A 663
C                                                                       A 664
23      UNOPP(I,J)=0.05                                                 A 665
        OPP(I,J)=0.05                                                   A 666
        RED(I,J)=0.05                                                   A 667
        GO TO 28                                                        A 668
C                                                                       A 669
C       FOR LANES ALLOWING ONLY STRAIGHT THROUGH TRAFFIC                A 670
C                                                                       A 671
24      UNOPP(I,J)=0.5                                                  A 672
        OPP(I,J)=0.5                                                    A 673
        RED(I,J)=0.05                                                   A 674
        GO TO 28                                                        A 675
C                                                                       A 676
C       FOR A LANE ALLOWING LEFT AND STRAIGHT THROUGH TRAFFIC           A 677
C                                                                       A 678
25      UNOPP(I,J)=VS/(2*VS+2*VL)                                       A 679
        OPP(I,J)=VS/(2*VS+10*VL)                                        A 680
        RED(I,J)=0.05                                                   A 681
        GO TO 28                                                        A 682
C                                                                       A 683
C       FOR LANES ALLOWING RIGHT AND STRAIGHT THROUGH TRAFFIC           A 684
C                                                                       A 685
26      UNOPP(I,J)=VS/(2*VS+7*VR)                                       A 686
        OPP(I,J)=VS/(2*VS+7*VR)                                         A 687
        RED(I,J)=0.05                                                   A 688
        GO TO 28                                                        A 689
C                                                                       A 690
C       FOR LANES ALLOWING ALL TYPES OF MOVEMENTS                       A 691
C                                                                       A 692
27      UNOPP(I,J)=VS/(2*VS+2.0*VL+7*VR)                                A 693
        OPP(I,J)=VS/(2*VS+10*VL+7*VR)                                   A 694
        RED(I,J)=0.05                                                   A 695
28      CONTINUE                                                        A 696
C                                                                       A 697
C       CALCULATE TOTAL NUMBER OF VEHICLE STARTS ALLOWED DURING THE     A 698
C       THREE TYPES OF PHASES                                           A 699
C                                                                       A 700
```

```
29      CONTINUE                                                        A 701
        DO 31 J=1,3                                                     A 702
        TUNOPP(J)=0                                                     A 703
        TOPP(J)=0                                                       A 704
        TRED(J)=0                                                       A 705
        DO 30 I=1,NLT                                                   A 706
        TUNOPP(J)=TUNOPP(J)+UNOPP(I,J)                                  A 707
        TOPP(J)=TOPP(J)+OPP(I,J)                                        A 708
        TRED(J)=TRED(J)+RED(I,J)                                        A 709
30      CONTINUE                                                        A 710
31      CONTINUE                                                        A 711
        DO 32 J2=1,3                                                    A 712
C                                                                       A 713
C       RECORD THE NUMBER OF VEHICLE STARTS OF TYPE J2 FROM LEG J1      A 714
C       WHEN THAT LEG IS IN A PHASE OF TYPE 1,2, AND 3.                 A 715
C                                                                       A 716
        TF(J1,1,J2)=TUNOPP(J2)                                          A 717
        TF(J1,2,J2)=TOPP(J2)                                            A 718
        TF(J1,3,J2)=TRED(J2)                                            A 719
32      CONTINUE                                                        A 720
C                                                                       A 721
C       END OF DATA MASSAGING FOR ONE LEG INTO AN INTERSECTION          A 722
C                                                                       A 723
33      CONTINUE                                                        A 724
C                                                                       A 725
C       FOR EACH OF THE POSSIBLE PHASES . . .                           A 726
C                                                                       A 727
        DO 36 K=1,20                                                    A 728
C                                                                       A 729
C       FOR EACH OF THE LEGS GOING INTO AN INTERSECTION . . .           A 730
C                                                                       A 731
        DO 35 J=1,4                                                     A 732
C                                                                       A 733
C       FOR EACH TYPE OF VEHICLE MOVEMENT . . .                         A 734
C                                                                       A 735
        DO 34 I=1,3                                                     A 736
C                                                                       A 737
C       CALCULATE THE CONSTANTS C(I,J,K)                                A 738
C                                                                       A 739
        ICOLJ=ICON(K,J)                                                 A 740
        C(I,J,K)=TF(J,ICOLJ,I)                                          A 741
34      CONTINUE                                                        A 742
35      CONTINUE                                                        A 743
36      CONTINUE                                                        A 744
C                                                                       A 745
C       FOR EACH OF THE PHASES SET THE MINIMUM AND PRESENT TIME         A 746
C       ALLOTMENTS TO ZERO                                              A 747
C                                                                       A 748
        DO 37 K=1,20                                                    A 749
        AMIN(K)=0.0                                                     A 750
```

```
      NOW(K)=0.0                                                        A 751
37    CONTINUE                                                          A 752
C                                                                       A 753
C     READ IN THE PRESENT AND MINIMUM TIME ALLOTMENTS FOR THE PHASES    A 754
C                                                                       A 755
      WRITE (6,73)                                                      A 756
      DO 39 I=1,2                                                       A 757
      READ (5,74) INT,LN,(ITEMP(J),J=1,15),IDENT                        A 758
      WRITE (6,75) INT,LN,(ITEMP(J),J=1,15),IDENT                       A 759
      DO 38 J=1,5                                                       A 760
      IJK=(J-1)*3+1                                                     A 761
      IF (ITEMP(IJK).EQ.0) GO TO 38                                     A 762
      AMIN(ITEMP(IJK))=ITEMP(IJK+1)/10.                                 A 763
      NOW(ITEMP(IJK))=ITEMP(IJK+2)/10.                                  A 764
38    CONTINUE                                                          A 765
39    CONTINUE                                                          A 766
      WRITE (6,73)                                                      A 767
C                                                                       A 768
C     FOR EACH OF THE PHASES DETERMINE THE TOTAL NUMBER OF VEHICLE      A 769
C     STARTS POSSIBLE PER SECOND                                        A 770
C                                                                       A 771
      DO 42 K=1,20                                                      A 772
      SUM=0.0                                                           A 773
      DO 41 J=1,4                                                       A 774
      DO 40 I=1,3                                                       A 775
      SUM=SUM+C(I,J,K)                                                  A 776
40    CONTINUE                                                          A 777
41    CONTINUE                                                          A 778
      E(K)=SUM                                                          A 779
42    CONTINUE                                                          A 780
C                                                                       A 781
C     CALCULATE THE NUMBER OF PHASES THE LIGHT GOES THROUGH             A 782
C     AND RECORD EACH OF THESE                                          A 783
C                                                                       A 784
      NP=0                                                              A 785
      WRITE (6,76)                                                      A 786
C                                                                       A 787
C     CALCULATE THE TOTAL TIME PRESENTLY ALOTTED                        A 788
C                                                                       A 789
      SUMN=0.0                                                          A 790
      DO 43 K=1,20                                                      A 791
      IF (NOW(K).EQ.0) GO TO 43                                         A 792
      SUMN=SUMN+NOW(K)                                                  A 793
      WRITE (6,77) K,AMIN(K),NOW(K)                                     A 794
      NP=NP+1                                                           A 795
      ISUB(NP)=K                                                        A 796
43    CONTINUE                                                          A 797
C                                                                       A 798
C     CHECK IF THE TOTAL TIME ALLOTTED IS "EQUAL" TO THE CYCLE LENGTH   A 799
C                                                                       A 800
```

```
      IF (ABS(SUMN-NSIC).GT.0.01) WRITE (6,78)                          A 801
C                                                                       A 802
C     CHECK IF PRESENT ALLOTMENT FOR EACH PHASE IS INSUFFICIENT         A 803
C                                                                       A 804
      DO 44 K=1,20                                                      A 805
      IF (NOW(K).LT.AMIN(K)) WRITE (6,79) K                             A 806
44    CONTINUE                                                          A 807
45    CONTINUE                                                          A 808
C                                                                       A 809
C     FOR EACH TYPE OF VEHICLE MOVEMENT . . .                           A 810
C                                                                       A 811
      DO 48 I=1,3                                                       A 812
C                                                                       A 813
C     FOR EACH LEG INTO THE INTERSECTION . . .                          A 814
C                                                                       A 815
      DO 47 J=1,4                                                       A 816
C                                                                       A 817
C     CALCULATE THE TOTAL NUMBER OF VEHICLES PRESENTLY ABLE TO USE THE  A 818
C     INTERSECTION IN EACH PHASE OF THE LIGHT.                          A 819
C                                                                       A 820
      SUM=0.0                                                           A 821
      DO 46 K=1,20                                                      A 822
      SUM=SUM+C(I,J,K)*NOW(K)                                           A 823
46    CONTINUE                                                          A 824
C                                                                       A 825
C     CHECK IF INSUFFICIENT FLOW                                        A 826
C                                                                       A 827
      IF (SUM.LT.D(I,J)*FLOW/3600*NSIC) WRITE (6,80) I,J                A 828
47    CONTINUE                                                          A 829
48    CONTINUE                                                          A 830
C                                                                       A 831
C     PREPARE TO SET UP THE SIMPLEX TABLEAU                             A 832
C                                                                       A 833
      NPP2=NP+2                                                         A 834
      NPP14=NP+14                                                       A 835
      M=NP                                                              A 836
      N=2*NP+14                                                         A 837
      MP1=M+1                                                           A 838
      MP2=M+2                                                           A 839
      MP3=M+3                                                           A 840
      MP4=M+4                                                           A 841
      NP1=N+1                                                           A 842
      NP2=N+2                                                           A 843
      NP3=N+3                                                           A 844
C                                                                       A 845
C     ZERO OUT THE ENTIRE TABLEAU                                       A 846
C                                                                       A 847
      DO 50 I=1,MP4                                                     A 848
      DO 49 J=1,NP3                                                     A 849
      A(I,J)=0.0                                                        A 850
```

```
49      CONTINUE                                                        A  851
50      CONTINUE                                                        A  852
C                                                                       A  853
C       FILL IN THE REST OF THE TABLEAU                                 A  854
C                                                                       A  855
        DO 52 I=1,M                                                     A  856
        A(I,1)=-1.0D0                                                   A  857
        A(I,2)=1.0D0                                                    A  858
        A(I,I+2)=-1.0                                                   A  859
        A(I,NPP14+I)=-1.0                                               A  860
        A(I,NP1)=E(ISUB(I))                                             A  861
        A(I,NP2)=-100-I                                                 A  862
        DO 51 J=1,12                                                    A  863
        IP=(J-1)/4+1                                                    A  864
        JP=J-(IP-1)*4                                                   A  865
        KP=ISUB(I)                                                      A  866
        A(I,J+NPP2)=-C(IP,JP,KP)                                        A  867
51      CONTINUE                                                        A  868
52      CONTINUE                                                        A  869
        DO 53 J=1,NPP14                                                 A  870
        A(MP3,J)=J                                                      A  871
53      CONTINUE                                                        A  872
        A(MP4,1)=-NSIC                                                  A  873
        A(MP4,2)=NSIC                                                   A  874
        DO 54 J=1,NP                                                    A  875
        A(MP4,J+2)=-AMIN(ISUB(J))                                       A  876
        IP=(J-1)/4+1                                                    A  877
        JP=J-(IP-1)*4                                                   A  878
        A(MP4,J+NPP2)=-(D(IP,JP)*FLOW/3600)*NSIC                        A  879
54      CONTINUE                                                        A  880
        DO 55 I=1,M                                                     A  881
        LABEL(I)=A(I,NP2)                                               A  882
55      CONTINUE                                                        A  883
C                                                                       A  884
C       PERFORM THE SIMPLEX ROUTINE                                     A  885
C                                                                       A  886
        CALL SIMPLX                                                     A  887
C                                                                       A  888
C       CHECK IF PROBLEMS ENCOUNTERED IN SIMPLEX ROUTINE                A  889
C                                                                       A  890
        ID=A(MP4,NP3)                                                   A  891
        IF (ID.NE.1) GO TO 67                                           A  892
C                                                                       A  893
C       FOR EACH PHASE ZERO THE ALLOTMENT                               A  894
C                                                                       A  895
        DO 56 K=1,20                                                    A  896
        ALOT(K)=0.0                                                     A  897
56      CONTINUE                                                        A  898
C                                                                       A  899
C       DETERMINE THE ALLOTMENTS TO THE PHASES                          A  900
```

```
C                                                                       A 901
      WRITE (6,81)                                                      A 902
      DO 58 I=1,M                                                       A 903
      DO 57 J=1,N                                                       A 904
      IF (A(MP3,J).NE.LABEL(I)) GO TO 57                                A 905
      TEM=A(MP1,J)                                                      A 906
      WRITE (6,82) ISUB(I),TEM                                          A 907
      ALOT(ISUB(I))=TEM                                                 A 908
57    CONTINUE                                                          A 909
58    CONTINUE                                                          A 910
      WRITE (6,73)                                                      A 911
C                                                                       A 912
C     CALCULATE THE PRESENT TOTAL FLOW AND THE PROJECTED FLOW           A 913
C                                                                       A 914
      Z1=0.0                                                            A 915
      Z2=0.0                                                            A 916
      DO 59 K=1,20                                                      A 917
      Z1=Z1+E(K)*NOW(K)                                                 A 918
      Z2=Z2+E(K)*ALOT(K)                                                A 919
59    CONTINUE                                                          A 920
C                                                                       A 921
C     DETERMINE THE PERCENTAGE INCREASE                                 A 922
C                                                                       A 923
      Z3=Z2/Z1*100-100.0                                                A 924
      WRITE (6,83) Z1,Z2,Z3                                             A 925
      TEM=-A(MP1,NP1)                                                   A 926
C                                                                       A 927
C     DETERMINE WHEN IN EACH CYCLE EACH OF THE PHASES BEGINS            A 928
C                                                                       A 929
      WPS(1)=0.0                                                        A 930
      DO 60 K=2,20                                                      A 931
      WPS(K)=WPS(K-1)+ALOT(K-1)                                         A 932
60    CONTINUE                                                          A 933
C                                                                       A 934
C     DETERMINE WHEN THE VARIOUS LIGHTS CHANGE COLOUR                   A 935
C                                                                       A 936
      SBTG=0.0                                                          A 937
      NBTG=0.0                                                          A 938
      IF (ALOT(1).NE.0.0) NBTG=WPS(3)                                   A 939
      IF (ALOT(2).NE.0.0) SBTG=WPS(3)                                   A 940
      IF (ALOT(8).NE.0.0) GO TO 61                                      A 941
      IF (ALOT(5).NE.0.0) GO TO 62                                      A 942
      SBTA=WPS(10)                                                      A 943
      NBTA=SBTA                                                         A 944
      SBTR=SBTA+ALOT(10)                                                A 945
      NBTR=SBTR                                                         A 946
      GO TO 63                                                          A 947
61    NBTA=WPS(7)                                                       A 948
      NBTR=WPS(8)                                                       A 949
      SBTA=WPS(9)                                                       A 950
```

