THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING ON THE PAGE BEING CROOKED.

THIS IS THE BEST IMAGE AVAILABLE.

USE OF COMPUTER PROGRAMMING TO SOLVE TRAFFIC PROGRESSION PROBLEMS

by

GREGORY WARREN HARDIN

B.S., Kansas State University, 1970

A MASTER'S REPORT

submitted in partial fulfillment

of the requirements for the degree

MASTER OF SCIENCE

Department of Civil Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

Approved by

Major Professor

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Traffic congestion in urban areas is a situation unwanted by everyone. Traffic signals are used for the assignment of the right of way at intersections, and thus help to alleviate the problem of congestion. Frequent stops that occur along a signal system serve only to place the driver "on edge," and should therefore be minimized as much as possible.

Developing a traffic signal system which can move the greatest number of vehicles in a given amount of time improves the efficiency of the traffic system. Such a method will maximize bandwidths of traffic, which in turn decreases the total delay in the system. A bandwidth is a platoon or wave of traffic which is able to progress through the green phase of each signal in a system without stopping. This is illustrated in Figure 1 on page 17.

Several ideas concerning signal progression have been developed within the last ten years. Most have been designed for a single street system only, but within the past three years considerable work has been done on developing a system that will apply for an entire network. Although the network approach is not completely successful in the field, is has great potential for improving traffic congestion throughout entire networks in the large cities.

An attempt is made in this report to bring many of these ideas together, but these ideas will not be explained in great detail. For complete explanation of each method, one will have to use the proper reference.

LITTLE'S METHOD

One of the earlier attempts at synchronizing traffic signals for maximal bandwidth was developed in 1964 at the Massachusetts Institute of Technology through the combined efforts of John T. Morgan, J.D.C. Little, and Brian V. Martin.[7] Their idea was to formulate a procedure that would solve traffic signal problems to produce bandwidths of equal width and as large as possible, and to resynchronize to favor one direction with as large a bandwidth as possible and then maximize the bandwidth in the opposite direction.

Notation

Two-way streets will be considered having n traffic signals, with n not to exceed 50 due to computer storage restriction. Directions are identified on the streets as outbound and inbound, with subscripts increasing in the outbound direction for each direction used. The signals are denoted by S_i and the red time of each street is denoted by r_i. The initial red time in seconds is converted to a fraction of a cycle length by dividing by the cycle length C. Bandwidth is denoted by b in the outbound direction and by $\overline{b}$ in the inbound direction. Travel time, t_{ij} and $\overline{t}_{ij}$, is measured from one signal to another in terms of cycle length. The offset, θ_{ij}, is measured as the fraction of the cycle length from the center of the red phase of one signal to the center of the next red phase of the next signal.

It is usually easier to gather data for the distances between signals and to state the expected speed for each section in the system, than to measure travel times. Travel time between adjacent signals is then computed in fraction of a cycle length by the equation:

$$t_{i,i+1} = (x_{i+1} - x_i) / v_i C \qquad (1)$$

Travel times are not necessarily equal in the opposite direction. This is because speeds could vary for the two directions, and the directional distances could possibly be different due to a large median.

Theory

In most traffic situations, it is desired to maximize equal bandwidths. This is stated as: maximize b subject to $b = \bar{b}$. This particular synchronization does not necessarily maximize $(b + \bar{b})$ however. Therefore, two possible cases exist.

Little states that every green band has a limiting critical signal, which he chooses to call S_j.[7] This critical signal can either limit the front of the outbound bandwidth (Group 1) or the front of the inbound bandwidth (Group 2).

Again two cases can be investigated. The first case involves two signals in the same group, while the signals are in opposite groups for the second case. Little proved that under the above conditions each group of signals has half-integer synchronization. This means that each signal has the center of its red phase located at exactly 0 or at $\frac{1}{2}$ the cycle length. Little proved by construction that under any half-integer synchronization that $b = \bar{b}$, which allows the traffic engineer to examine only one direction. A complete proof can be found in pages 9-23 of the manual by Little, Martin, and Morgan.[7]

Input

The original program developed by Little, Martin, and Morgan was given the name TSS 3. It is written in Load and Go Fortran to be run on an IBM 1620 computer. The output from this program is used as input for program TSS 4, which plots a time-space diagram on a Calcomp Digital plotter. However, since this plotter was not available for this project, no further discussion of it will be given. The program for a 1620 computer is given in Appendix A.

About a year after this program was developed, E. T. Miller transferred the language to Fortran IV so that it could be run on all IBM 360 computers. This program is called EMTSS 3, and the program is given in Appendix B.

Four types of input cards are required for the TSS 3 program, and are summarized in Table 1. Card 1 contains the number of signals, the cycle length, the inbound and outbound volumes, and the vehicle headway. This headway is used to convert hourly volumes into platoon lengths, in vehicles, by the following equation:

$$\text{Platoon length} = (\text{Hourly Volume} \times \text{Vehicle Headway}) / 3600 \qquad (2)$$

Card 2 contains the distance of each signal from the origin and the red phase of that signal. Increasing X_i must be in the outbound direction. Card 3 gives the inbound and outbound block speeds. Card 4 directs the program to either continue or stop. If changes are to be made in cycle length and/or speeds, set NSAVE equal to zero and CYCLE equal to the value desired. Otherwise, set NSAVE equal to a positive integer and a new problem can be run.