```
      SBTR=SBTA+ALOT(9)                                                    A  951
      GO TO 63                                                             A  952
62    SBTA=WPS(4)                                                          A  953
      SBTR=WPS(5)                                                          A  954
      NBTA=WPS(6)                                                          A  955
      NBTR=NBTA+ALOT(6)                                                    A  956
63    CONTINUE                                                             A  957
      WG=NBTR                                                              A  958
      IF (NBTR.LT.SBTR) WG=SBTR                                            A  959
      EBTG=WG                                                              A  960
      WBTG=WG                                                              A  961
      IF (ALOT(11).NE.0.0) WBTG=WPS(13)                                    A  962
      IF (ALOT(12).NE.0.0) EBTG=WPS(13)                                    A  963
      IF (ALOT(18).NE.0.0) GO TO 64                                        A  964
      IF (ALOT(15).NE.0.0) GO TO 65                                        A  965
      EBTA=WPS(20)                                                         A  966
      WBTA=EBTA                                                            A  967
      EBTR=EBTA+ALOT(20)                                                   A  968
      WBTR=EBTR                                                            A  969
      GO TO 66                                                             A  970
64    WBTA=WPS(17)                                                         A  971
      WBTR=WPS(18)                                                         A  972
      EBTA=WPS(19)                                                         A  973
      EBTR=EBTA+ALOT(19)                                                   A  974
      GO TO 66                                                             A  975
65    EBTA=WPS(14)                                                         A  976
      EBTR=WPS(15)                                                         A  977
      WBTA=WPS(16)                                                         A  978
      WBTR=WBTA+ALOT(16)                                                   A  979
66    CONTINUE                                                             A  980
C                                                                          A  981
C     WRITE OUT THE DATA FOR USE IN SUBROUTINE SIMU                        A  982
C                                                                          A  983
      WRITE (6,84)                                                         A  984
      IF (IDIR(1).NE.0) WRITE (6,85) IN,IDIR(1),NBTG,NBTA,NBTR,SBTG,SBTA   A  985
     1,SBTR                                                                A  986
      IF (IDIR(2).NE.0) WRITE (6,85) IN,IDIR(2),WBTG,WBTA,WBTR,EBTG,EBTA   A  987
     1,EBTR                                                                A  988
      IF (IDIR(3).NE.0) WRITE (6,85) IN,IDIR(3),SBTG,SBTA,SBTR,NBTG,NBTA   A  989
     1,NBTR                                                                A  990
      IF (IDIR(4).NE.0) WRITE (6,85) IN,IDIR(4),EBTG,EBTA,EBTR,WBTG,WBTA   A  991
     1,WBTR                                                                A  992
      IF (IDIR(1).NE.0) WRITE (7,85) IN,IDIR(1),NBTG,NBTA,NBTR,SBTG,SBTA   A  993
     1,SBTR                                                                A  994
      IF (IDIR(2).NE.0) WRITE (7,85) IN,IDIR(2),WBTG,WBTA,WBTR,EBTG,EBTA   A  995
     1,EBTR                                                                A  996
      IF (IDIR(3).NE.0) WRITE (7,85) IN,IDIR(3),SBTG,SBTA,SBTR,NBTG,NBTA   A  997
     1,NBTR                                                                A  998
      IF (IDIR(4).NE.0) WRITE (7,85) IN,IDIR(4),EBTG,EBTA,EBTR,WBTG,WBTA   A  999
     1,WBTR                                                                A1000
```

```
      GO TO 1                                                                  A1001
C                                                                              A1002
C     PROBLEMS IN THE SIMPLEX ROUTINE                                          A1003
C                                                                              A1004
67    CONTINUE                                                                 A1005
C                                                                              A1006
C     CHECK IF PROBLEM IS INFEASIBLE                                           A1007
C                                                                              A1008
      IF (ID.EQ.2) GO TO 68                                                    A1009
C                                                                              A1010
C     OTHERWISE THE PROBLEM IS UNBOUNDED                                       A1011
C                                                                              A1012
      WRITE (6,86)                                                             A1013
      GO TO 1                                                                  A1014
C                                                                              A1015
C     PROBLEM WAS INFEASIBLE...TRY REDUCING TRAFFIC FLOW                       A1016
C                                                                              A1017
68    FLOW=FLOW-0.05                                                           A1018
      WRITE (6,87) FLOW                                                        A1019
C                                                                              A1020
C     CHECK IF FLOW TOO LITTLE                                                 A1021
C                                                                              A1022
      IF (FLOW.GT.0.75) GO TO 45                                               A1023
      WRITE (6,88)                                                             A1024
C                                                                              A1025
C     GO BACK AND DO IT ALL OVER FOR ANOTHER INTERSECTION.                     A1026
C                                                                              A1027
      GO TO 1                                                                  A1028
C                                                                              A1029
C                                                                              A1030
69    STOP                                                                     A1031
C                                                                              A1032
C                                                                              A1033
C                                                                              A1034
C                                                                              A1035
C                                                                              A1036
70    FORMAT (1H1,///,1X,'ALLOTMENT OF CYCLE TO PHASE',///)                    A1037
71    FORMAT (1X,I3,I4,I4,3F6.0,3I3,17X,6A4)                                   A1038
72    FORMAT (1X,3I5,3F10.1,3I6,10X,6A4)                                       A1039
73    FORMAT (//)                                                              A1040
74    FORMAT (1X,I3,1X,I1,5(3I3),5X,6A4)                                       A1041
75    FORMAT (1X,I5,I5,1X,5(3I3),17X,6A4)                                      A1042
76    FORMAT (//)                                                              A1043
77    FORMAT (1X,'IN PHASE',I5,'   FOR A MINIMUM OF ',F6.2,'   SECONDS;        A1044
     1  NOW ALLOCATED ',F6.2,'   SECONDS.')                                    A1045
78    FORMAT (1X,'WARNING...PRESENT ALLOTMENT DOES NOT SUM TO CYCLE LEN        A1046
     1GTH.')                                                                  A1047
79    FORMAT (1X,'WARNING...PRESENT ALLOTMENT TO PHASE',I4,'    IS NOT S       A1048
     1UFFICIENT')                                                             A1049
80    FORMAT (1X,'WARNING...PRESENT ALLOTMENT OF CYCLE DOES NOT ALLOW SU       A1050
```

```
     1FFICIENT VEHICLE STARTS OF TYPE ',I5,'   FROM LEG ',I5,'  .')          A1051
81    FORMAT (//,1X,'AFTER OPTIMIMIZATION',///)                              A1052
82    FORMAT (1X,'IN PHASE   ',I5,'   FOR ',F8.2,'   SECONDS.')              A1053
83    FORMAT (///,1X,'MAXIMUM FLOW OF VEHICLES BEFORE OPTIMIZATION =',F8     A1054
     1.2,'   VEHICLES PER CYCLE  ',/,1X,'MAXIMUM FLOW OF VEHICLES  AFTER     A1055
     2 OPTIMIZATION =',F8.2,'   VEHICLES PER CYCLE  ',/,1X,'PERCENTAGE I     A1056
     3NCREASE =',F8.2,///)                                                   A1057
84    FORMAT (1X,'THE FOLLOWING CARDS ARE TO USED IN THE SIMULATION',///     A1058
     1)                                                                      A1059
85    FORMAT (1X,2I3,6F10.4)                                                 A1060
86    FORMAT (1X,'PROBLEM FORMULATED INCORRECTLY')                           A1061
87    FORMAT (1X,'PROBLEM INFEASIBLE AS STATED; TRAFFIC LEVEL REDUCED TO     A1062
     1 ',F8.3)                                                               A1063
88    FORMAT (1X,'PROBLEM INFEASIBLE AS STATED')                             A1064
      END                                                                    A1065-
```

```
      SUBROUTINE SIMPLX                                                 B    1
C                                                                       B    2
CC                                                                      B    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR             B    4
C     VICTORIA, BRITISH COLUMBIA                                        B    5
C     CANADA                                                            B    6
C                                                        NOV. 1974      B    7
C                                                                       B    8
C                                                                       B    9
C     PURPOSE:     TO SOLVE THE DUAL OF A LINEAR PROGRAMMING PROBLEM    B   10
C                  BY USING THE SIMPLEX ROUTINE.                        B   11
C                                                                       B   12
C                                                                       B   13
C     VARIABLES:                                                        B   14
C                                                                       B   15
C                                                                       B   16
C                  (LISTED IN SUBROUTINE SPLIT)                         B   17
C                                                                       B   18
C                                                                       B   19
C     CALLED ROUTINES:                                                  B   20
C                                                                       B   21
C                  GJELIM                                               B   22
C                  MINRAT                                               B   23
C                  SWAP                                                 B   24
C                                                                       B   25
C                                                                       B   26
      DOUBLE PRECISION A(15,35),TOLER,LABEL(30),DABS                    B   27
C                                                                       B   28
CC                                                                      B   29
      COMMON A,TOLER,N,NP1,NP2,NP3,NP4,M,MP1,MP2,MP3,MP4                B   30
      COMMON LABEL                                                      B   31
C                                                                       B   32
CC    SET UP BOTTOM TWO ROWS OF TABLEAU                                 B   33
C                                                                       B   34
      DO 2 J=1,NP1                                                      B   35
      DO 1 I=1,M                                                        B   36
      IF (A(I,NP2).GT.0.0D0) A(MP1,J)=A(MP1,J)+A(I,J)*A(I,NP3)          B   37
      IF (A(I,NP2).LT.0.0D0) A(MP2,J)=A(MP2,J)+A(I,J)                   B   38
1     CONTINUE                                                          B   39
2     CONTINUE                                                          B   40
      DO 3 J=1,NP1                                                      B   41
      IF (A(MP3,J).GT.0.0D0) A(MP1,J)=A(MP1,J)-A(MP4,J)                 B   42
      IF (A(MP3,J).LT.0.0D0) A(MP2,J)=A(MP2,J)-1.0D0                    B   43
3     CONTINUE                                                          B   44
CC                                                                      B   45
CC    DETERMINE NUMBER OF ARTIFICIAL VARIABLES IN BASIS                 B   46
C                                                                       B   47
4     NAVIB=0                                                           B   48
      DO 5 I=1,M                                                        B   49
      IF (A(I,NP2).LT.0.0D0) NAVIB=NAVIB+1                              B   50
```

```
5       CONTINUE                                                          B   51
        IF (NAVIB.EQ.0) GO TO 11                                          B   52
        DO 6 K2=1,N                                                       B   53
        IF (A(MP2,K2).GT.TOLER) GO TO 7                                   B   54
6       CONTINUE                                                          B   55
        IF (A(MP2,NP1).LT.TOLER) GO TO 11                                 B   56
C                                                                         B   57
C       ORIGINAL PROBLEM IS UNBOUNDED                                     B   58
C                                                                         B   59
        A(MP4,NP3)=3.0D0                                                  B   60
        RETURN                                                            B   61
C                                                                         B   62
C       DETERMINE VARIABLE TO ENTER BASIS                                 B   63
C                                                                         B   63
7       DO 8 K2=1,N                                                       B   64
        IF (A(MP2,K2).GT.TOLER.AND.A(MP3,K2).GT.0.0D0) GO TO 9            B   65
8       CONTINUE                                                          B   66
C                                                                         B   67
C       ENTER SECOND PHASE OF SIMPLEX ROUTINE                             B   68
C                                                                         B   69
        GO TO 11                                                          B   70
9       K=K2                                                              B   71
        DO 10 K2=1,N                                                      B   72
        IF (A(MP2,K2).GT.TOLER.AND.A(MP3,K2).GT.0.0D0.AND.A(MP2,K2).GT.A(M B   73
       1P2,K)) K=K2                                                       B   74
10      CONTINUE                                                          B   75
C                                                                         B   76
C       DETERMINE WHICH VARIABLE SHOULD LEAVE THE BASIS                   B   77
C                                                                         B   78
        CALL MINRAT (K,L)                                                 B   79
        IF (A(MP4,NP3).EQ.2.0D0) RETURN                                   B   80
        CALL GJELIM (MP2,NP1,L,K)                                         B   81
        CALL SWAP (L,NP2,MP3,K)                                           B   82
        CALL SWAP (L,NP3,MP4,K)                                           B   83
        GO TO 4                                                           B   84
C                                                                         B   85
C       DETERMINE IF A VARIABLE CAN ENTER THE BASIS                       B   86
C                                                                         B   87
11      DO 12 I=1,N                                                       B   88
        IF (DABS(A(MP2,I)).LT.TOLER.AND.A(MP1,I).GT.TOLER.AND.A(MP3,I).GT B   89
       10.0D0) GO TO 13                                                   B   90
12      CONTINUE                                                          B   91
C                                                                         B   92
C       OPTIMAL SOLUTION FOUND                                            B   93
C                                                                         B   94
        A(MP4,NP3)=1.0D0                                                  B   95
        RETURN                                                            B   96
C                                                                         B   97
C       DETERMINE VARIABLE TO ENTER BASIS                                 B   98
C                                                                         B   99
```

```
13      K1=I                                                                    B 100
        K=K1                                                                    B 101
        DO 14 I=K1,N                                                            B 102
        IF (A(MP1,I).GT.A(MP1,K).AND.DABS(A(MP2,I)).LT.TOLER.AND.A(MP3,I),      B 103
       1GT.0.0D0) K=I                                                           B 104
14      CONTINUE                                                                B 105
C                                                                               B 106
C       DETERMINE VARIABLE TO LEAVE THE BASIS                                   B 107
C                                                                               B 108
        CALL MINRAT (K,L)                                                       B 109
        IF (A(MP4,NP3).EQ.2.0D0) RETURN                                         B 110
        CALL GJELIM (MP2,NP1,L,K)                                               B 111
        CALL SWAP (L,NP2,MP3,K)                                                 B 112
        CALL SWAP (L,NP3,MP4,K)                                                 B 113
        GO TO 11                                                                B 114
        END                                                                     B 115-
```

```
      SUBROUTINE GJELIM (NR,NC,L,K)                                          C    1
C                                                                            C    2
C                                                                            C    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR                  C    4
C     VICTORIA, BRITISH COLUMBIA                                             C    5
C     CANADA                                                                 C    6
C                                                          NOV. 1974         C    7
C                                                                            C    8
C     PURPOSE:     TO PERFORM GAUSS-JORDAN ELIMINATION ON A                  C    9
C                                                                            C   10
C                                                                            C   11
C     VARIABLES:                                                             C   12
C                                                                            C   13
C                                                                            C   14
C                  (LISTED IN SUBROUTINE SPLIT)                              C   15
C                                                                            C   16
C                                                                            C   17
      DOUBLE PRECISION A(15,35),TOLER,LABEL(30),DABS                         C   18
C                                                                            C   19
C                                                                            C   20
      COMMON A,TOLER,N,NP1,NP2,NP3,NP4,M,MP1,MP2,MP3,MP4                     C   21
      COMMON LABEL                                                           C   22
C                                                                            C   23
C                                                                            C   24
      PIVOT=A(L,K)                                                           C   25
      DO 2 I=1,NR                                                            C   26
      IF (I.EQ.L) GO TO 2                                                    C   27
      DO 1 J=1,NC                                                            C   28
      IF (J.EQ.K) GO TO 1                                                    C   29
      IF (DABS(A(L,J)).LT.TOLER.OR.DABS(A(I,K)).LT.TOLER) GO TO 1            C   30
      A(I,J)=A(I,J)-A(L,J)*A(I,K)/PIVOT                                      C   31
1     CONTINUE                                                               C   32
2     CONTINUE                                                               C   33
      DO 3 J=1,NC                                                            C   34
      A(L,J)=A(L,J)/PIVOT                                                    C   35
3     CONTINUE                                                               C   36
      DO 4 I=1,NR                                                            C   37
      A(I,K)=-A(I,K)/PIVOT                                                   C   38
4     CONTINUE                                                               C   39
      A(L,K)=1.0D0/PIVOT                                                     C   40
      RETURN                                                                 C   41
      END                                                                    C   42-
```

```
      SUBROUTINE SWAP (I1,J1,I2,J2)                                     D    1
C                                                                       D    2
C                                                                       D    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR             D    4
C     VICTORIA, BRITISH COLUMBIA                                        D    5
C     CANADA                                                            D    6
C                                                        NOV. 1974      D    7
C                                                                       D    8
C                                                                       D    9
C     PURPOSE:     TO SWAP ONE ELEMENT OF A WITH ANOTHER.               D   10
C                                                                       D   11
C                                                                       D   12
C     VARIABLES:                                                        D   13
C                                                                       D   14
C                                                                       D   15
C                  (LISTED IN SUBROUTINE SPLIT)                         D   16
C                                                                       D   17
C                                                                       D   18
      DOUBLE PRECISION A(15,35),TOLER,LABEL(30),TEMP                    D   19
C                                                                       D   20
C                                                                       D   21
      COMMON A,TOLER,N,NP1,NP2,NP3,NP4,M,MP1,MP2,MP3,MP4                D   22
      COMMON LABEL                                                      D   23
C                                                                       D   24
C                                                                       D   25
      TEMP=A(I1,J1)                                                     D   26
      A(I1,J1)=A(I2,J2)                                                 D   27
      A(I2,J2)=TEMP                                                     D   28
      RETURN                                                            D   29
      END                                                               D   30-
```