The deck setup for the TSS 3 program requires three cards to precede the deck. They are the following:

```
MONITOR COLD START
$$JOB
$$XEQ FORUN
```

Table 1.--TSS 3 Input Data

Entry	Description of Data Item	Variable Name	Units	Form
Card type 1				
1	Number of Signals	NSIG	Integer	Fixed
2	Cycle Length	CYCLE	Seconds	Floating
3	Inbound Volume	VOLIN	Veh/hr	Floating
4	Outbound Volume	VOLOT	Veh/hr	Floating
5	Vehicle Headway	HEDWY	Seconds	Floating
	ONE CARD PER PROBLEM			
Card type 2				
1	Distance of Signal from origin	X(I)	Feet	Floating
2	Red Phase of Signal	RED(I)	Seconds	Floating
	NUMBER OF CARDS = NSIG			
	CARDS IN ORDER OF INCREASING Y			
Card type 3				
1	Inbound block speed	SPEDI(I)	mph	Floating
2	Outbound block speed	SPEDO(I)	mph	Floating
	NUMBER OF CARDS = NSIG - 1			
	CARDS IN ORDER OF INCREASING Y			
Card type 4				
1	Constant	NSAVE	Integer	Fixed
2	Cycle length for next iteration	CYCLE	Seconds	Floating

Five types of input cards are required to run EMTSS 3 on the IBM 360 computer, and they are summarized in Table 2. Card 1 is an extra identification card not used in TSS 3. Card 2 is similar to Card 1 of TSS 3, except that three scale factors are added. SCD should be left blank or set equal to 1000 ft/in. SCT should be left blank, and KEY is either left blank or set equal to one. Card 3 is identical to TSS 3 Card 2 except that the name of the cross street at the signal is added for convenience. Card 4 includes the inbound and outbound speeds between signals. Card 5 contains NSAVE and the cycle length. The cycle length is omitted in all cases except when NSAVE is 2 or 4, allowing the problem to be re-run using the same data. If NSAVE is 1 or 3 a completely new problem is read, and if NSAVE if 5 or 6 the program will terminate execution.

The deck setup for the EMTSS 3 program requires a few more cards than TSS 3. The setup should be made as follows:

```
//          JOB (FF18C4Q7,10,10),'NAME',MSGLEVEL=1
//          EXEC FORTGCLG
//FORT.SYSIN DD *

[Program deck goes here]

/*
//GO.SYSIN DD *

[Data goes here]

/*
```

The execution and program cost for an example problem are given on page 52. The running time depends on the number of signals along the street system under study. Special care should be taken to code the proper value for NSAVE in Card 5 of the EMTSS 3 input.

Table 2.--EMTSS 3 Input Data

Columns	Description of Data Item	Symbol	Units	Form
Card type 1	ONE CARD PER PROBLEM			
1-72	Identification information			Alpham.
Card type 2	ONE CARD PER PROBLEM			
1- 4	Number of Signals	NSIG	Integer	Fixed
5-14	Cycle Length	CYCLE	Seconds	Floating
15-24	Inbound Volume	VOLIN	v/hr/ln	Floating
25-34	Outbound Volume	VOLOT	v/hr/ln	Floating
35-44	Vehicle Headway	HEDWY	Seconds	Floating
45-54	Distance Scale Factor	SCD	ft/in	Floating
55-64	Time Scale Factor	SCT	sec/in	Floating
65-68	Speed Card Indicator	KEY		Fixed
Card type 3	NUMBER OF CARDS = NSIG			
1-10	Distance of Signal from origin	X(I)	Feet	Floating
11-20	Red Phase of Signal	RED(I)	Seconds	Floating
21-50	Name of Cross Street at Signal	NAMES(I,J)		Alpham.
Card type 4	NUMBER OF CARDS = NSIG - 1			
1-10	Inbound Speed between Signals	SPEDI(I)	mph	Floating
11-20	Outbound Speed between Signals	SPEDO(I)	mph	Floating
Card type 5	ONE CARD PER PROBLEM			
1- 4	Constant	NSAVE	Integer	Fixed
5-14	Cycle Length	CYCLE	Seconds	Floating

Output

Output from TSS 3 is in the form of punched cards, which then yield the output shown on pages 41-43 when run through a printer. Five types of cards are punched. Type 1 gives the number of signals, the cycle length, and the headway which were all read in as input. Card type 2 gives the input values of inbound and outbound volumes, followed by the inbound and outbound volumes possible through the computed green bandwidths for the input vehicle headway. Card type 3 gives the inbound and outbound bandwidths and the number of the critical signal.

Card type 4 contains as many cards as there are signals. It gives the input values of distance and red split plus the output values of the right hand side of the red band for each signal and the offset of the signal with respect to the critical signal. Card type 5 is primarily of use to program TSS 4. The first entry is the front edge of the outbound band, and it is followed by the rear edge of the inbound band. Both of these values are at the first signal. The third and fourth entries are total system travel time, both inbound and outbound in terms of a fraction of the cycle length.

The notation given for inbound volume of 9.9000E+02 is read as 990. This is found by moving the decimal point two places to the right. Similarly the start of the cycle at the second intersection is read as .16075 of a cycle length.

Output from program EMTSS 3 is shown on page 51 for a sample problem from Fort Worth, Texas. The first five values given are all input data. They are followed vertically by the possible inbound and outbound volumes, the inbound and outbound bandwidths, and the restricting signal. Directly below this is a table of values for each cross street. Given in the table are the distance from the origin, the length of the red phase of the signal, the end of each red phase, the offset from the restricting signal, and the directional progressive speeds. Progressive speed is the speed of the bandwidth. Below this table values are given for the front edge of the outbound band, the rear edge of the inbound band, and the system travel time inbound and outbound. These last four are given in fraction form and in seconds.

Sample Problems

Three sample problems were run on the IBM 1620 computer and both the data and the output are shown in Appendix A. The first two sets of data were from two different streets in Wichita, Kansas. No discussion will be made on these results. The third set of data is for Rosedale Avenue in Fort Worth, Texas. This set of data was taken from an example problem in a handout, and was not obtained by the author.

Only one sample problem was run on the 360 computer with program EMTSS 3. Again the data used was from Rosedale Avenue in Fort Worth, Texas. The results of this run are shown on page 51. A comparison of the outputs for Rosedale Avenue from the two computers shows that the results are exactly the same. This is not surprising, because EMTSS 3 is merely a changeover to Fortran IV from the original program TSS 3. Therefore, Little's method will work on either the 360 or 1620 computer with equal accuracy. The 360 offers the speed that engineers desire.

CHAMBERLAIN'S PROGRAM

The second progression program, developed by Robert Chamberlain, maximizes the green bandwidth and obtains the offsets of the sequence of traffic signals.[3] An initial band is drawn from the intersection with the shortest green time, called the reference intersection. This initial band has a width equal to the green time of the reference intersection. Minimizing the interferences with this band at all intersections, the maximum through-band is obtained.

The program is based on the assumptions that (1) the cycle length is the same for all signals under consideration, and (2) the traffic speed in both directions is constant throughout the section under study.

Theory

Traffic speed is the independent argument used to compute the minimum interference. Speed increments are specified by the input argument Z (Notation can be found on page 11), which should be kept at a lower limit of .25 mph to keep computer running time short. The cycle length used in the computations is CMIN. The speed range is extended to an upper limit of VMAX x CMAX / CMIN to permit generalization to other cycle lengths.

The speed yielding the smallest minimum interference is called the optimum speed. The optimum cycle length is determined by:

$$\text{COPT} = \text{CMIN} \times \text{VOPT} / \text{VMAX} \qquad \text{when VOPT} > \text{VMAX} \qquad (3)$$

$$\text{COPT} = \text{CMIN} \qquad \text{when VOPT} \leq \text{VMAX} \qquad (4)$$

Table 3.--Chamberlain's Notation

Symbol	Explanation of the Symbol
BW	The sum of the inbound and outbound bandwidths in seconds
BW_i	The inbound bandwidth in seconds
BW_o	The outbound bandwidth in seconds
CMAX	The maximum cycle length in seconds
CMIN	The minimum cycle length in seconds
COPT	The optimum cycle length in seconds
DIST(J)	The distance to the jth intersection from the reference intersection, in seconds
FI	The inbound traffic flow in vehicles per hour
FO	The outbound traffic flow in vehicles per hour
TJ(J)	The green time at the jth intersection, in seconds
TO	The green time at the reference intersection, in seconds
VMAX	The maximum traffic speed in mph
VMIN	The minimum traffic speed in mph
VOPT	The optimum traffic speed in mph
XL	The number of intersections minus one under consideration (n-1)
Z	The size of the speed increment in mph

When equation (3) is used, the optimum speed becomes VMAX. It is used in conjunction with COPT. Various cycle lengths may be used to obtain speeds lower than VMAX, while keeping the bandwidth to cycle ratio constant. If a speed of V is desired such that $VMIN \leq V \leq VMAX$, the corresponding cycle length, C, is given by:

$$C = COPT \times VMAX / V \tag{5}$$

This equation may be reversed if a cycle length, C, is given and the resulting speed, V, is to be determined:

$$V = VMAX \times COPT / C \tag{6}$$

If both C and V are known initially, offsets may be determined by setting CMIN equal to CMAX and also setting VMIN equal to VMAX.

The program initially computes offsets based on equal flow in both directions and then apportions the available bandwidth to favor the direction with the largest average traffic flow. The resulting bandwidths are expressed as follows:

$$BW_o = FO \times BW / (FO + FI) \qquad \text{when } BW_o < TO \tag{7}$$

$$BW_i = FI \times BW / (FO + FI) \qquad \text{when } BW_i \leq TO \tag{8}$$

The apportionment of bandwidth is accomplished by shifting the individual offsets to reduce the interferences in the desired direction at the expense of the bandwidth in the opposing direction. If the desired bandwidth in one direction exceeds the available green time (TO), the program selects a bandwidth of TO for that direction.

Input

This program can accommodate up to 50 intersections. There is no limit to the size of the cycle length and speed ranges or to the number of speed increments. However, it is suggested that CMIN have a lower limit of 40 seconds and that CMAX have an upper limit of 120 seconds. The speed increment should have a lower limit of .25 mph as mentioned above.