```
      SUBROUTINE MINRAT (K,L)                                                E    1
C                                                                            E    2
CC                                                                           E    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR                  E    4
C     VICTORIA, BRITISH COLUMBIA                                             E    5
C     CANADA                                                                 E    6
C                                                        NOV. 1974           E    7
C                                                                            E    8
C                                                                            E    9
C     PURPOSE:     TO DETERMINE WHICH VARIABLE SHOULD LEAVE THE BASIS.       E   10
C                                                                            E   11
C                                                                            E   12
C     VARIABLES:                                                             E   13
C                                                                            E   14
C                                                                            E   15
C                  (LISTED IN SUBROUTINE SPLIT)                              E   16
C                                                                            E   17
C                                                                            E   18
      DOUBLE PRECISION A(15,35),TOLER,LABEL(30)                              E   19
C                                                                            E   20
C                                                                            E   21
      COMMON A,TOLER,N,NP1,NP2,NP3,NP4,M,MP1,MP2,MP3,MP4                     E   22
      COMMON LABEL                                                           E   23
C                                                                            E   24
C                                                                            E   25
      DO 1  I=1,M                                                            E   26
      IF (A(I,K).GT.TOLER) GO TO 2                                           E   27
1     CONTINUE                                                               E   28
C                                                                            E   29
C     NONE ARE POSITIVE, THEREFORE ORIGINAL PROBLEM IS INFEASIBLE.           E   30
C                                                                            E   31
      A(MP4,NP3)=2.0D0                                                       E   32
      RETURN                                                                 E   33
2     L1=I                                                                   E   34
      L=L1                                                                   E   35
      DO 3  I=L1,M                                                           E   36
      IF (A(I,K).LT.TOLER) GO TO 3                                           E   37
      IF (A(I,NP1)/A(I,K).LT.A(L,NP1)/A(L,K)) L=I                            E   38
3     CONTINUE                                                               E   39
      RETURN                                                                 E   40
      END                                                                    E   41-
```

APPENDIX B

DOCUMENTATION OF SUBROUTINES USED IN PHASE II

```
      SUBROUTINE SIMU                                                   A    1
C                                                                       A    2
C                                                                       A    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR             A    4
C     VICTORIA, BRITISH COLUMBIA                                        A    5
C     CANADA                                                            A    6
C                                                    NOV. 1974          A    7
C                                                                       A    8
C     PURPOSE:      TO SIMULATE TRAFFIC ALONG A SINGLE LEG OF A       A    9
C                   ROADWAY BETWEEN TWO INTERSECTIONS TO EVALULATE THE A   10
C                   EFFECTS OF DIFFERENT RELATIVE OFFSETS.            A   11
C                                                                       A   12
C                                                                       A   13
C     METHOD:       THE SUBROUTINE IS A FORTRAN IMPLEMENTATION        A   14
C                   OF THE MODEL OUTLINED IN CHAPTER 5 OF THE         A   15
C                   COVERING WORK.                                    A   16
C                                                                       A   17
C                                                                       A   18
C     VARIABLES (INCLUDING THOSE IN CALLED ROUTINES):                   A   19
C                                                                       A   20
C                                                                       A   21
C     ABD           DIRECTION OF TRAVEL.                              A   22
C                      1: FROM I TO J.                                  A   23
C                      2: FROM J TO I.                                  A   24
C                                                                       A   25
C     ALENLT(K)     (K=1,2)                                           A   26
C                   LENGTH OF LEFT TURN ONLY LANE GOING IN            A   27
C                   DIRECTION K.                                      A   28
C                                                                       A   29
C     ALENRT(K)     (K=1,2)                                           A   30
C                   LENGTH OF RIGHT TURN ONLY LANE GOING IN           A   31
C                   DIRECTION K.                                      A   32
C                                                                       A   33
C     ALENST        LENGTH OF LANES BETWEEN INTERSECTIONS I AND J.     A   34
C                                                                       A   35
C     ALL           LOWER LIMIT ON ACCELERATION.                      A   36
C                                                                       A   37
C     ALV           AVERAGE LENGTH OF VEHICLE.                        A   38
C                                                                       A   39
C     AMI           TIME LIGHT TURNS AMBER.                           A   40
C                                                                       A   41
C     ANXT          ACCELERATION OF VEHICLE AHEAD.                    A   42
C                                                                       A   43
C     APLIT(K)      (K=1,2,3,...,10)                                  A   44
C                   COLOUR OF PRIMARY LIGHT FOR LANE K.               A   45
C                                                                       A   46
C     ASLIT(K)      (K=1,2,3,...,10)                                  A   47
C                   COLOUR OF SECONDARY LIGHT FOR LANE K.             A   48
C                                                                       A   49
C     AT(K)         (K=1,2,3,...,1000)                                A   50
```

```
C                AVERAGE TIME FOR VEHICLE TO TRAVERSE LEG WITH          A  51
C                RELATIVE OFFSET OF (K-1)*2.                            A  52
C                                                                       A  53
C      AUL       UPPER LIMIT ON ACCELERATION.                           A  54
C                                                                       A  55
C      AVE       AVERAGE TIME OF VEHICLE TRAVERSING LEG.                A  56
C                                                                       A  57
C      AW        DESIRED ACCELERATION OF VEHICLE.                       A  58
C                                                                       A  59
C      BLANK     CONTAINS ' ' (USED FOR PLOTTING OF DELAY FUNCTION).    A  60
C                                                                       A  61
C      BLOB      CONTAINS '@' (USED FOR ROUGH PLOTTING OF DELAY         A  62
C                FUNCTION).                                             A  63
C                                                                       A  64
C      BOT       TEMPORARY STORAGE FOR COMPUTATION OF STANDARD          A  65
C                DEVIATION.                                             A  66
C                                                                       A  67
C      BS        BASE SUBSCRIPT (1 DURING ODD ITERATIONS AND 7 DURING   A  68
C                EVEN ITERATIONS).                                      A  69
C                                                                       A  70
C      BSP1      BS + 1                                                 A  71
C                                                                       A  72
C      BSP2      BS + 2                                                 A  73
C                                                                       A  74
C      BSP3      BS + 3                                                 A  75
C                                                                       A  76
C      BSP4      BS + 4                                                 A  77
C                                                                       A  78
C      BSP5      BS + 5                                                 A  79
C                                                                       A  80
C      BSS       INDICATED ODD OR EVEN ITERATION (1 FOR ODD AND         A  81
C                2 FOR EVEN).                                           A  82
C                                                                       A  83
C      CARDOP(I,J)  (I=1,2,3,...,200); (J=1,2,3,...,12)               A  84
C                ITH ROW OF MATRIX REPRESENTS THE STATUS OF THE ITH     A  85
C                VEHICLE.                                               A  86
C                J=1,7  POINTER TO NEXT VEHICLE IN SAME LANE            A  87
C                J=2,8  NUMBER OF LANE                                  A  88
C                J=3,9  LEFT, RIGHT, STRAIGHT FLAG                      A  89
C                J=4,10 VELOCITY                                        A  90
C                J=5,11 ACCELERATION                                    A  91
C                J=6,12 DISTANCE FROM FAR LIGHT                         A  92
C                                                                       A  93
C      COL       COLOUR OF LIGHT FOR LANE IN QUESTION.                  A  94
C                   1: GREEN.                                           A  95
C                   2: AMBER.                                           A  96
C                   3: RED.                                             A  97
C                                                                       A  98
C      CPL(K,L)  (K=1,2,...,10); (L=1,2,3)                             A  99
C                TIME IN CYCLE THAT PRIMARY LIGHT FOR LANE K TURNS      A 100
```

```
C                COLOUR L.                                                  A 101
C                                                                           A 102
C     CSL(K,L)   (K=1,2,...,10); (L=1,2,3)                                  A 103
C                TIME IN CYCLE THAT SECONDARY LIGHT FOR LANE K              A 104
C                TURNS COLOUR L.                                            A 105
C                                                                           A 106
C     DLP        DISTANCE THAT LAST CAR IN LANE IS FROM LIGHT.              A 107
C                                                                           A 108
C     DNS        ESTIMATED DISTANCE NEEDED FOR VEHICLE AHEAD TO STOP.       A 109
C                                                                           A 110
C     DNXT       PRESENT DISTANCE TO VEHICLE AHEAD.                         A 111
C                                                                           A 112
C     DT         DISTANCE TRAVELLED BY VEHICLE IN TIME-SLICE.               A 113
C                                                                           A 114
C     EDN        ESTIMATED DISTANCE TO VEHICLE AHEAD UNDER PRESENT          A 115
C                CONDITIONS.                                                A 116
C                                                                           A 117
C     EPS        TIME INCREMENT.                                            A 118
C                                                                           A 119
C     FA         FUTURE ACCELERATION OF THE VEHICLE.                        A 120
C                                                                           A 121
C     FDL        FUTURE DISTANCE TO LIGHT.                                  A 122
C                                                                           A 123
C     FL         FUTURE LANE OF VEHICLE.                                    A 124
C                                                                           A 125
C     FLRS       FUTURE LEFT-RIGHT-STRAIGHT FLAG.                           A 126
C                                                                           A 127
C     FV         FUTURE VELOCITY OF VEHICLE.                                A 128
C                                                                           A 129
C     GRI        TIME LIGHT TURNS GREEN.                                    A 130
C                                                                           A 131
C     GT         TOTAL GREEN TIME OF LIGHT FOR LANE IN QUESTION.            A 132
C                                                                           A 133
C     I          DO-LOOP PARAMETER.                                         A 134
C                                                                           A 135
C     I1         POINTER TO VEHICLE TO BE REMOVED.                          A 136
C                                                                           A 137
C     I3         DO-LOOP PARAMETER.                                         A 138
C                                                                           A 139
C     I5         DO-LOOP PARAMETER.                                         A 140
C                                                                           A 141
C     IAA        TIME (NEGLECTING OFFSET) LIGHT TURNS AMBER.                A 142
C                                                                           A 143
C     IAG        TIME (NEGLECTING OFFSET) LIGHT TURNS GREEN.                A 144
C                                                                           A 145
C     IAR        TIME (NEGLECTING OFFSET) LIGHT TURNS RED.                  A 146
C                                                                           A 147
C     IAT        PRESENT TIME IN CYCLE IF IT HAD TURNED GREEN               A 148
C                AT OFFSET ZERO.                                            A 149
C                                                                           A 150
```

```
C       IBP             POINTER TO LAST CAR IN LANE.                                    A 151
C                                                                                       A 152
C       IDENT           ALPHANUMERIC IDENTIFICATION OF LEG.                             A 153
C                                                                                       A 154
C       IETA            TIME IN CYCLE WHEN LIGHT I TURNS AMBER FOR                      A 155
C                       TRAFFIC ENTERING LEG.                                           A 156
C                                                                                       A 157
C       IETG            TIME IN CYCLE WHEN LIGHT I TURNS GREEN FOR                      A 158
C                       TRAFFIC ENTERING LEG.                                           A 159
C                                                                                       A 160
C       IETR            TIME IN CYCLE WHEN LIGHT I TURNS RED FOR                        A 161
C                       TRAFFIC ENTERING LEG.                                           A 162
C                                                                                       A 163
C       IIN             NUMBER OF INTERSECTION I.                                       A 164
C                                                                                       A 165
C       IJF             DIRECTION OF TRAVEL:                                            A 166
C                          1: FROM I TO J.                                              A 167
C                          2: FROM J TO I.                                              A 168
C                                                                                       A 169
C       IJK             DIRECTION OF TRAVEL:                                            A 170
C                          1: FROM I TO J.                                              A 171
C                          2: FROM J TO I.                                              A 172
C                                                                                       A 173
C       ILRS            LEFT-RIGHT-STRAIGHT FLAG FOR VEHICLE                            A 174
C                       ENTERING SIMULATION.                                            A 175
C                                                                                       A 176
C       ILTA            TIME IN CYCLE WHEN LIGHT I TURNS AMBER FOR TRAFFIC              A 177
C                       LEAVING LEG.                                                    A 178
C                                                                                       A 179
C       ILTG            TIME IN CYCLE WHEN LIGHT I TURNS GREEN FOR                      A 180
C                       TRAFFIC LEAVING LEG.                                            A 181
C                                                                                       A 182
C       ILTR            TIME IN CYCLE WHEN LIGHT I TURNS RED FOR TRAFFIC                A 183
C                       LEAVING LEG.                                                    A 184
C                                                                                       A 185
C       IPNT            POINTER TO FREE VEHICLE TO BE ADDED TO                          A 186
C                       THE SIMULATION.                                                 A 187
C                                                                                       A 188
C       IRID1           ALPHANUMERIC IDENTIFIER OF LEG.                                 A 189
C                                                                                       A 190
C       IRID2           ALPHANUMERIC IDENTIFIER OF LEG.                                 A 191
C                                                                                       A 192
C       ITIME           TIME IN CYCLE.                                                  A 193
C                                                                                       A 194
C       IX              RANDOM NUMBER GENERATOR SEED.                                   A 195
C                                                                                       A 196
C       IX1             TEMPORARY STORAGE LOCATION.                                     A 197
C                                                                                       A 198
C       IXX             TEMPORARY LOCATION FOR NUMBER OF INTERSECTION I.                A 199
C                                                                                       A 200
```