The object program for this method is shown in Appendix C. It was written in Fortran originally for an 1130 computer. With the control cards given below, this program will run on an IBM 360. The input data are defined in floating point mode, with one number not to exceed 6 columns including the decimal point. The first data card must contain the following information in F6.1 format, and the numbers below represent the first significant digit of each field:

1	7	13	19	25	31	37	43	49
CMIN	CMAX	TO	XL	VMIN	VMAX	Z	FO	FI

The next group of cards is under the format 2F10.2, and the number of cards should equal XL. The intersections should be ordered according to increasing distance from the reference intersection with the smallest distance first. The cards should contain the following information with the number again representing the first digit:

1	11
DIST(J)	TJ(J)

The cards should be placed in the following order with the control cards as shown:

```
//          JOB (FF18C4Q7,10,10),'NAME',MSGLEVEL=1
//          EXEC FTGCLGKS
//FORT.SYSIN DD *

[Program deck goes here]

/*
//GO.SYSIN DD *

[Data goes here]

/*
```

Output

Output are printed as shown on pages 59-61 for the sample used later in this section. The input data are printed first and are followed by the individual minimum interferences. Only the first two and the last iteration have been shown. Following the last iteration the input data are again printed for convenience. The optimum speed, the optimum cycle length, and the optimum bandwidth follow the input data.

Individual offsets are then given for each intersection with the reference intersection having an offset of zero. The offset is the starting time of the green time of each intersection relative to the beginning of green of the reference intersection. For example intersection 1 on page 60 has its green phase starting 8.8 seconds after the green phase of the reference intersection. For unequal flow, the apportioned offsets along with flows in both directions, are also printed. Outbound and inbound flow are given on page 61.

The running time to calculate the results for the sample problem below is shown on page 62. The cost for this 5-intersection problem was $3.30 for a range of 30 seconds in cycle length.

Sample Calculation

The data used for this problem were obtained from Anderson Avenue in Manhattan, Kansas, between Sunset Avenue and Fourteenth Street. From the output on page 60, the speed associated with the smallest interference is 44 mph, or VOPT. Since VOPT is greater than VMAX, a new cycle length must be calculated from equation (3). Hence, from equation (3):

COPT = CMIN x VOPT / VMAX

Substituting gives:

COPT = 50 x 44 / 30 = 73.3 seconds

and VOPT becomes VMAX.

The bandwidth is given as follows:

BW = ((2 x TO) - XMIN) x S

where

TO = 19 seconds (input)

XMIN = 8.465 seconds (output)

S = VOPT / VMAX for VOPT > VMAX

S = 1 for VOPT ≤ VMAX

Substituting yields:

BW = ((2 x 19) - 8.465) 44/30 = 29.535 x 1.467 = 43.32 seconds

If the resultant speed is too high, a longer cycle length can be used provided BW/C remains constant. The values obtained above would be substituted into equation (3) to determine a new C. The new bandwidth BW' is found from:

BW'/C' = BW/C

or

BW' = BW x C'/C

Results

The results from this program are based upon the initial values given as input. Since green time for each intersection is initially fixed, it is essential that the minimum cycle length, CMIN, be selected with some thought. As can be seen from the four equations and Table 4, CMIN is the controlling factor in determining a reasonable bandwidth.

Table 4.--Sample Data for Anderson Avenue

CMIN	CMAX	VMAX	VOPT	XMIN	COPT	BW
40	80	30	55	4.5	73.3	61.41
50	80	30	44	8.465	73.3	43.32
55	80	30	39	10.343	71.5	35.95

If it is desired to allow the cycle length to range from 40 to 80 seconds, the green time for each intersection should initially be reduced so that reasonable results would be obtained from the program for bandwidth. The efficiency of a signal system is the percentage of the cycle length occupied by the bandwidth. For a minimum cycle of 40 seconds, a bandwidth of 61.41 seconds is given in Table 4. This gives an efficiency of nearly 84%, which is not a very practical solution. This would allow for almost no green time for the cross streets.

Similarly, with a minimum cycle of 50 seconds a bandwidth of 43.32 seconds is obtained. This bandwidth produces a system efficiency of about 59%, which again is higher than expected. The reason for these high efficiencies is that the green times used for the intersections are those that now occur for the street as it operates on a 60-second cycle length. Since CMIN is the limiting factor and the initial run used a CMIN of 40 seconds, a better solution would have been to decrease the green times for each intersection by a factor of 40/60, or two-thirds.

SIGPROG

Another traffic-signal-system timing plan was developed by R. L. Bleyl of the Yale University Bureau of Highway Traffic.[1] His approach converts all time and distance units to travel time units.

Approach

A conventional time-space diagram, as shown in Figure 1, plots time along one axis and distance along the other. Speeds are represented by the slope of a line along the diagram. The modified diagram, as shown in Figure 2, changes the vertical axis to average travel time. To determine travel time, travel distances between stop lines must be known for each direction as well as the desired progressive speed.

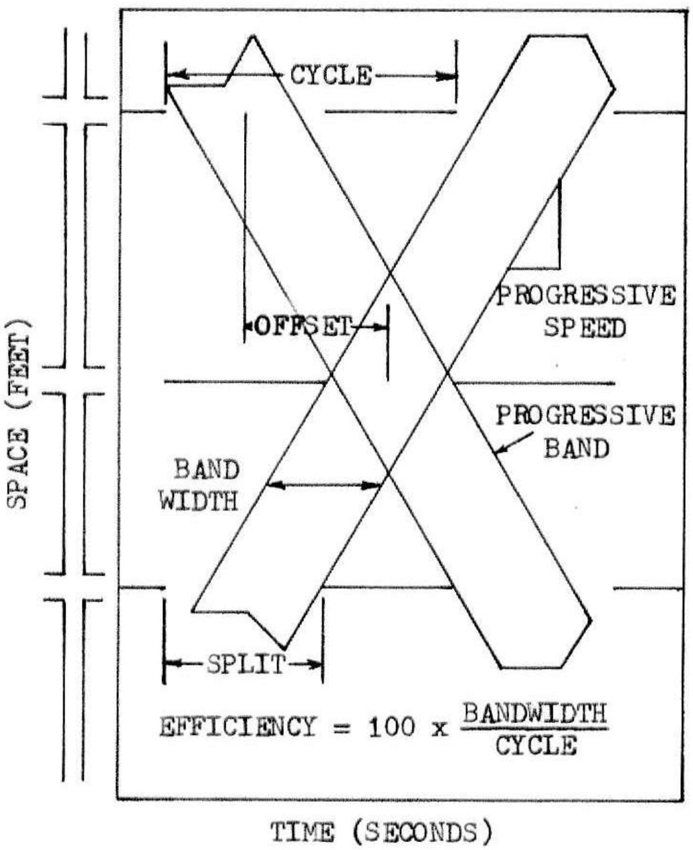

Figure 1. Conventional time-space diagram.[1]

The time to travel between successive signals is found by dividing the distance between intersections by the desired progressive speed. This speed is multiplied by 1.47 to convert to feet per second.

$$\text{Travel time} = \text{Distance (feet)} / (\text{Speed (mph)} \times 1.47) \quad (9)$$

The average travel time is the average of the travel times for each direction. The computer program SIGPROG has an option of using either distance and speed as input or to simply use travel time.

Speed is indicated by the difference in slope of the progressive band from a 45-degree angle in Figure 2. This differs from the slope used in the conventional method. The per cent offset shift is the difference between travel time in either direction and the average for the section divided by the cycle length:

$$\text{Offset } (\%) = (TT_i - TT_a) / C = (TT_o - TT_a) / C \quad (10)$$

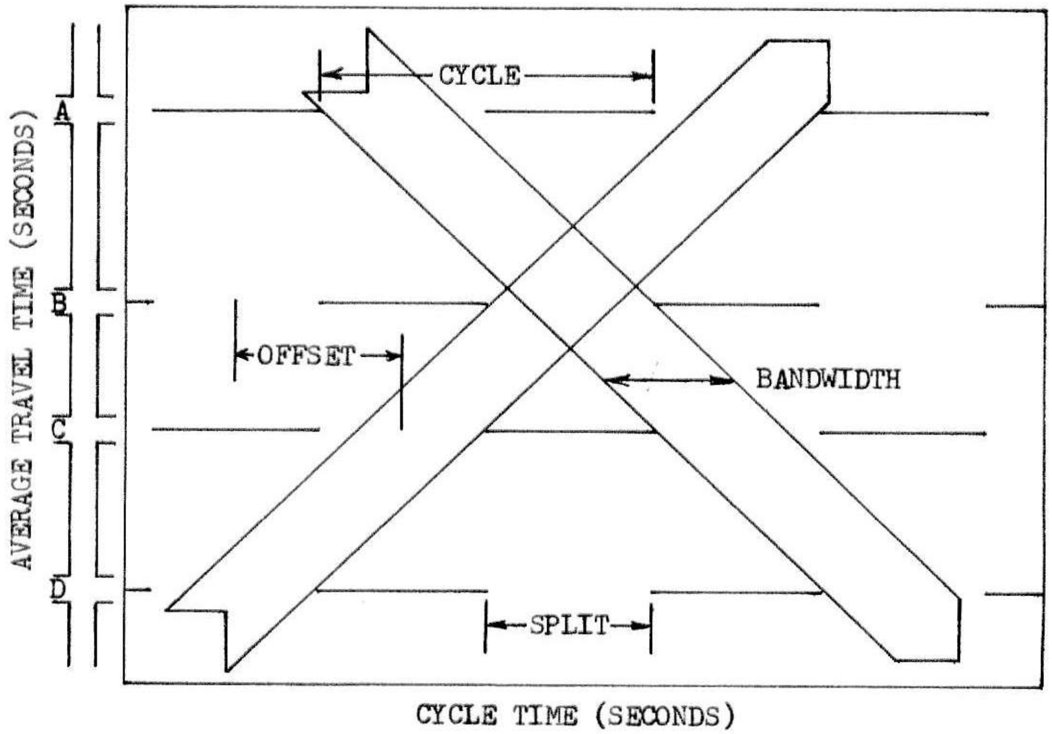

Figure 2. Modified time-space diagram.[1]

The offset of all signals on one side of each section having different directional travel times must be shifted an amount equal to the difference between the travel time for the section and the directional average travel time for the section. This shift is shown in equation (9).