```
C      IY              USED IN CALCULATING YFL.                                  A 201
C                                                                                A 202
C      J               DO-LOOP PARAMETER.                                        A 203
C                                                                                A 204
C      J1              DO-LOOP PARAMETER.                                        A 205
C                                                                                A 206
C      JETA            TIME IN CYCLE WHEN LIGHT J TURNS AMBER FOR                A 207
C                      TRAFFIC ENTERING LEG.                                     A 208
C                                                                                A 209
C      JETG            TIME IN CYCLE WHEN LIGHT J TURNS GREEN FOR                A 210
C                      TRAFFIC ENTERING LEG.                                     A 211
C                                                                                A 212
C      JETR            TIME IN CYCLE WHEN LIGHT J TURNS RED FOR                  A 213
C                      TRAFFIC ENTERING LEG.                                     A 214
C                                                                                A 215
C      JIN             NUMBER OF INTERSECTION J.                                 A 216
C                                                                                A 217
C      JJ              DO-LOOP PARAMETER (EQUALS JTHETA MINUS 1).                A 218
C                                                                                A 219
C      JJM1            JJ-1                                                      A 220
C                                                                                A 221
C      JLTA            TIME IN CYCLE WHEN LIGHT J TURNS AMBER FOR                A 222
C                      TRAFFIC LEAVING LEG.                                      A 223
C                                                                                A 224
C      JLTG            TIME IN CYCLE WHEN LIGHT J TURNS GREEN FOR                A 225
C                      TRAFFIC LEAVING LEG.                                      A 226
C                                                                                A 227
C      JLTR            TIME IN CYCLE WHEN LIGHT J TURNS RED FOR                  A 228
C                      TRAFFIC LEAVING LEG.                                      A 229
C                                                                                A 230
C      JS              INITIAL OFFSET FOR INTERSECTION J.                        A 231
C                                                                                A 232
C      JTHETA          OFFSET OF INTERSECTION J.                                 A 233
C                                                                                A 234
C      JXX             TEMPORARY LOCATION FOR NUMBER OF INTERSECTION J.          A 235
C                                                                                A 236
C      K               DO-LOOP PARAMETER.                                        A 237
C                                                                                A 238
C      K1 TO K6        TEMPORARY STORAGE LOCATIONS.                              A 239
C                                                                                A 240
C      KK              DO-LOOP PARAMETER.                                        A 241
C                                                                                A 242
C      L               NUMBER OF LANES GOING IN APPROPRIATE DIRECTION.           A 243
C                                                                                A 244
C      NA              NUMBER OF ACCIDENTS THAT OCCURRRED.                       A 245
C                                                                                A 246
C      NALT            NEXT AVAILABLE LOCATION IN ARRAY T.                       A 247
C                                                                                A 248
C      NBS             NEW BASIC SUBSCRIPT.                                      A 249
C                                                                                A 250
```

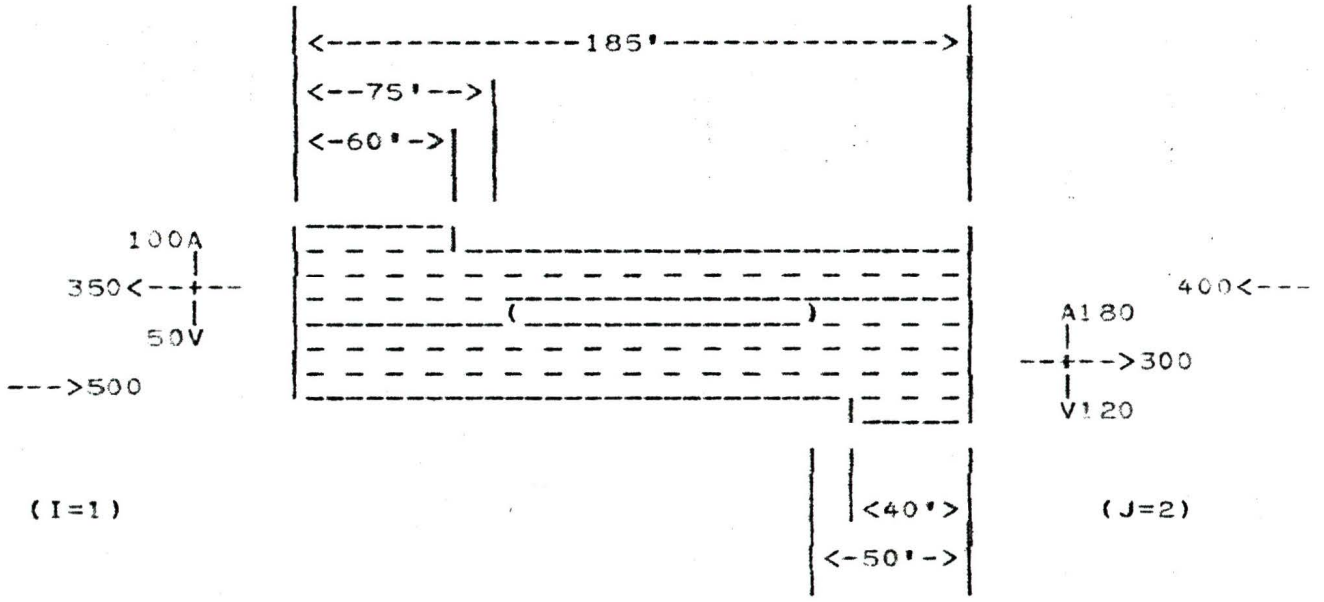
```
C     NBSP1          NBS + 1                                                  A 251
C                                                                             A 252
C     NBSP2          NBS + 2                                                  A 253
C                                                                             A 254
C     NBSP3          NBS + 3                                                  A 255
C                                                                             A 256
C     NBSP4          NBS + 4                                                  A 257
C                                                                             A 258
C     NBSP5          NBS + 5                                                  A 259
C                                                                             A 260
C     NBSS           NEW BSS.                                                 A 261
C                                                                             A 262
C     NC             NUMBER OF VEHICLES.                                      A 263
C                                                                             A 264
C     NLA            NUMBER OF LANES FROM I TO J.                             A 265
C                                                                             A 266
C     NLC            NUMBER OF LANES FORM J TO I.                             A 267
C                                                                             A 268
C     NLL            NUMBER OF LEFT TURN ONLY LANES.                          A 269
C                                                                             A 270
C     NLR            NUMBER OF RIGHT TURN ONLY LANES.                         A 271
C                                                                             A 272
C     NLT            TOTAL NUMBER OF LANES IN LEG.                            A 273
C                                                                             A 274
C     NLTM1          NLT - 1                                                  A 275
C                                                                             A 276
C     NNN            LAST VALUE RETURNED FROM ITIMER.                         A 277
C                                                                             A 278
C     NRL            NUMBER OF RIGHT TURN ONLY LANE                           A 279
C                                                                             A 280
C     NSIC           NUMBER OF SECONDS IN CYCLE.                              A 281
C                                                                             A 282
C     NTC            NUMBER OF VEHICLES TO BE SIMULATED.                      A 283
C                                                                             A 284
C     OPTSPD         AVERAGE MAXIMUM SPEED ALLOWED.                           A 285
C                                                                             A 286
C     PA             PRESENT ACCELERATION.                                    A 287
C                                                                             A 288
C     PARM1          PARAMETER USED IN CALCULATING FUTURE VELOCITY.           A 289
C                                                                             A 290
C     PCOL           PRESENT COLOUR OF NEXT LIGHT.                            A 291
C                                                                             A 292
C     PDE            EXPECTED DISTANCE TO LIGHT IF PRESENT VELOCITY           A 293
C                    WERE TO CONTINUE.                                        A 294
C                                                                             A 295
C     PDL            PRESENT DISTANCE TO PRIMARY LIGHT.                       A 296
C                                                                             A 297
C     PDM            PRESENT DISTANCE TO NEXT LIGHT MINUS A BUFFER            A 298
C                    DISTANCE OF FOUR FEET.                                   A 299
C                                                                             A 300
```

```
C      PFFV          POINTER TO FIRST FREE VEHICLE.                          A 301
C                                                                            A 302
C      PFV(I)        (I=1,2,3,...,100)                                       A 303
C                    POINTER TO NEXT FREE VEHICLE (0 IMPLIES NO              A 304
C                    SUCH VEHICLE).                                          A 305
C                                                                            A 306
C      PIC(K)        (K=1,2,3,...,60)                                        A 307
C                    USED TO PRINT ROUGH PLOT OF DELAY FUNCTION.             A 308
C                                                                            A 309
C      PL            PRESENT LANE OF VEHICLE (0 IMPLIES NOT IN THE           A 310
C                    SIMULATION AT THE PRESENT TIME).                        A 311
C                                                                            A 313
C      PLCIL(I,L)    (I=1,2); (L=1,2,3,...,10)                               A 314
C                    POINTER TO LAST VEHICLE IN LANE L (I INDICATES          A 315
C                    EVEN OR ODD ITERATION).                                 A 316
C                                                                            A 317
C      PLRS          PRESENT LEFT-RIGHT-STRAIGHT FLAG.                       A 318
C                                                                            A 319
C      PNIL          POINTER TO VEHICLE AHEAD IN SAME LANE.                  A 320
C                                                                            A 321
C      PNILL         POINTER TO VEHICLE AHEAD IN LEFT TURN ONLY LANE.        A 322
C                                                                            A 323
C      PNIRL         POINTER TO VEHICLE AHEAD IN RIGHT TURN ONLY LANE.       A 324
C                                                                            A 325
C      PNT           POINTER TO VEHICLE.                                     A 326
C                                                                            A 327
C      PPNT          POINTER TO LAST CAR IN LANE.                            A 328
C                                                                            A 329
C      PROB          PROBABILITY OF VEHICLE ENTERING LANE FROM SIDE          A 330
C                    STREET DURING TIME-SLICE.                               A 331
C                                                                            A 332
C      PROBL         PROBABILITY OF VEHICLE TURNING LEFT.                    A 333
C                                                                            A 334
C      PROBR         PROBABILITY OF VEHICLE TURNING RIGHT.                   A 335
C                                                                            A 336
C      PV            PRESENT VELOCITY.                                       A 337
C                                                                            A 338
C      RDI           TIME LIGHT TURNS RED.                                   A 339
C                                                                            A 340
C      SD            STANDARD DEVIATION OF TIME TAKEN TO TRAVERSE LEG        A 341
C                                                                            A 342
C      SDTL          STARTING DISTANCE TO LIGHT.                             A 343
C                                                                            A 344
C      SUM           TOTAL TIME OF VEHICLES TRAVERSING LEG.                  A 345
C                                                                            A 346
C      SUMS          SUM OF SQUARED TIMES OF VEHICLES TRAVERSING LEG.        A 347
C                                                                            A 348
C      T(I)          (I=1,2,...,1000)                                        A 349
C                    TIME THAT VEHICLE I TOOK TO TRAVERSE LEG.               A 350
C                                                                            A 351
```

```
C     TE(I)          (I=1,2,3...,1000)                                    A 352
C                    TIME THAT VEHICLE I ENTERED THE LEG.                 A 353
C                                                                         A 354
C     TEMP(K)        (K=1,2,3,....,12)                                    A 355
C                    TEMPORARY DATA STORAGE.                              A 356
C                                                                         A 357
C     TIME           SIMULATED CLOCK TIME.                                A 358
C                                                                         A 359
C     TOP            TEMPORARY STORAGE FOR COMPUTATION OF                 A 360
C                    STANDARD DEVIATION.                                  A 361
C                                                                         A 362
C     TTIME          TOTAL TIME.                                          A 363
C                                                                         A 364
C     TVEL(K)        (K=1,2,3,....,10)                                    A 365
C                    TIME VEHICLE LAST ENTERED LANE K.                    A 366
C                                                                         A 367
C     VEL            NUMBER OF VEHICLES ENTERING LANE.                    A 368
C                                                                         A 369
C     VEL1           DESIRED VELOCITY DUE TO NEXT LIGHT.                  A 370
C                                                                         A 371
C     VEL2           DESIRED VELOCITY DUE TO MOVEMENT TO BE MADE          A 372
C                    AT NEXT CORNER.                                      A 373
C                                                                         A 374
C     VEL3           DESIRED VELOCITY DUE TO VEHICLE AHEAD.               A 375
C                                                                         A 376
C     VEL4           DESIRED VELOCITY DUE TO VEHICLE IN LEFT TURN         A 377
C                    ONLY LANE.                                           A 378
C                                                                         A 379
C     VEL5           DESIRED VELOCITY DUE TO VEHICLE IN RIGHT             A 380
C                    TURN ONLY LANE.                                      A 381
C                                                                         A 382
C     VELT           TOTAL NUMBER OF VEHICLES TRAVERSING LEG PER HOUR.    A 383
C                                                                         A 384
C     VES(K)         (K=1,2)                                              A 385
C                    VEHICLES PER HOUR ENTERING LEG STRAIGHT              A 386
C                    TRAVELLING IN DIRECTION K.                           A 387
C                                                                         A 388
C     VGS(K)         (K=1,2)                                              A 389
C                    VEHICLES PER HOUR GOING STRAIGHT (TRAVELLING         A 390
C                    IN DIRECTION K).                                     A 391
C                                                                         A 392
C     VGSX           VEHICLES PER HOUR GOING STRAIGHT.                    A 393
C                                                                         A 394
C     VICL(K)        (K=1,2)                                              A 395
C                    VEHICLES PER HOUR ENTERING A NO TURN LANE            A 396
C                    GOING IN DIRECTION K.                                A 397
C                                                                         A 398
C     VICLX          VEHICLES PER HOUR ENTERING A NO TURN LANE            A 399
C                    IN APPROPRIATE DIRECTION.                            A 400
C                                                                         A 401
```

```
C       VILL(K)         (K=1,2)                                                 A 402
C                       VEHICLES PER HOUR ENTERING A LEFT TURN LANE             A 403
C                       GOING IN DIRECTION K.                                   A 404
C                                                                               A 405
C       VILLX           VEHICLES PER HOUR ENTERING A LEFT TURN LANE             A 406
C                       IN APPROPRIATE DIRECTION.                               A 407
C                                                                               A 408
C       VIRL(K)         (K=1,2)                                                 A 409
C                       VEHICLES PER HOUR ENTERING A RIGHT TURN LANE            A 410
C                       GOING IN DIRECTION K.                                   A 411
C                                                                               A 412
C       VIRLX           VEHICLES PER HOUR ENTERING A RIGHT TURN LANE            A 413
C                       IN APPROPRIATE DIRECTION.                               A 414
C                                                                               A 415
C       VNXT            PRESENT VELOCITY OF VEHICLE AHEAD.                      A 416
C                                                                               A 417
C       VT              VEHICLES PER HOUR TRAVERSING LEG IN APPROPRIATE         A 418
C                       DIRECTION.                                              A 419
C                                                                               A 420
C       VTL(K)          (K=1,2)                                                 A 421
C                       VEHICLES PER HOUR TURNING LEFT OFF LEG                  A 422
C                       (TRAVELLING IN DIRECTION K).                            A 423
C                                                                               A 424
C       VTLX            VEHICLES PER HOUR TURNING LEFT.                         A 425
C                                                                               A 426
C       VTOT(K)         (K=1,2)                                                 A 427
C                       NUMBER OF VEHICLES LEAVING LEG TRAVELLING IN            A 428
C                       DIRECTION K.                                            A 429
C                                                                               A 430
C       VTR(K)          (K=1,2)                                                 A 431
C                       VEHICLES PER HOUR TURNING RIGHT                         A 432
C                       OFF LEG (TRAVELLING IN DIRECTION K)                     A 433
C                                                                               A 434
C       VTRX            VEHICLES PER HOUR TURNING RIGHT.                        A 435
C                                                                               A 436
C       VW              DESIRED FUTURE VELOCITY OF VEHICLE.                     A 437
C                                                                               A 438
C       W               TEMPORARY STORAGE LOCATION.                             A 439
C                                                                               A 440
C       X               TEMPORARY STORAGE LOCATION.                             A 441
C                                                                               A 442
C       YFL             RANDOM NUMBER BETWEEN 0 AND 1.                          A 443
C                                                                               A 444
C       Z               AVERAGE TIME BETWEEN VEHICLES ENTERING LANE.            A 445
C                                                                               A 446
C                                                                               A 447
C       CALLING:        WHEN CALLING THE SUBROUTINE THE FOLLOWING               A 448
C                       VARIABLES MUST BE DEFINED.                              A 449
C                                                                               A 450
C                       ALENLT(K)    (K=1,2)                                    A 451
```

```
C                 ALENRT(K)    (K=1,2)                                    A 452
C                 ALENST                                                  A 453
C                 IETA                                                    A 454
C                 IETG                                                    A 455
C                 IETR                                                    A 456
C                 IIN                                                     A 457
C                 ILTA                                                    A 458
C                 ILTG                                                    A 459
C                 ILTR                                                    A 460
C                 JETA                                                    A 461
C                 JETG                                                    A 462
C                 JIN                                                     A 463
C                 JLTA                                                    A 464
C                 JLTG                                                    A 465
C                 JLTR                                                    A 466
C                 NLA                                                     A 467
C                 NLC                                                     A 468
C                 NSIC                                                    A 469
C                 NTL                                                     A 470
C                 NTC                                                     A 471
C                 VES(K)    (K=1,2)                                       A 472
C                 VGS(K)    (K=1,2)                                       A 473
C                 VTL(K)    (K=1,2)                                       A 474
C                 VTR(K)    (K=1,2)                                       A 475
C                 VTR(K)    (K=1,2)                                       A 476
C                                                                         A 477
C                                                                         A 478
C     FOR EXAMPLE:                                                        A 479
C                                                                         A 480
C                 SUPPOSE WE ARE TO SIMULATE TRAFFIC ALONG THE LEG        A 481
C                 BETWEEN INTERSECTIONS NUMBERED 1 AND 2                  A 482
C                 WITH THE PHYSICAL LAYOUT AND TRAFFIC FLOW AS SHOWN      A 483
C                 BELOW.                                                  A 484
```



FURTHER SUPPOSE THAT THE LIGHTS HAVE A CYCLE LENGTH OF 60 SECONDS, 1000 VEHICLES ARE TO BE TIMED FOR EACH DIFFERENT RELATIVE OFFSET, AND THE LIGHT PATTERNS MAY BE SUMMARIZED AS BELOW:

INTERSECTION #1:

0-7 SECONDS: ADVANCE GREEN TO EASTBOUND
7-30 SECONDS: GREEN TO EAST & WESTBOUND
30-35 SECONDS: AMBER TO EAST & WESTBOUND
35-54 SECONDS: GREEN TO NORTH & SOUTHBOUND
54-0 SECONDS: AMBER TO NORTH & SOUTHBOUND

INTERSECTION #2:

0-6 SECONDS: AMBER TO NORTH & SOUTHBOUND
6-16 SECONDS: ADVANCE GREEN TO EASTBOUND
16-36 SECONDS: GREEN TO EAST & WESTBOUND