After the time-travel time diagram has been determined, the optimum or satisfactory linear progressive bands for both directions can be solved manually by the usual trial-and-error, graphical, or mathematical techniques, or by a computer using the appropriate algorithms.

Input

The input for the computer program consists of an organization identification card, 12 general control cards, and a series of signal cards.[2] The program is written in Fortran IV, and is run on the IBM 360 with a minimum of 64K storage capacity. The object deck and sample input and output are listed in Appendix D.

The Organization Identification Card gives the name of the organization using the program. Only one of these cards is used, even if multiple runs are specified. Up to 32 characters may be used for each of two lines of identification, with the right-most position of each line in columns 40 and 80 in the ID card.

The General Control Cards follow the ID card, with the first nine preceding the signal cards and the last three following the signal cards. Card 1 contains the name of the system and is used to identify the output and also the problem when several runs are being made at one time. Columns 41 and 42 must contain the number of signals in the system, while the system name begins in column 45 and can contain up to 36 characters.

Card 2 contains the subtitle of the run, which must begin in column 41 and thus may contain up to 40 characters. It is used to identify the time of day and other conditions of the run, and will also be included in the output.

Card 3 identifies the number of signals in the system, with at least two signals but no more than 99 may be included. The units position of the number of signals must be located in column 45.

Card 4 indicates the minimum and maximum cycle lengths to be considered, with the units positions to be located in columns 45 and 55 respectively. If the cycle length is to be held constant, it should be coded as both the minimum and maximum.

Card 5 contains the suggested maximum speed tolerance from the desired progressive speeds specified on the signals cards. If coded other than 0 the program will find the optimum cycle from all possible cycles, within the limits specified. No tolerance will be allowed if it is coded 0, and only cycle lengths which are multiples of 5 seconds will be considered. For this case, the minimum and maximum cycle lengths on card 4 must also be multiples of 5 seconds. The units position is in column 45.

Card 6 is used to establish the proper offset relationship for signals common to more than one system. It is coded 0 in initial runs and coded with the desired offset transposition in subsequent runs, when desired. The units position of the offset is in per cent, and must be in column 45.

Card 7 contributes nothing to the program, but may not be omitted. It aids in reviewing a listing of the input cards.

Card 8 identifies the two general directions in which the traffic moves along the system, such as INBOUND, OUTBOUND, N-BOUND, or even SE-BOUND. The directions must begin in columns 41 and 51, respectively, and may use up to eight characters each.

Card 9 signifies the desired proportionment of the bandwidths by a percentage (such as 60-40) or a ratio (such as 2-1). Directional traffic volumes may also be used to specify the proportionment. In favoring the bandwidth in one direction one of three things will be accomplished: (a) if the proportion is realistically attainable, it will be obtained; (b) if the bandwidth in the preferred direction reaches its maximum possible width before the desired proportion is reached, no further adjustment will be made and the maximum bandwidth condition will be indicated by asterisks on the printout; and (c) if the bandwidth in the unfavored direction becomes small enough to no longer be meaningful (5 seconds or less), the maximum attainable bandwidth in the favored direction will be selected. The units positions must be in columns 45 and 55 for the two proportionments.

Card 10 contains the processing instruction RUN, to process the full program using the cycle length found in the scanning process to have the highest efficiency, or SCAN, to scan the cycle range in order to find the peaks in the efficiency curve only. The last letter of the message must be placed in column 45.

Card 11 is used to produce an intermediate deck of output cards containing the parameters of the optimum timing plan. This deck is used by a TMSPAC routine to plot a time-space diagram on a Cal-Comp plotter. A YES message produces the deck only if the RUN option was used in card 10. The last letter either YES or NO should be in column 45.

Card 12 contains a message to inform the computer whether or not additional runs follow. A YES indicates another run follows, while NO causes the program to terminate after the current run. If a YES was punched in card 11, a NO must be punched in card 12 due to the fact that only one run at a time can be input to create a plot deck. The last letter of the message is punched in column 45.

The signal cards contain a series of either ABCD or ABE cards, followed by another series of the same kind. The last series in the set will contain only AB. This set of cards is located between control cards 9 and 10 in the deck. Card E which contains the desired directional travel times is complementary to Cards C and D. The reason for this is explained by equation (1) and its explanation.

Card A contains the name of the signal location, which may occupy up to 12 characters and must begin in column 61, and the percent phase split to be devoted to the system, whose units position is in column 77.

Card B contains the minimum green time requirements for the cross street and the required clearance interval for the traffic in the system. The units positions are located in columns 67 and 77, respectively.

Card C contains the distance in feet between two adjacent signals for both directions, and is measured from the signal described on Card A to the signal on the next Card A. The units positions are in columns 45 and 55 and follow the directions on Card 8. If the distance in both directions is equal, the second field may be left blank and the computer will make the other entry.

Card D contains the desired directional progressive speeds in mph from the signal on Card A to the signal on the next Card A. The units positions must be in columns 45 and 55. If the speed in both directions is identical, the second field may be left blank and the computer will make the other entry.