```
C     36-42 SECONDS: AMBER TO EAST & WESTBOUND                                  A 535
C     42-0  SECONDS: AMBER TO NORTH & SOUTHBOUND                                A 536
C                                                                               A 537
C                                                                               A 538
C                    UPON ENTERING THE ROUTINE THE VARIABLES WOULD              A 539
C                    HAVE THE FOLLOWING VALUES:                                 A 540
C                                                                               A 541
C                                                                               A 542
C                    ALENLT(K)   (K=1,2)          50, 75                       A 543
C                    ALENRT(K)    (K=1,2)         40, 60                       A 544
C                    ALENST                       185                          A 545
C                    IETA                         30                           A 546
C                    IETG                         0                            A 547
C                    IETR                         35                           A 548
C                    IIN                          1                            A 549
C                    ILTA                         30                           A 550
C                    ILTG                         7                            A 551
C                    ILTR                         35                           A 552
C                    JETA                         36                           A 553
C                    JETG                         36                           A 554
C                    JETG                         16                           A 555
C                    JETR                         42                           A 556
C                    JIN                          2                            A 557
C                    JLTA                         36                           A 558
C                    JLTG                         6                            A 559
C                    JLTR                         42                           A 560
C                    NLA                          5                            A 561
C                    NLC                          4                            A 562
C                    NSIC                         60                           A 563
C                    NTC                          1000                         A 564
C                    VES(K)    (K=1,2)            500, 400                     A 565
C                    VGS(K)    (K=1,2)            300, 350                     A 566
C                    VTL(K)    (K=1,2)            180, 50                      A 567
C                    VTR(K)    (K=1,2)            120, 100                     A 568
C                                                                               A 569
C                                                                               A 570
C     OUTPUT:        FOR EACH DIFFERENT RELATIVE OFFSET THE                     A 571
C                    NUMBERS OF THE INTERSECTIONS AT THE ENDS ARE               A 572
C                    PRINTED ALONG WITH THE DIFFERENCE IN OFFSETS,              A 573
C                    THE AVERAGE TIME TAKEN TO TRAVERSE THE LEG WITH            A 574
C                    THIS RELATIVE OFFSET, AND THE STANDARD DEVIATION           A 575
C                    OF THE TIMES TAKEN TO TRAVERSE THE LEG.                    A 576
C                    UPON RETURN FROM THE ROUTINE ALL                           A 577
C                    INPUT VARIABLES ARE LEFT UNCHANGED AND THE                 A 578
C                    AVERAGE TIMES TAKEN TO TRAVERSE THE LEG ARE                A 579
C                    RETURNED IN THE ARRAY AT. A ROUGH PLOT OF THE              A 580
C                    AVERAGE TIME TAKEN TO TRAVERSE THE LEG IS ALSO             A 581
C                    PRODUCED.                                                  A 582
C                                                                               A 583
C     ROUTINES CALLED:                                                          A 584
```

```
C                                                                       A 585
CC                                                                      A 586
CCC                  INITR                                              A 587
CCC                  ITER                                               A 588
CC                                                                      A 589
C                                                                       A 590
      COMMON NA,IX,JTHETA,T,NTC,ALENLT,ALENRT,ALENST,ALL,ALV,AUL,BS     A 591
      COMMON CARDOP,CPL,CSL,EPS,IIN,JIN,NC,NLA,NLT,NSIC,OPTSPD          A 592
      COMMON PARM1,PFFV,SDTL,TE,TIME,TTIME,VGS,VTL,VTR,VES,IRID1,IRID2  A 593
      COMMON PLCIL,IDENT,VELT,IETG,IETA,IETR,ILTG,ILTA,ILTR,JETG,JETA   A 594
      COMMON JETR,JLTG,JLTA,JLTR                                        A 595
C                                                                       A 596
C                                                                       A 597
      DIMENSION VGS(2), VTR(2), VTL(2), VES(2), T(1000)                 A 598
      INTEGER BLANK/' '/,BLOB/'@'/,PIC(100),PLCIL(2,10),VELT,BS,PFFV    A 599
      REAL ALENLT(2),ALENRT(2),CARDOP(200,12),CPL(10,3),CSL(10,3)       A 600
      REAL TE(1000),AT(1000)                                            A 601
C                                                                       A 602
CC    INITIALIZATION SECTION                                            A 603
C                                                                       A 604
      ALL=-11.0                                                         A 605
      ALV=22.0                                                          A 606
      AUL=5.0                                                           A 607
      EPS=0.75                                                          A 608
      NC=80                                                             A 609
      OPTSPD=44.0                                                       A 610
      PARM1=350.0                                                       A 611
C                                                                       A 612
CC    CALCULATE NUMBER OF LEFT AND RIGHT TURN LANES                     A 613
C                                                                       A 614
      NLA=NLA+2                                                         A 615
      IF (ALENLT(1).GT.5.0) NLA=NLA-1                                   A 616
      IF (ALENRT(1).GT.5.0) NLA=NLA-1                                   A 617
      NLC=NLC+2                                                         A 618
      IF (ALENLT(2).GT.5.0) NLC=NLC-1                                   A 619
      IF (ALENRT(2).GT.5.0) NLC=NLC-1                                   A 620
      NLT=NLA+NLC                                                       A 621
C                                                                       A 622
CC    IF LEFT OR RIGHT TURN ONLY LANE DOESN'T EXIST CREATE A SHORT ONE  A 623
C                                                                       A 624
      DO 1 I=1,2                                                        A 625
      IF (ALENLT(I).LT.5.0) ALENLT(I)=5.0                               A 626
      IF (ALENRT(I).LT.5.0) ALENRT(I)=5.0                               A 627
1     CONTINUE                                                          A 628
C                                                                       A 629
CC    STEP OFFSET OF JTH SIGNAL FROM 0 TO CYCLE LENGTH                  A 630
C                                                                       A 631
      DO 4 JJ=1,NSIC                                                    A 632
C                                                                       A 633
CC    INITIALIZE TIME TAKEN TO TRAVERSE LEG FOR EACH VEHICLE            A 634
```

```
C                                                                       A 635
      DO 2 I=1,1000                                                     A 636
      T(I)=0                                                            A 637
2     CONTINUE                                                          A 638
C                                                                       A 639
C     CALCULATE OFFSET                                                  A 640
C                                                                       A 641
      JTHETA=JJ-1.0                                                     A 642
C                                                                       A 643
C     INITIALIZE FOR A SIMULATION RUN WITH THIS OFFSET                  A 644
C                                                                       A 645
      CALL INITR                                                        A 646
C                                                                       A 647
C     SIMULATE USING THIS OFFSET                                        A 648
C                                                                       A 649
      CALL ITER                                                         A 650
C                                                                       A 651
C     CALCULATE AVERAGE TIME TO TRAVERSE LEG                            A 652
C                                                                       A 653
      SUM=0.0                                                           A 654
      SUMS=0.0                                                          A 655
      DO 3 I3=1,NTC                                                     A 656
      SUMS=SUMS+T(I3)*T(I3)                                             A 657
      SUM=SUM+T(I3)                                                     A 658
3     CONTINUE                                                          A 659
      AVE=SUM/NTC                                                       A 660
      AT(JJ)=AVE                                                        A 661
      TOP=NTC*SUMS-SUM*SUM                                              A 662
      BOT=NTC*(NTC-1)                                                   A 663
      SD=SQRT(TOP/BOT)                                                  A 664
      WRITE (6,7) IIN,JIN,JTHETA,AVE,SD,VELT,IDENT                      A 665
C                                                                       A 666
C     END OF SIMULATION WITH THIS OFFSET                                A 667
C                                                                       A 668
4     CONTINUE                                                          A 669
C                                                                       A 670
C     DO ROUGH PLOT OF DELAY FUNCTION                                   A 671
C                                                                       A 672
      WRITE (6,8)                                                       A 673
      DO 6 JJ=1,NSIC                                                    A 674
      DO 5 K=1,100                                                      A 675
      PIC(K)=BLANK                                                      A 676
      IF (K.LT.AT(JJ)) PIC(K)=BLOB                                      A 677
5     CONTINUE                                                          A 678
      JJM1=JJ-1                                                         A 679
      WRITE (6,9) JJM1,PIC                                              A 680
6     CONTINUE                                                          A 681
      RETURN                                                            A 682
C                                                                       A 683
C                                                                       A 684
```

```
C                                                                       A 685
7       FORMAT (3I4,2F8.2,I6,34X,2(A4,2X))                              A 686
8       FORMAT (1H1)                                                    A 687
9       FORMAT (1X,I3,2X,100A1)                                         A 688
        END                                                             A 689-
```

```
      SUBROUTINE INITR                                                        B    1
C                                                                             B    2
CC                                                                            B    3
CC    THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR                   B    4
CC    VICTORIA, BRITISH COLUMBIA                                              B    5
CC    CANADA                                                                  B    6
CC                                                      NOV. 1974             B    7
CC                                                                            B    8
C                                                                             B    9
      COMMON NA,IX,JTHETA,T,NTC,ALENLT,ALENRT,ALENST,ALL,ALV,AUL,BS           B   10
      COMMON CARDOP,CPL,CSL,EPS,IIN,JIN,NC,NLA,NLT,NSIC,OPTSPD                B   11
      COMMON PARM1,PFFV,SDTL,TE,TIME,TTIME,VGS,VTL,VTR,VES,IRID1,IRID2        B   12
      COMMON PLCIL,IDENT,VELT,IETG,IETA,IETR,ILTG,ILTA,ILTR,JETG,JETA         B   13
      COMMON JETR,JLTG,JLTA,JLTR                                              B   14
C                                                                             B   15
C                                                                             B   16
      DIMENSION VGS(2), VTR(2), VTL(2), VES(2), T(1000)                       B   17
      INTEGER BLANK/' '/,BLOB/'@'/,PIC(100),PLCIL(2,10),VELT,BS,PFFV          B   18
      REAL ALENLT(2),ALENRT(2),CARDOP(200,12),CPL(10,3),CSL(10,3)             B   19
      REAL TE(1000),AT(1000)                                                  B   20
C                                                                             B   21
CC    INITIALIZATION SECTION                                                  B   22
C                                                                             B   23
      BS=7                                                                    B   24
      IX=967532465                                                            B   25
      NA=0                                                                    B   26
      SDTL=ALENST+400.0                                                       B   27
      TIME=0.0                                                                B   28
C                                                                             B   29
CC    CLEAR ALL LANES                                                         B   30
C                                                                             B   31
      DO 1  I=1,10                                                            B   32
      PLCIL(1,I)=0                                                            B   33
      PLCIL(2,I)=0                                                            B   34
1     CONTINUE                                                                B   35
CC                                                                            B   36
CC    CALCULATE TIME IN CYCLE THAT LIGHTS CHANGE COLOURS                      B   37
C                                                                             B   38
      DO 4 I=1,NLT                                                            B   39
C                                                                             B   40
CC    TEST WHICH DIRECTION LANE FLOWS                                         B   41
C                                                                             B   42
      IF (I.GT.NLA) GO TO 2                                                   B   43
C                                                                             B   44
CC    FOR LANES GOING FROM I TO J                                             B   45
C                                                                             B   46
      K1=IETG                                                                 B   47
      K2=IETA                                                                 B   48
      K3=IETR                                                                 B   49
      K4=JLTG+JTHETA                                                          B   50
```

```
      K5=JLTA+JTHETA                                                    B   51
      K6=JLTR+JTHETA                                                    B   52
      GO TO 3                                                           B   53
C                                                                       B   54
C     FOR LANES GOING FROM J TO I                                       B   55
C                                                                       B   56
2     K1=JETG+JTHETA                                                    B   57
      K2=JETA+JTHETA                                                    B   58
      K3=JETR+JTHETA                                                    B   59
      K4=ILTG                                                           B   60
      K5=ILTA                                                           B   61
      K6=ILTR                                                           B   62
C                                                                       B   63
C     RECORD WHEN LIGHTS CHANGE COLOURS                                 B   64
C                                                                       B   65
3     CONTINUE                                                          B   66
      CSL(I,1)=MOD(K1,NSIC)                                             B   67
      CSL(I,2)=MOD(K2,NSIC)                                             B   68
      CSL(I,3)=MOD(K3,NSIC)                                             B   69
      CPL(I,1)=MOD(K4,NSIC)                                             B   70
      CPL(I,2)=MOD(K5,NSIC)                                             B   71
      CPL(I,3)=MOD(K6,NSIC)                                             B   72
4     CONTINUE                                                          B   73
C                                                                       B   74
C     CLEAR ALL VEHICLES FROM SYSTEM                                    B   75
C                                                                       B   76
      DO 6 I=1,NC                                                       B   77
      TE(I)=-1.0                                                        B   78
      DO 5 J=1,12                                                       B   79
      CARDOP(I,J)=0.0                                                   B   80
5     CONTINUE                                                          B   81
6     CONTINUE                                                          B   82
C                                                                       B   83
C                                                                       B   84
      RETURN                                                            B   85
C                                                                       B   86
C                                                                       B   87
      END                                                               B   88-
```

```
      SUBROUTINE ITER                                                    C   1
C                                                                        C   2
CCC                                                                      C   3
CCC   THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR              C   4
CCC   VICTORIA, BRITISH COLUMBIA                                         C   5
CCC   CANADA                                                             C   6
CCC                                                   NOV. 1974          C   7
C                                                                        C   8
                                                                         C   9
      COMMON NA,IX,JTHETA,T,NTC,ALENLT,ALENRT,ALENST,ALL,ALV,AUL,BS      C  10
      COMMON CARDOP,CPL,CSL,EPS,IIN,JIN,NC,NLA,NLT,NSIC,OPTSPD           C  11
      COMMON PARM1,PFFV,SDTL,TE,TIME,TTIME,VGS,VTL,VTR,VES,IRID1,IRID2   C  12
      COMMON PLCIL,IDENT,VELT,IETG,IETA,IETR,ILTG,ILTA,ILTR,JETG,JETA    C  13
      COMMON JETR,JLTG,JLTA,JLTR                                         C  14
      DIMENSION VGS(2), VTR(2), VTL(2), VES(2), T(1000), TVEL(10)        C  15
      DIMENSION VTOT(2), VILL(2), VIRL(2), VICL(2)                       C  16
      INTEGER BLANK/' '/,BLOB/'@'/,PIC(100),PLCIL(2,10),VELT,BS,PFFV     C  17
      INTEGER ABD,BSP1,BSP2,BSP3,BSP4,BSP5,BSS,COL,FL,FLRS,PCOL,PL,PLRS  C  18
      INTEGER PNIL,PNILL,PNIRL,PNT,PPNT,APLIT(10),ASLIT(10),PFV(80)      C  19
      REAL ALENLT(2),ALENRT(2),CARDOP(200,12),CPL(10,3),CSL(10,3)        C  20
      REAL TE(1000),AT(1000)                                             C  21
C                                                                        C  22
CC    INITIALIZE TIME VEHICLE LAST ENTERED LANE                          C  23
C                                                                        C  24
      DO 1 I=1,10                                                        C  25
      TVEL(I)=0.0                                                        C  26
1     CONTINUE                                                           C  27
C                                                                        C  28
CCC   INITIALIZE POINTER TO NEXT AVAILABLE LOCATION IN T                 C  29
C                                                                        C  30
      NALT=1                                                             C  31
C                                                                        C  32
CCC   SET UP POINTERS TO AVAILABLE VEHICLES                              C  33
C                                                                        C  34
      DO 2 I=1,NC                                                        C  35
      PFV(I)=I+1                                                         C  36
2     CONTINUE                                                           C  37
      PFV(NC)=0                                                          C  38
      PFFV=1                                                             C  39
C                                                                        C  40
CCC   CALCULATE NUMBER OF VEHICLES USING THE LANES                       C  41
C                                                                        C  42
      DO 7 IJK=1,2                                                       C  43
      VTOT(IJK)=VGS(IJK)+VTL(IJK)+VTR(IJK)                               C  44
      VILLX=0.0                                                          C  45
      VIRLX=0.0                                                          C  46
      VICLX=0.0                                                          C  47
C                                                                        C  48
CC    DETERMINE NUMBER OF LANES IN THIS DIRECTION                        C  49
C                                                                        C  50
```