Card E contains the desired directional travel times in seconds from the signal on Card A to the signal on the next Card A. If Card E is used, Card C and D must be omitted. The units positions of the travel times must be located in columns 45 and 55. It the directional travel times are identical, the second field may be left blank and the computer will make the second entry.

The cards should be placed in the following order with the control cards as shown:

```
//          JOB (FF18C4Q7,10,10),'NAME',MSGLEVEL=1
//          EXEC FORTGLC
//FORT.SYSIN DD *

[Program deck goes here]

/*
//GO.SYSIN DD *

[Organization Identification Card goes here
 Data goes here]

/*
```

Output

The output is in the form of three printed tables, which are shown in Appendix D. The first table is a list of the entire input card deck, which rapidly reveals mispunched or miscoded information. It serves primarily as reference convenience for the other two tables.

The second table contains the results of an incremental cycle scan between the minimum and maximum cycle lengths to find the maximum efficiency obtainable. The efficiency is defined by the program as EFF = 100 x BANDWIDTH / CYCLE. A plot of the maximum efficiency obtainable at each cycle length is included. The cycle having the highest efficiency is identified, and improvements in the efficiency by an iterative process are included.

If RUN is contained on Card 10, the third table is printed using the cycle having the highest efficiency. Adjacent to the name of each signal is its offset and green and amber intervals. The offset is given to the beginning, middle, and end of the phase. The cycle length is given above for the optimum, best, next 5 second interval below, and next 5 second interval above and is followed by bandwidths, efficiencies, and progressive speeds for each of those four conditions. If Card E is used in place of Card C and D, the values in the signal spacing column are in terms of seconds instead of feet, and the progressive speeds are in terms of seconds instead of mph. The progressive band offsets are given on the same line and to the left of the progressive speeds. If the progressive band utilizes the full green interval at any signal, an asterisk will precede the progressive band offsets on this printout. The plan number is given at the upper left. It is the same as the optimum cycle with the requested bandwidth proportions in parentheses.

TMSPAC

If YES is punched in Card 11 and RUN was selected on Card 10, a deck for the Time-Space Diagram Plot Program (TMSPAC) will be punched.

This deck contains two identification cards and one additional card for each signal. The first identification card contains the title and subtitle as contained on Cards 1 and 2 of the input deck. The second card contains the input of Card 8, the optimum cycle length, the two directional bandwidths, and the two directional efficiencies.

Each remaining card contains the name of the signal location, the offsets to the beginning, middle, and end of the green interval, the phase split, the offsets to the beginning and ending edges of the progressive band in direction 1, the average distance to the next signal, the progressive speed to the next signal, the offsets to the beginning and ending edges of the progressive band in direction 2, the progressive speed from the next signal, and the code 1 or 2 to signify that distance and speed are in units of feet and mph or in units of seconds and seconds, respectively. The last card is identified by zero distance and speeds.

Sample Problems

The program was first run for the sample data obtained from Anderson Avenue and Poyntz Avenue in Manhattan, Kansas. The input data for Anderson Avenue are listed in Table 5, and the input data for Poyntz Avenue are listed in Table 6. Since this program converts the distance and desired speed to travel times, it was decided to vary the desired speed in several runs to observe the effects of such a variation. Since the speed limit on Anderson Avenue is 30 mph, the speed was varied by 1 mph from 25 mph to 30 mph and also included a run with 20 mph. Cycle length varied from 35 to 65 seconds, and the results are listed in Table 7.

Table 5.--Anderson Avenue Data

Street Name	Dist to Next St	Green Phase	Ped. Xing	Clear Inter
Sunset	---	56	15	3
Denison	953	56	15	3
Seventeenth	639	56	20	3
Crosswalk	1003	68	15	3
Fourteenth	395	44	15	3

These results show that the efficiency does vary for each increase in 1 mph of the desired progressive speed. With the exception of the 30 mph case, it can be seen that the proper cycle length appears to be either 50 or 55 seconds. Although there is quite a bit of variation between the efficiencies of the two at each speed, it is also extremely difficult to be able to predict speeds within 1 mph. For this reason, a cycle length of 50 or 55 seconds should be selected.

Table 6.--Poyntz Avenue Data

Street Name	Dist to Next St	Green Phase	Ped. Xing	Clear Inter
Seventeenth	---	55	15	3
Fourteenth	1380	62	15	3
Eleventh	1420	62	15	3
Crosswalk	680	57	15	3
Juliette	1180	60	15	3
Fifth	940	62	15	3
Fourth	460	60	15	3
Third	460	55	15	3
Second	460	55	15	3

Table 7.--Anderson Avenue Efficiencies

Cycle	Speed (in mph)						
	20	25	26	27	28	29	30
35	21.026	13.273	12.405	13.501	17.457	18.534	17.649
40	12.085	22.682	21.646	20.687	16.506	12.229	10.068
45	19.717	16.788	12.040	14.660	17.462	20.070	22.505
50	26.932	17.745	20.678	23.394	25.916	24.859	22.364
55	20.806	25.223	26.451	23.640	21.029	17.474	15.099
60	15.568	23.364	20.418	16.217	15.610	15.180	17.841
65	22.063	16.266	15.613	15.247	19.073	21.705	24.161
Opt	27.151	27.374	27.161	26.190	26.738	26.526	26.423
Cyc	49.481	56.629	54.451	53.002	50.562	48.818	66.998

The results of the Poyntz Avenue problem indicate that the best cycle length is probably 70 or 75 seconds. This is substantially greater than the 60 second cycle length currently in use. Two speeds are indicated in Table 8, because the speed limit is 20 mph between Second Street and Juliette Avenue and 30 mph from Juliette Avenue through Seventeenth Street. This change in speed can be altered on each Card D in the input deck.