```
      L=NLA                                                                  C   51
      IF (IJK.EQ.2) L=NLT-NLA                                                C   52
C                                                                            C   53
CC    DETERMINE VEHICLES USING LEFT LANE, RIGHT LANE, ANY CENTRAL LANE       C   54
C                                                                            C   55
      VTLX=VTL(IJK)                                                          C   56
      VTRX=VTR(IJK)                                                          C   57
      VGSX=VGS(IJK)                                                          C   58
      VT=VTLX+VGSX+VTRX                                                      C   59
      IF (L.GT.2) GO TO 3                                                    C   60
C                                                                            C   61
CC    FOR TWO LANES                                                          C   62
C                                                                            C   63
      W=VT/2                                                                 C   64
      VILLX=W                                                                C   65
      VIRLX=W                                                                C   66
      VICLX=W                                                                C   67
      GO TO 6                                                                C   68
C                                                                            C   69
CC                                                                           C   70
3     W=VT/(L-2)                                                             C   71
      VILLX=W                                                                C   72
      IF (W.LT.VTLX) VILLX=VTLX                                              C   73
      IF (L.GT.3) GO TO 4                                                    C   74
C                                                                            C   75
CC    FOR THREE LANES                                                        C   76
C                                                                            C   77
      VIRLX=W                                                                C   78
      VICLX=W                                                                C   79
      GO TO 6                                                                C   80
C                                                                            C   81
CC                                                                           C   82
4     W=(VT-VILLX)/(L-3)                                                     C   83
      VIRLX=W                                                                C   84
      IF (W.LT.VTRX) VIRLX=VTRX                                              C   85
      IF (L.GT.4) GO TO 5                                                    C   86
C                                                                            C   87
CC    FOR FOUR LANES                                                         C   88
C                                                                            C   89
      VICLX=W                                                                C   90
      GO TO 6                                                                C   91
C                                                                            C   92
CC    FOR FIVE OR MORE LANES                                                 C   93
CC                                                                           C   94
5     W=(VT-VILLX-VIRLX)/(L-4)                                               C   95
      VICLX=W                                                                C   96
C                                                                            C   97
CC                                                                           C   98
6     CONTINUE                                                               C   99
      VILL(IJK)=VILLX                                                        C  100
```

```
      VIRL(IJK)=VIRLX                                                   C 101
      VICL(IJK)=VICLX                                                   C 102
7     CONTINUE                                                          C 103
C                                                                       C 104
CCC   BEGINNING OF A NEW TIME-SLICE                                     C 105
C                                                                       C 106
8     TIME=TIME+EPS                                                     C 107
      ITIME=TIME-IFIX(TIME/NSIC)*NSIC                                   C 108
C                                                                       C 109
CC    CALCULATE COLOURS OF LIGHTS FOR EACH LANE                         C 110
CCC                                                                     C 111
C                                                                       C 112
CC    FOR FIRST INTERSECTION FOR THE LANE                               C 113
C                                                                       C 114
      DO 9 I=1,NLT                                                      C 115
      GRI=CPL(I,1)                                                      C 116
      AMI=CPL(I,2)                                                      C 117
      RDI=CPL(I,3)                                                      C 118
      IX1=AMI-GRI                                                       C 119
      IX1=IX1+60                                                        C 120
      IAA=MOD(IX1,NSIC)                                                 C 121
      IX1=RDI-GRI                                                       C 122
      IX1=IX1+60                                                        C 123
      IAR=MOD(IX1,NSIC)                                                 C 124
      IX1=ITIME-GRI                                                     C 125
      IX1=IX1+60                                                        C 126
      IAT=MOD(IX1,NSIC)                                                 C 127
      COL=3                                                             C 128
      IF (IAT.LT.IAA) COL=1                                             C 129
      IF (IAT.GE.IAA.AND.IAT.LT.IAR) COL=2                              C 130
      APLIT(I)=COL                                                      C 131
9     CONTINUE                                                          C 132
C                                                                       C 133
CC    FOR SECOND INTERSECTION FOR THE LANE                              C 134
C                                                                       C 135
      DO 10 I=1,NLT                                                     C 136
      GRI=CSL(I,1)                                                      C 137
      AMI=CSL(I,2)                                                      C 138
      RDI=CSL(I,3)                                                      C 139
      IX1=AMI-GRI+60                                                    C 140
      IAA=MOD(IX1,NSIC)                                                 C 141
      IX1=RDI-GRI+60                                                    C 142
      IAR=MOD(IX1,NSIC)                                                 C 143
      IX1=ITIME-GRI+60                                                  C 144
      IAT=MOD(IX1,NSIC)                                                 C 145
      COL=3                                                             C 146
      IF (IAT.LT.IAA) COL=1                                             C 147
      IF (IAT.GE.IAA.AND.IAT.LT.IAR) COL=2                              C 148
      ASLIT(I)=COL                                                      C 149
1     CONTINUE                                                          C 150
```

```
C                                                                       C 151
CC      DETERMINE WHICH HALF OF MATRIX CARDOP TO USE                    C 152
C                                                                       C 153
        IF (BS.EQ.1) GO TO 11                                           C 154
        BS=1                                                            C 155
        BSS=1                                                           C 156
        NBS=7                                                           C 157
        NBSS=2                                                          C 158
        GO TO 12                                                        C 159
C                                                                       C 160
CC                                                                      C 161
11      BS=7                                                            C 162
        BSS=2                                                           C 163
        NBS=1                                                           C 164
        NBSS=1                                                          C 165
C                                                                       C 166
CC      SET UP SUBSCRIPTS FOR LATER USE                                 C 167
C                                                                       C 168
12      BSP1=BS+1                                                       C 169
        BSP2=BS+2                                                       C 170
        BSP3=BS+3                                                       C 171
        BSP4=BS+4                                                       C 172
        BSP5=BS+5                                                       C 173
        NBSP1=NBS+1                                                     C 174
        NBSP2=NBS+2                                                     C 175
        NBSP3=NBS+3                                                     C 176
        NBSP4=NBS+4                                                     C 177
        NBSP5=NBS+5                                                     C 178
C                                                                       C 179
CC      CLEAR POINTER TO LAST VEHICLE IN LANE                           C 180
C                                                                       C 181
        DO 13 I=1,NLT                                                   C 182
        PLCIL(NBSS,I)=0                                                 C 183
13      CONTINUE                                                        C 184
C                                                                       C 185
CC      SIMULATE MOVEMENT FOR EACH VEHICLE                              C 186
C                                                                       C 187
        DO 36 I=1,NC                                                    C 188
C                                                                       C 189
CC      DETERMINE PRESENT STATUS OF VEHICLE                             C 190
C                                                                       C 191
        PNIL=CARDOP(I,BS)                                               C 192
        PL=CARDOP(I,BSP1)                                               C 193
C                                                                       C 194
CC      CHECK IF VEHICLE ISN'T IN THE SIMULATION AS YET                 C 195
C                                                                       C 196
        IF (PL.EQ.0) GO TO 36                                           C 197
C                                                                       C 198
CC                                                                      C 199
        ABD=1                                                           C 200
```

```
      IF (PL.GT.NLA) ABD=2                                                    C 201
      PLRS=CARDOP(I,BSP2)                                                     C 202
      PV=CARDOP(I,BSP3)                                                       C 203
      PA=CARDOP(I,BSP4)                                                       C 204
      PDL=CARDOP(I,BSP5)                                                      C 205
C                                                                             C 206
C     CALCULATE VELOCITY REQUIRED DUE TO COLOUR OF LIGHT                      C 207
C                                                                             C 208
      VEL1=OPTSPD                                                             C 209
C                                                                             C 210
C     CHECK IF MAKING RIGHT HAND TURN                                         C 211
C                                                                             C 212
      IF (PL.EQ.NLA.OR.PL.EQ.NLT) GO TO 16                                    C 213
C                                                                             C 214
C                                                                             C 215
      PDE=PDL-PV*EPS-4.0                                                      C 216
      IF (PDE.LT.0.0) PDE=0.0                                                 C 217
      PCOL=ASLIT(PL)                                                          C 218
C                                                                             C 219
C     CHECK IF APPROACHING FIRST LIGHT FOR LANE                               C 220
C                                                                             C 221
      IF (PDL.GT.ALENST) GO TO 15                                             C 222
C                                                                             C 223
C     DETERMINE LANE NUMBERS OF LEFT AND RIGHT TURN LANES                     C 224
C                                                                             C 225
      NLL=1                                                                   C 226
      NLR=NLA                                                                 C 227
      IF (ABD.EQ.1) GO TO 14                                                  C 228
      NLL=NLA+1                                                               C 229
      NLR=NLT                                                                 C 230
C                                                                             C 231
C     DETERMINE COLOUR OF LIGHT                                               C 232
C                                                                             C 233
14    PCOL=APLIT(NLL)                                                         C 234
      IF (PLRS.EQ.2) PCOL=APLIT(NLR)                                          C 235
      IF (PLRS.EQ.3) PCOL=APLIT(PL)                                           C 236
C                                                                             C 237
C     IF LIGHT IS GREEN NO NEED FOR SPEED REDUCTION                           C 238
C                                                                             C 239
15    IF (PCOL.EQ.1) GO TO 16                                                 C 240
C                                                                             C 241
C                                                                             C 242
      PDM=PDL-4.0                                                             C 243
      IF (PDL.GT.ALENST) PDM=PDL-ALENST-4.0                                   C 244
      IF (PDM.LT.0.0) PDM=0.0                                                 C 245
      IF (PDL.GT.ALENST) PDE=PDE-ALENST                                       C 246
C                                                                             C 247
C     IF VEHICLE IS CLOSE ENOUGH TO LIGHT REDUCE SPEED                        C 248
C                                                                             C 249
      IF (PDE.LE.PARM1) VEL1=OPTSPD*SQRT(PDM/PARM1)                           C 250
```

```
C                                                                       C 251
C                                                                       C 252
16    CONTINUE                                                          C 253
      IF (VEL1.LT.0.0) VEL1=0.0                                         C 254
C                                                                       C 255
C     CALCULATE VELOCITY REQUIRED DUE TO MOVEMENT TO BE MADE            C 256
C                                                                       C 257
      VEL2=OPTSPD                                                       C 258
C                                                                       C 259
C     IF AT FIRST LIGHT OR GOING STRAIGHT NO REDUCTION NECESSARY        C 260
C                                                                       C 261
      IF (PLRS.EQ.3.OR.PDL.GT.ALENST) GO TO 17                          C 262
C                                                                       C 263
C                                                                       C 264
      PDE=PDL-PV*EPS-4.0                                                C 265
      PDM=PDL                                                           C 266
C                                                                       C 267
C     IF DISTANCE WARRENTS IT REDUCE SPEED                              C 268
C                                                                       C 269
      IF (PDE.LE.PARM1) VEL2=OPTSPD*SQRT(PDM/PARM1)                     C 270
17    CONTINUE                                                          C 271
C                                                                       C 272
C     INSURE VEHICLE STILL "CREEPS" AHEAD                               C 273
C                                                                       C 274
      IF (VEL2.LT.1.0) VEL2=1.0                                         C 275
C                                                                       C 276
C     CALCULATE VELOCITY REQUIRED DUE TO VEHICLE DIRECTLY AHEAD         C 277
C                                                                       C 278
      VEL3=OPTSPD                                                       C 279
C                                                                       C 280
C     CHECK IF NO VEHICLE IN LEFT TURN LANE                             C 281
C                                                                       C 282
      IF (PNIL.EQ.0) GO TO 18                                           C 283
C                                                                       C 284
C     DETERMINE CHARACTERISTICS OF VEHICLE AHEAD                        C 285
C                                                                       C 286
      VNXT=CARDOP(PNIL,BSP3)                                            C 287
      ANXT=CARDOP(PNIL,BSP4)                                            C 288
      DNXT=PDL-CARDOP(PNIL,BSP5)                                        C 289
C                                                                       C 290
C     CHECK IF ACCIDENT HAS OCCURRED                                    C 291
C                                                                       C 292
      IF (DNXT.LE.ALV) GO TO 31                                         C 293
C                                                                       C 294
C                                                                       C 295
      EDN=(2.0*VNXT+EPS*ANXT)+DNXT-(2.0*PV+EPS*PA)                      C 296
      IF (EDN.LT.0.0) EDN=0.0                                           C 297
      DNS=0.0                                                           C 298
      IF (VNXT.GT.0.001) DNS=(VNXT/3.50)*(VNXT/2.0)                     C 299
      PDE=EDN+DNS-26.0                                                  C 300
```

```
      IF (PDE.LT.0.0) PDE=0.0                                                  C 301
      PDM=DNXT-26.0                                                            C 302
      IF (PDM.LT.0.0) PDM=0.0                                                  C 303
C                                                                              C 304
CC    IF VEHICLE IS CLOSE REDUCE SPEED                                         C 305
C                                                                              C 306
      IF (PDE.LE.PARM1) VEL3=OPTSPD*SQRT(PDM/PARM1)                            C 307
C                                                                              C 308
C                                                                              C 309
18    CONTINUE                                                                 C 310
      IF (VEL3.LT.0.0) VEL3=0.0                                                C 311
C                                                                              C 312
CC    CALCULATE VELOCITY DUE TO VEHICLE IN LEFT TURN ONLY LANE                 C 313
C                                                                              C 314
      VEL4=OPTSPD                                                              C 315
C                                                                              C 316
CC    DETERMINE NUMBER OF LEFT TURN LANE                                       C 317
C                                                                              C 318
      NLL=1                                                                    C 319
      IF (ABD.EQ.2) NLL=NLA+1                                                  C 320
C                                                                              C 321
CC    CHECK IF ALREADY PAST END OF LEFT TURN ONLY LANE                         C 322
C                                                                              C 323
      IF (PDL.LT.ALENLT(ABD)-ALV) GO TO 20                                     C 324
C                                                                              C 325
CC    CHECK IF NECESSARY TO REDUCE SPEED                                       C 326
C                                                                              C 327
      PNILL=PLCIL(BSS,NLL)                                                     C 328
      IF (PL.EQ.NLL) GO TO 20                                                  C 329
      IF (PNILL.EQ.0) GO TO 20                                                 C 330
      IF (ABD.EQ.1.AND.PL.GE.3) GO TO 20                                       C 331
      IF (ABD.EQ.2.AND.PL.GE.NLA+3) GO TO 20                                   C 332
      IF (PLRS.EQ.1) GO TO 19                                                  C 333
      IF (CARDOP(PNILL,BSP5)+ALV.LE.ALENLT(ABD)) GO TO 20                      C 334
C                                                                              C 335
CC    DETERMINE STATUS OF VEHICLE IN LEFT TURN LANE                            C 336
C                                                                              C 337
19    VNXT=CARDOP(PNILL,BSP3)                                                  C 338
      ANXT=CARDOP(PNILL,BSP4)                                                  C 339
      DNXT=PDL-CARDOP(PNILL,BSP5)                                              C 340
C                                                                              C 341
CC    CHECK IF ACCIDENT HAS OCCURRED                                           C 342
C                                                                              C 343
      IF (DNXT.LT.ALV) GO TO 31                                                C 344
C                                                                              C 345
      EDN=(2.0*VNXT+EPS*ANXT)+DNXT-(2.0*PV+EPS*PA)                             C 346
      IF (EDN.LT.0.0) EDN=0.0                                                  C 347
      DNS=0.0                                                                  C 348
      IF (VNXT.GT.0.001) DNS=(VNXT/3.50)*(VNXT/2.0)                            C 349
      PDE=EDN+DNS-4.0-ALV                                                      C 350
```

```
      IF (PDE.LT.0.0) PDE=0.0                                                  C 351
      PDM=DNXT-26.0                                                            C 352
      IF (PDM.LT.0.0) PDM=0.0                                                  C 353
C                                                                              C 354
CC    IF VEHICLE IS CLOSE REDUCE SPEED                                         C 355
C                                                                              C 356
      IF (PDE.LE.PARM1) VEL4=OPTSPD*SQRT(PDM/PARM1)                            C 357
C                                                                              C 358
CC                                                                             C 359
20    CONTINUE                                                                 C 360
      IF (VEL4.LT.0.0) VEL4=0.0                                                C 361
C                                                                              C 362
CC    CALCULATE VELOCITY DUE TO VEHICLE IN RIGHT TURN ONLY LANE                C 363
C                                                                              C 364
      VEL5=OPTSPD                                                              C 365
C                                                                              C 366
CC    DETERMINE NUMBER OF RIGHT TURN ONLY LANE                                 C 367
C                                                                              C 368
      NRL=NLA                                                                  C 369
      IF (ABD.EQ.2) NRL=NLT                                                    C 370
C                                                                              C 371
CC    CHECK IF ALREADY PAST END OF RIGHT TURN LANE                             C 372
C                                                                              C 373
      IF (PDL.LT.ALENRT(ABD)-ALV) GO TO 22                                     C 374
C                                                                              C 375
CC    CHECK IF NECESSARY TO REDUCE SPEED                                       C 376
C                                                                              C 377
      PNIRL=PLCIL(BSS,NRL)                                                     C 378
      IF (PL.EQ.NRL) GO TO 22                                                  C 379
      IF (PNIRL.EQ.0) GO TO 22                                                 C 380
      IF (ABD.EQ.1.AND.PL.LE.NLA-2) GO TO 22                                   C 381
      IF (ABD.EQ.2.AND.PL.LE.NLT-2) GO TO 22                                   C 382
      IF (PLRS.EQ.2) GO TO 21                                                  C 383
      IF (CARDOP(PNIRL,BSP5)+ALV.LE.ALENRT(ABD)) GO TO 22                      C 384
C                                                                              C 385
CC    DETERMINE STATUS OF VEHICLE IN RIGHT TURN LANE                           C 386
CC                                                                             C 387
21    VNXT=CARDOP(PNIRL,BSP3)                                                  C 388
      ANXT=CARDOP(PNIRL,BSP4)                                                  C 389
      DNXT=PDL-CARDOP(PNIRL,BSP5)                                              C 390
C                                                                              C 391
CC    CHECK IF ACCIDENT HAS OCCURRED                                           C 392
C                                                                              C 393
      IF (DNXT.LT.ALV) GO TO 31                                                C 394
C                                                                              C 395
CC                                                                             C 396
      EDN=(2.0*VNXT+EPS*ANXT)+DNXT-(2.0*PV+EPS*PA)                             C 397
      IF (EDN.LT.0.0) EDN=0.0                                                  C 398
      DNS=0.0                                                                  C 399
      IF (VNXT.GT.0.001) DNS=(VNXT/3.50)*(VNXT/2.0)                            C 400
```

```
      PDE=EDN+DNS-4.0-ALV                                                    C 401
      IF (PDE.LT.0.0) PDE=0.0                                                C 402
      PDM=DNXT-26.0                                                          C 403
      IF (PDM.LT.0.0) PDM=0.0                                                C 404
C                                                                            C 405
CC    IF VEHICLE IS CLOSE REDUCE SPEED                                       C 406
C                                                                            C 407
      IF (PDE.LE.PARM1) VEL5=OPTSPD*SQRT(PDM/PARM1)                          C 408
C                                                                            C 409
CC                                                                           C 410
22    CONTINUE                                                               C 411
      IF (VEL5.LT.0.0) VEL5=0.0                                              C 412
C                                                                            C 413
CC    CALCULATE STATUS OF VEHICLE AT NEXT TIME-SLICE                         C 414
CC                                                                           C 415
CC                                                                           C 416
CC    CALCULATE FUTURE ACCELERATION AND VELOCITY                             C 417
C                                                                            C 418
      VW=AMIN1(VEL1,VEL2,VEL3,VEL4,VEL5)                                     C 419
      IF (VW.GT.OPTSPD) VW=OPTSPD                                            C 420
      AW=(VW-PV)/EPS                                                         C 421
      IF (AW.LT.ALL) AW=ALL                                                  C 422
      IF (AW.GT.AUL) AW=AUL                                                  C 423
      FA=AW                                                                  C 424
      FV=PV                                                                  C 425
      IF (ABS(FA).GT.0.001) FV=PV+FA*EPS                                     C 426
      IF (FV.LT.0.0) FV=0.0                                                  C 427
C                                                                            C 428
CC    CHECK IF VEHICLE HAS LEFT SIMULATION AREA                              C 429
C                                                                            C 430
      DT=(PV+FV)/2.0*EPS                                                     C 431
      IF (DT.GE.PDL) GO TO 27                                                C 432
C                                                                            C 433
CC    CALCULATE FUTURE LANE NUMBER                                           C 434
C                                                                            C 435
      FL=PL                                                                  C 436
      IF (PLRS.EQ.1.AND.PDL-DT.LE.ALENLT(ABD)) FL=(ABD-1)*NLA+1              C 437
      IF (PLRS.EQ.2.AND.PDL-DT.LE.ALENRT(ABD)) FL=NLT-(NLT-NLA)*(2-ABD)      C 438
C                                                                            C 439
CC    CALCULATE FUTURE LEFT-RIGHT-STRAIGHT CODE AND DISTANCE TO LIGHT        C 440
C                                                                            C 441
      FLRS=PLRS                                                              C 442
      FDL=PDL-DT                                                             C 443
C                                                                            C 444
CC    CHECK IF VEHICLE HAS JUST ENTERED TIMED AREA                           C 445
C                                                                            C 446
      IF (FDL.LT.ALENST.AND.TE(I).EQ.-1.0) TE(I)=TIME                        C 447
C                                                                            C 448
CC    ADJUST POINTERS TO NEXT VEHICLE ETC.                                   C 449
C                                                                            C 450
```

```
      PPNT=PLCIL(NBSS,FL)                                                C 451
      IF (PPNT.EQ.0) GO TO 24                                            C 452
      IF (FDL.GE.CARDOP(PPNT,NBSP5)) GO TO 24                            C 453
23    PNT=CARDOP(PPNT,NBS)                                               C 454
      IF (PNT.EQ.0) GO TO 25                                             C 455
      DLP=CARDOP(PNT,NBSP5)                                              C 456
      IF (FDL.GT.DLP) GO TO 25                                           C 457
      PPNT=PNT                                                           C 458
      GO TO 23                                                           C 459
C                                                                        C 460
C     THIS (SO FAR) IS THE LAST VEHICLE IN THIS LANE                     C 461
C                                                                        C 462
24    CARDOP(I,NBS)=PLCIL(NBSS,FL)                                       C 463
      PLCIL(NBSS,FL)=I                                                   C 464
      GO TO 26                                                           C 465
C                                                                        C 466
C     THIS IS NOT THE LAST VEHICLE IN THIS LANE                          C 467
C                                                                        C 468
25    CARDOP(PPNT,NBS)=I                                                 C 469
      CARDOP(I,NBS)=PNT                                                  C 470
26    CONTINUE                                                           C 471
      GO TO 30                                                           C 472
C                                                                        C 473
C     REMOVE VEHICLE FROM SIMULATION                                     C 474
C                                                                        C 475
27    PFV(I)=PFFV                                                        C 476
      PFFV=I                                                             C 477
      IF (PLCIL(NBSS,PL).EQ.I) PLCIL(NBSS,PL)=0                          C 478
C                                                                        C 479
C     RECORD TIME TAKEN TO TRAVERSE LEG                                  C 480
C                                                                        C 481
      IF (TE(I).EQ.-1.0) GO TO 28                                        C 482
      X=TIME-TE(I)                                                       C 483
      T(NALT)=X                                                          C 484
      NALT=NALT+1                                                        C 485
C                                                                        C 486
C     CHECK IF SIMULATION COMPLETE                                       C 487
C                                                                        C 488
      IF (NALT.GT.NTC) RETURN                                            C 489
C                                                                        C 490
C                                                                        C 491
28    CONTINUE                                                           C 492
C                                                                        C 493
C     CLEAR RECORD FOR THIS VEHICLE                                      C 494
C                                                                        C 495
      TE(I)=-1.0                                                         C 496
      DO 29 J1=1,12                                                      C 497
      CARDOP(I,J1)=0.0                                                   C 498
29    CONTINUE                                                           C 499
      GO TO 36                                                           C 500
```

```
C                                                                       C 501
CC      RECORD STATUS OF VEHICLE AT NEXT TIME-SLICE                     C 502
C                                                                       C 503
30      CONTINUE                                                        C 504
        CARDOP(I,NBSP1)=FL                                              C 505
        CARDOP(I,NBSP2)=FLRS                                            C 506
        CARDOP(I,NBSP3)=FV                                              C 507
        CARDOP(I,NBSP4)=FA                                              C 508
        CARDOP(I,NBSP5)=FDL                                             C 509
        GO TO 36                                                        C 510
C                                                                       C 511
CC      REMOVE VEHICLE FROM SIMULATION (ACCIDENT)                       C 512
C                                                                       C 513
31      CONTINUE                                                        C 514
        I1=I                                                            C 515
        NA=NA+1                                                         C 516
C                                                                       C 517
CC      CHECK IF LAST VEHICLE IN THIS LANE                              C 518
C                                                                       C 519
        PL=CARDOP(I1,BSP1)                                              C 520
        IBP=PLCIL(BSS,PL)                                               C 521
        IF (IBP.EQ.I1) GO TO 33                                         C 522
C                                                                       C 523
CC      FIND VEHICLE IMMEDIATELY BEHIND ONE BEING REMOVED               C 524
C                                                                       C 525
32      IBP=CARDOP(IBP,BS)                                              C 526
        IF (IBP.NE.I1) GO TO 32                                         C 527
        CARDOP(IBP,BS)=CARDOP(I1,BS)                                    C 528
        GO TO 34                                                        C 529
C                                                                       C 530
CC      ADJUST POINTERS AND CLEAR STATUS ARRAY                          C 531
C                                                                       C 532
33      PLCIL(BSS,PL)=CARDOP(I1,BS)                                     C 533
34      PFV(I1)=PFFV                                                    C 534
        PFFV=I1                                                         C 535
C                                                                       C 536
CC      CLEAR RECORD FOR THIS VEHICLE                                   C 537
C                                                                       C 538
        TE(I1)=-1.0                                                     C 539
        DO 35 I5=1,12                                                   C 540
        CARDOP(I1,I5)=0.0                                               C 541
35      CONTINUE                                                        C 542
36      CONTINUE                                                        C 543
        NLTM1=NLT-1                                                     C 544
C                                                                       C 545
CC      CHECK TO SEE IF ANY VEHICLES ARE TO BE ADDED TO SIMULATION      C 546
C                                                                       C 547
        DO 43 I=2,NLTM1                                                 C 548
C                                                                       C 549
CC      CHECK IF NO VEHICLE AVAILABLE                                   C 550
```

```
C                                                                       C 551
      IF (PFFV.EQ.0) GO TO 8                                            C 552
C                                                                       C 553
C     CHECK IF ATTEMPTING TO ADD VEHICLES DIRECTLY TO LEFT TURN OR      C 554
C     RIGHT TURN ONLY LANES                                             C 555
C                                                                       C 556
      IF (I.EQ.NLA.OR.I.EQ.NLA+1) GO TO 43                              C 557
      IJF=1                                                             C 558
C                                                                       C 559
C     DETERMINE NUMBER OF VEHICLES ENTERING LANE                        C 560
C                                                                       C 561
      IF (I.GT.NLA) IJF=2                                               C 562
      VEL=VICL(IJF)                                                     C 563
      IF (I.EQ.2.OR.I.EQ.NLA+1) VEL=VILL(IJF)                           C 564
      IF (I.EQ.NLA-1.OR.I.EQ.NLT-1) VEL=VIRL(IJF)                       C 565
C                                                                       C 566
C     CHECK IF NOT TIME TO ADD ANOTHER VEHICLE YET                      C 567
C                                                                       C 568
      Z=3600/VEL+TVEL(I)                                                C 569
      IF (Z.GT.TIME) GO TO 43                                           C 570
      TVEL(I)=TIME                                                      C 571
C                                                                       C 572
C     DETERMINE NUMBER OF VEHICLE TO BE ADDED                           C 573
C                                                                       C 574
      IPNT=PFFV                                                         C 575
      PFFV=PFV(IPNT)                                                    C 576
C                                                                       C 577
C     DETERMINE PROBABILITY THAT VEHICLE ENTERS BY TURNING FROM SIDE    C 578
C     STREET                                                            C 579
C                                                                       C 580
      IF (ASLIT(I).EQ.1) GO TO 37                                       C 581
      GT=CSL(I,2)-CSL(I,1)                                              C 582
      IF (CSL(I,1).GT.CSL(I,2)) GT=60-CSL(I,1)+CSL(I,2)                 C 583
      PROB=(1.0-VES(IJF)/VTOT(IJF))*(NSIC/GT)                           C 584
      CALL RANDU (IX,IX,YFL)                                            C 585
C                                                                       C 586
C     CHECK IF VEHICLE ENTERS BY TURNING FROM SIDE STREET               C 587
C                                                                       C 588
      IF (YFL.GE.PROB) GO TO 37                                         C 589
C                                                                       C 590
C     VEHICLE TO BE ADDED UPSTREAM                                      C 591
C                                                                       C 592
      FV=0.0                                                            C 593
      FA=0.0                                                            C 594
      FDL=ALENST-26.0                                                   C 595
      TE(IPNT)=TIME                                                     C 596
      GO TO 38                                                          C 597
C                                                                       C 598
C     VEHICLE TURNED INTO SIMULATION AREA FROM STREET                   C 599
C                                                                       C 600
```

```
37      FV=OPTSPD                                                          C 601
        FDL=SDTL                                                           C 602
        FA=0.0                                                             C 603
        TE(IPNT)=-1.0                                                      C 604
38      CONTINUE                                                           C 605
C                                                                          C 606
C       DETERMINE IF VEHICLE IS TO TURN LEFT OR RIGHT                      C 607
C                                                                          C 608
        FL=I                                                               C 609
        PROBR=0.0                                                          C 610
        PROBL=0.0                                                          C 611
        IF (I.EQ.2.OR.I.EQ.NLA+1) PROBL=VTL(IJF)/VILL(IJF)                 C 612
        IF (I.EQ.NLA-1.OR.I.EQ.NLT-1) PROBR=VTR(IJF)/VIRL(IJF)             C 613
        ILRS=3                                                             C 614
        CALL RANDU (IX,IX,YFL)                                             C 615
        IF (YFL.LE.PROBL) ILRS=1                                           C 616
        IF (YFL.GE.1.0-PROBR) ILRS=2                                       C 617
        FLRS=ILRS                                                          C 618
C                                                                          C 619
C       ADJUST POINTER TO VEHICLE AHEAD                                    C 620
C                                                                          C 621
        PPNT=PLCIL(NBSS,FL)                                                C 622
        IF (PPNT.EQ.0) GO TO 40                                            C 623
        IF (FDL.GE.CARDOP(PPNT,NBSP5)) GO TO 40                            C 624
39      PNT=CARDOP(PPNT,NBS)                                               C 625
        IF (PNT.EQ.0) GO TO 41                                             C 626
        DLP=CARDOP(PNT,NBSP5)                                              C 627
        IF (FDL.GT.DLP) GO TO 41                                           C 628
        PPNT=PNT                                                           C 629
        GO TO 39                                                           C 630
C                                                                          C 631
C       VEHICLE ADDED IS LAST IN LANE                                      C 632
C                                                                          C 633
40      CARDOP(IPNT,NBS)=PLCIL(NBSS,FL)                                    C 634
        PLCIL(NBSS,FL)=IPNT                                                C 635
        GO TO 42                                                           C 636
C                                                                          C 637
C       VEHICLE ADDED IS NOT THE LAST IN THE LANE                          C 638
C                                                                          C 639
41      CARDOP(PPNT,NBS)=IPNT                                              C 640
        CARDOP(IPNT,NBS)=PNT                                               C 641
C                                                                          C 642
C       RECORD STATUS OF NEW VEHICLE                                       C 643
C                                                                          C 644
42      CONTINUE                                                           C 645
        CARDOP(IPNT,NBSP1)=FL                                              C 646
        CARDOP(IPNT,NBSP2)=FLRS                                            C 647
        CARDOP(IPNT,NBSP3)=FV                                              C 648
        CARDOP(IPNT,NBSP4)=FA                                              C 649
        CARDOP(IPNT,NBSP5)=FDL                                             C 650
```

```
C                                                                       C 651
CC      END OF ITERATION FOR ONE VEHICLE                                C 652
C                                                                       C 653
43      CONTINUE                                                        C 654
C                                                                       C 655
CC      BEGIN ANOTHER TIME-SLICE                                        C 656
C                                                                       C 657
        GO TO 8                                                         C 658
C                                                                       C 659
CC                                                                      C 660
        END                                                             C 661-
```

APPENDIX C

DOCUMENTATION OF SUBROUTINES USED IN PHASE III

```
      SUBROUTINE TREE                                                   A    1
C                                                                       A    2
C                                                                       A    3
C     THIS SUBROUTINE IS COPYRIGHT 1974 BY ROBERT J. TAYLOR             A    4
C     VICTORIA, BRITISH COLUMBIA                                        A    5
C     CANADA                                                            A    6
C                                                  NOV. 1974            A    7
C                                                                       A    8
C                                                                       A    9
C     PURPOSE:     TO CHOOSE A SPANNING TREE OF THE GRAPH ASSOCIATED    A   10
C                  WITH A GIVEN STREET SYSTEM AND TO SET THE OFFSETS OF  A   11
C                  THE SIGNALS SO THAT THE DELAY IS MINIMAL.            A   12
C                                                                       A   13
C                                                                       A   14
C     METHOD:      RANDOM WEIGHTS BETWEEN 2 AND 3 ARE ASSIGNED TO ALL   A   15
C                  EDGES OF THE GRAPH; EDGES THAT ARE NOT IN THE GRAPH  A   16
C                  ARE ASSIGNED A LARGE WEIGHT; EDGES THAT ARE TO       A   18
C                  BE INCLUDED IN THE TREE ARE ASSIGNED A WEIGHT        A   19
C                  OF 1.  USING PRIM'S ALGORITHM A MINIMAL SPANNING     A   20
C                  TREE IS FORMED.  THE OFFSETS OF THE SIGNALS ARE      A   21
C                  CALULATED AS DESCRIBED IN CHAPTER 5 OF THE COVERING  A   22
C                  WORK.                                                A   23
C                                                                       A   24
C                                                                       A   25
C     VARIABLES:                                                        A   26
C                                                                       A   27
C                                                                       A   28
C     Q(I,J)       (I=1,2,...,100); (J=1,2,...,100)                    A   29
C                  WEIGHT OF EDGE FROM INTERSECTION I TO J.             A   30
C                                                                       A   31
C     A(K)         (K=1,2,...,100)                                      A   32
C                  A VALUE FOR KTH LEG.                                 A   33
C                                                                       A   34
C     B(K)         (K=1,2,...,100)                                      A   35
C                  B VALUE FOR KTH LEG.                                 A   36
C                                                                       A   37
C     C(K)         (K=1,2,...,100)                                      A   38
C                  C VALUE FOR KTH LEG.                                 A   39
C                                                                       A   40
C     CGIJ(K)      (K=1,2,...,100)                                      A   41
C                  NUMBER OF VEHICLES GOING FROM I TO J.                A   42
C                                                                       A   43
C     CGJI(K)      (K=1,2,...,100)                                      A   44
C                  NUMBER OF VEHICLES GOING FROM J TO I.                A   45
C                                                                       A   46
C     EI(K)        (K=1,2,...,100)                                      A   47
C                  NUMBER OF THE FIRST SIGNAL OF LEG K.                 A   48
C                                                                       A   49
C     EJ(K)        (K=1,2,...,100)                                      A   50
C                  NUMBER OF THE SECOND SIGNAL OF LEG K.                A   51
```

```
C
C      I              DO-LOOP PARAMETER                                        A   52
C                                                                              A   53
C      IEI            NUMBER OF THE FIRST SIGNAL OF LEG K.                     A   54
C                                                                              A   55
C      IMIN           NUMBER OF INTERSECTION ON ONE END OF EDGE TO BE          A   56
C                     ADDED.                                                   A   57
C                                                                              A   58
C      INTI           NUMBER OF SIGNAL I.                                      A   59
C                                                                              A   60
C      INTJ           NUMBER OF SIGNAL J.                                      A   61
C                                                                              A   62
C      IX             RANDOM NUMBER GENERATOR SEED.                            A   63
C                                                                              A   64
C      J              DO-LOOP PARAMETER                                        A   65
C                                                                              A   66
C      JEJ            NUMBER OF THE SECOND SIGNAL OF LEG K.                    A   67
C                                                                              A   68
C      JMIN           NUMBER OF INTERSECTION ON ONE END OF EDGE TO BE          A   69
C                     ADDED.                                                   A   70
C                                                                              A   71
C      K              DO-LOOP PARAMETER.                                       A   72
C                                                                              A   73
C      N              NUMBER OF SIGNALS (VERTICES)                             A   74
C                                                                              A   75
C      NL             NUMBER OF LEGS (EDGES)                                   A   76
C                                                                              A   77
C      NM1            N - 1                                                    A   78
C                                                                              A   79
C      NSIC           NUMBER OF SECONDS IN THE CYCLE.                          A   80
C                                                                              A   81
C      OFF(I)         (I=1,2,...,100)                                          A   82
C                     OFFSET OF SIGNAL I.                                      A   83
C                                                                              A   84
C      OT             OPTIMAL RELATIVE OFFSET FOR THE LEG.                     A   85
C                                                                              A   86
C      RMIN           MINIMUM WEIGHT OF THE EDGES.                             A   87
C                                                                              A   88
C      S              DO-LOOP PARAMETER.                                       A   89
C                                                                              A   90
C      TAKEN(J)       (J=1,2,...,100)                                          A   91
C                     INDICATES IF JTH INTERSECTION INCLUDED IN TREE           A   92
C                          0:  NO                                              A   93
C                          1:  YES                                             A   94
C                                                                              A   95
C      XIETG(K)       (K=1,2,...,100)                                          A   96
C                     TIME THAT SIGNAL I TURNS GREEN FOR VEHICLES              A   97
C                     ENTERING LEG K.                                          A   98
C                                                                              A   99
C      XILTG(K)       (K=1,2,...,100)                                          A  100
C                                                                              A  101
```

```
C                 TIME THAT SIGNAL I TURNS GREEN FOR VEHICLES                  A 102
C                 LEAVING LEG K.                                               A 103
C                                                                              A 104
C       XJETG(K)  (K=1,2,...,100)                                              A 105
C                 TIME THAT SIGNAL J TURNS GREEN FOR VEHICLES                  A 106
C                 ENTERING LEG K.                                              A 107
C                                                                              A 108
C       XJLTG(K)  (K=1,2,...,100)                                              A 109
C                 TIME THAT SIGNAL J TURNS GREEN FOR VEHICLES                  A 110
C                 LEAVING LEG K.                                               A 111
C                                                                              A 112
C       YFL       RANDOM NUMBER BETWEEN 0 AND 1.                               A 113
C                                                                              A 114
C       Z4        TEMPORARY LOCATION.                                          A 115
C                                                                              A 116
C                                                                              A 117
C       CALLING:  WHEN CALLING THE SUBROUTINE THE FOLLOWING                    A 118
C                 VARIABLES MUST BE DEFINED.                                   A 119
C                                                                              A 120
C                                                                              A 121
C                 NSIC                                                         A 122
C                 N                                                            A 123
C                 NL                                                           A 124
C                 EI(K)       K=1,2,...,NL                                     A 126
C                 EJ(K)       K=1,2,...,NL                                     A 127
C                 A(K)        K=1,2,...,NL                                     A 128
C                 B(K)        K=1,2,...,NL                                     A 129
C                 C(K)        K=1,2,...,NL                                     A 130
C                 CGIJ(K)     K=1,2,...,NL                                     A 131
C                 CGJI(K)     K=1,2,...,NL                                     A 132
C                 XIETG(K)    K=1,2,...,NL                                     A 133
C                 XILTG(K)    K=1,2,...,NL                                     A 134
C                 XJETG(K)    K=1,2,...,NL                                     A 135
C                 XJLTG(K)    K=1,2,...,NL                                     A 136
C                                                                              A 137
C                                                                              A 138
C       OUTPUT:                                                                A 139
C                                                                              A 140
C                 OUTPUT CONSISTS OF A LISTING OF THE EDGES                    A 141
C                 INCLUDED IN THE TREE AS WELL AS THE OFFSETS OF THE           A 142
C                 SIGNALS (EXPRESSED IN SECONDS AND % OF CYCLE).               A 143
C                                                                              A 144
C                                                                              A 145
C       INPUT:                                                                 A 146
C                                                                              A 147
C                 FOR EACH EDGE FORCED TO BE INCLUDED IN THE TREE              A 148
C                 THERE SHOULD BE A SINGLE CARD INDICATING THE NUMBERS         A 149
C                 OF THE SIGNALS AT EITHER END OF THE LEG.                     A 150
C                 FORMAT(I3,2X,I3).                                            A 151
C                                                                              A 152
```

```
C                                                                       A 153
      COMMON N,NL,EI,EJ,A,B,C,CGIJ,CGJI,NSIC                            A 154
      COMMON XIETG,XILTG,XJETG,XJLTG                                    A 155
C                                                                       A 156
C                                                                       A 157
      DIMENSION Q(100,100), OFF(100), A(100), B(100), C(100)            A 158
      DIMENSION CGJI(100), CGIJ(100)                                    A 159
      DIMENSION XIETG(100), XILTG(100), XJETG(100), XJLTG(100)          A 160
      INTEGER EI(100),EJ(100),TAKEN(100),S                              A 161
C                                                                       A 162
CC                                                                      A 163
CC    INITIALIZATION SECTION                                            A 164
C                                                                       A 165
      RMIN=100000.0                                                     A 166
      IX=654231753                                                      A 167
C                                                                       A 168
CC    SET THE WEIGHTS TO A LARGE VALUE                                  A 169
C                                                                       A 170
      DO 2 I=1,N                                                        A 171
      DO 1 J=1,N                                                        A 172
      Q(I,J)=RMIN                                                       A 173
1     CONTINUE                                                          A 174
2     CONTINUE                                                          A 175
C                                                                       A 176
CC    FOR THOSE EDGES THAT EXIST SET THE WEIGHTS TO BETWEEN 2 AND 3     A 177
C                                                                       A 178
      DO 3 K=1,NL                                                       A 179
      IEI=EI(K)                                                         A 180
      JEJ=EJ(K)                                                         A 181
      CALL RANDU (IX,IX,YFL)                                            A 182
      Q(IEI,JEJ)=YFL+2.0                                                A 183
      Q(JEJ,IEI)=YFL+2.0                                                A 184
3     CONTINUE                                                          A 185
CC                                                                      A 186
CC    READ IN AN EDGE THAT SHOULD BE INCLUDED IN THE TREE               A 187
C                                                                       A 188
4     READ (5,15,END=16) INTI,JNTI                                      A 189
CC                                                                      A 190
CC    ASSIGN IT WEIGHT 1                                                A 191
C                                                                       A 192
      Q(INTI,JNTI)=1.0                                                  A 193
      Q(JNTI,INTI)=1.0                                                  A 194
      GO TO 4                                                           A 195
C                                                                       A 196
CC    SET TAKEN TO INDICATE THAT NONE OF THE INTERSECTIONS HAVE BEEN    A 197
CC    INCLUDED IN THE TREE AS YET.  SET ALL THE OFFSETS TO 0.           A 198
C                                                                       A 199
16    DO 5 I=1,N                                                        A 200
      TAKEN(I)=0                                                        A 201
      OFF(I)=0.0                                                        A 202
```

```
5       CONTINUE                                                                A 203
C                                                                               A 204
C       CHOOSE THE FIRST INTERSECTION AND GIVE IT OFFSET EQUAL TO 0             A 205
C                                                                               A 206
        TAKEN(1)=1                                                              A 207
        OFF(1)=0.0                                                              A 208
C                                                                               A 209
C       FOR EACH OF THE EDGES TO BE INCLUDED IN THE TREE                        A 210
C                                                                               A 211
        NM1=N-1                                                                 A 212
        DO 10 S=1,NM1                                                           A 213
C                                                                               A 214
C       FIND THE ONE THAT IS CLOSEST                                            A 215
C                                                                               A 216
        RMIN=100000.0                                                           A 217
        DO 7 I=1,N                                                              A 218
        IF (TAKEN(I).EQ.0) GO TO 7                                              A 219
        DO 6 J=1,N                                                              A 220
        IF (TAKEN(J).EQ.1.OR.Q(I,J).GE.RMIN) GO TO 6                            A 221
        IMIN=I                                                                  A 222
        JMIN=J                                                                  A 223
        RMIN=Q(I,J)                                                             A 224
6       CONTINUE                                                                A 225
7       CONTINUE                                                                A 226
C                                                                               A 227
C       DETERMINE THE CORRESPONDING EDGE                                        A 228
C                                                                               A 229
        DO 8 K=1,NL                                                             A 230
        IEI=EI(K)                                                               A 231
        JEJ=EJ(K)                                                               A 232
        IF (IMIN.EQ.IEI.AND.JMIN.EQ.JEJ) GO TO 9                                A 233
        IF (JMIN.EQ.IEI.AND.IMIN.EQ.JEJ) GO TO 9                                A 234
8       CONTINUE                                                                A 235
9       CONTINUE                                                                A 236
C                                                                               A 237
C       WRITE OUT THAT THE EDGE WAS INCLUDED                                    A 238
C                                                                               A 239
        I=IEI                                                                   A 240
        J=JEJ                                                                   A 241
        WRITE (6,12) I,J                                                        A 242
C                                                                               A 243
C       DETERMINE THE OPTIMAL OFFSET FOR THE LEG                                A 244
C                                                                               A 245
C       OT=45.0+B(K)                                                            A 246
C                                                                               A 247
C       DETERMINE THE OFFSET OF THE INTERSECTION JUST ADDED                     A 248
C                                                                               A 249
        IF (TAKEN(IEI).EQ.1.AND.CGIJ(K).GE.CGJI(K)) OFF(J)=OT+XIETG(K)+OFF      A 250
     1(I)-XJLTG(K)                                                              A 251
        IF (TAKEN(IEI).EQ.1.AND.CGIJ(K).LT.CGJI(K)) OFF(J)=-OT-XJETG(K)+OF      A 252
```

```
     1F(I)+XILTG(K)                                                                    A 253
      IF (TAKEN(JEJ).EQ.1.AND.CGIJ(K).GE.CGJI(K)) OFF(I)=-OT-XIETG(K)+OF               A 254
     1F(J)+XJLTG(K)                                                                    A 255
      IF (TAKEN(JEJ).EQ.1.AND.CGIJ(K).LT.CGJI(K)) OFF(I)=OT+XJETG(K)+OFF               A 256
     1(J)-XILTG(K)                                                                     A 257
C                                                                                      A 258
C     INDICATE THAT THE INTERSECTION IS NOW INCLUDED IN THE TREE                       A 259
C                                                                                      A 260
      TAKEN(JMIN)=1                                                                    A 261
10    CONTINUE                                                                         A 262
      WRITE (6,14)                                                                     A 263
C                                                                                      A 264
C     OUTPUT THE OFFSETS AND THE % OFFSETS                                             A 265
C                                                                                      A 266
      DO 11 I=1,N                                                                      A 267
      OFF(I)=AMOD(OFF(I)+10.0*NSIC,NSIC*1.0)                                           A 268
      Z4=OFF(I)/NSIC*100.0                                                             A 269
      WRITE (6,13) I,OFF(I),Z4                                                         A 270
11    CONTINUE                                                                         A 271
C                                                                                      A 272
C                                                                                      A 273
      RETURN                                                                           A 274
C                                                                                      A 275
C                                                                                      A 276
C                                                                                      A 277
12    FORMAT (1X,'EDGE ',I6,'-',I6,'   OPTIMIZED')                                     A 278
13    FORMAT (I6,5X,F8.1,'  SECONDS   ',F8.1,'  %')                                    A 279
14    FORMAT (1H1,///,' OPTIMAL OFFSETS',///,' INTERSECTION   OFFSET'/)                A 280
15    FORMAT (I3,2X,I3)                                                                A 281
      END                                                                              A 282-
```

VITA

Surname: TAYLOR Given Names: ROBERT JOHN

Place of Birth: REGINA, SASKATCHEWAN Date of Birth: April 13, 1950

Educational Institutions Attended, with Dates of Entering and Leaving:

UNIVERSITY OF LETHBRIDGE, LETHBRIDGE, ALBERTA	1968 to 1971
UNIVERSITY OF VICTORIA, VICTORIA, B.C.	1972 to 1975
	____ to ____
	____ to ____

Degrees, Diplomas, Etc., Awarded, with Dates and Names of Institutions:

Bachelor of Arts and Science	1971	University of Lethbridge, Lethbridge

Honors and Awards:

University of Victoria Fellowship, 1972/73

N.R.C. Grants, 1973/1974

Publications:

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis or dissertation (the title of which is shown below) to users of the University of Victoria Library, and to make *single copies only* for such users or in response to a request from the library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis/Dissertation

IMPROVING TRAFFIC CONTROL THROUGH SIMULATION AND OPERATIONS RESEARCH

Author [redacted]
Signature

ROBERT JOHN TAYLOR
Name

March 12, 1975
Date