Table 8.--Poyntz Avenue Efficiencies

Cycle	Speed (in mph)					
	15/25	16/26	17/27	18/28	19/29	20/30
40	17.727	22.823	18.032	16.315	12.661	14.130
45	18.032	13.026	10.674	12.113	17.486	10.255
50	12.408	13.489	13.090	15.971	16.896	13.699
55	15.885	17.730	17.891	15.130	22.033	19.017
60	15.229	16.730	21.752	19.706	13.744	20.682
65	20.717	22.728	14.865	17.687	21.983	15.770
70	19.926	12.909	26.008	18.138	22.307	25.617
75	18.819	24.911	17.645	25.738	25.509	22.138
80	24.130	20.036	25.811	25.530	20.197	15.380
Opt	26.947	26.907	26.986	26.608	26.002	25.900
Cyc	78.253	73.881	79.283	75.486	73.002	68.923

Table 9.--Topeka Avenue Data

Street Name	Dist to Next St	Prog Speed	Green Split			Prog Speed	Green Split
			B	M	S		
Third	---	25	41	41	41	22	52
Fourth	555	25	59	59	59	22	45
Fifth	555	25	51	51	51	22	60
Sixth	555	25	65	65	65	22	55
Eighth	1135	25	60	60	60	22	45
Tenth	1135	25	65	65	65	22	58
Twelfth	1188	25	53	46	50	22	50

The program was again run with the data obtained from Topeka Avenue in Topeka, Kansas, as shown in Table 9. The distances are given in feet, the speeds in mph, and the green phase split in per cent of cycle. The pedestrian crossing is 15 seconds for all streets, and the clearance interval is 3.25 seconds for all intersections. Splits are given for four cases: (1) the normal balance period, B; (2) the morning rush period, M; (3) certain spot periods during the day, S; and (4) the evening rush period. The first three cases have a desired progressive speed of 25 mph, while the evening rush has a desired speed of 22 mph.

The difficulty in this problem arises from single left turn phases at the first three intersections. At Third Street on Topeka Avenue, southbound traffic has a single "left turn only" phase with no other movements in any direction. Northbound Topeka Avenue traffic has a "left turn only" phase at Fourth Street, and southbound traffic again has a "left turn only" phase at Fifth Street. These "left turn only" phases were treated as all red phases, so that the green times for Third, Fourth, and Fifth consider only the all green phases on

Topeka Avenue. After the results are obtained, the "left turn only" phase should then be given its per cent of time from the red phase for Topeka Avenue. The results for these data are given in Table 10. It indicates that during the first three periods, that a cycle length of 68 seconds is optimum, and that during the evening rush period that a cycle length of 75 seconds is optimum. Topeka is currently using 65 and 70 seconds for the two, which is slightly under the optimum values.

Table 10.--Topeka Avenue Efficiencies

Cycle	Bal	Morn	Spot	Even
50	14.391	10.891	12.891	10.339
55	21.570	20.079	21.570	9.547
60	19.811	19.811	19.811	7.664
65	21.713	21.713	21.713	19.962
70	21.980	21.980	21.980	19.285
75	17.776	15.848	16.276	21.100
80	14.858	14.085	14.858	19.938
85	16.626	13.559	15.559	10.230
90	18.207	18.207	18.207	9.480
95	17.512	17.512	17.512	11.052
100	16.886	16.886	16.886	11.326
Optimum	22.479	22.479	22.479	21.154
Cycle	68.002	68.002	68.002	74.784

SIGOP

An off-line computer program was developed in 1966 by Peat, Marwick, Livingston Company under contract to the Bureau of Public Roads which will solve traffic progression problems for an entire network rather than a single street.[6] The purpose of this program was to determine optimum cycle lengths, phase splits and offsets for traffic signals in a grid network of not more than 150 signalized intersections.

The program begins by computing the phase split for each intersection for a specified cycle length. The length of each phase is then set proportional to the total flow or critical flow for each phase. The total flow is the sum of the observed traffic for all lanes and all approaches during a given phase, while the critical flow is the maximum observed per lane flow through an intersection for a given phase.

Offsets are calculated to minimize a given linear combination of vehicular delay and stops. It is also possible to minimize system costs. Under the assumption that on the average the traffic flow pattern repeats itself on successive cycles, vehicular delay is calculated for each approach. The sum of the delays for all vehicles in the system is measured, and stops are calculated as the total number of stops over one cycle on a given approach.

SIGOP is capable of optimizing up to twelve time periods in one computer run. Morning peak, evening peak, and off-peak periods are most commonly used. This allows for separate traffic signal settings for each period.

The SIGOP program consists of six program blocks written in Fortran IV programming language.

The first block is the INPUTS block which reads and verifies the basic data supplied by the user and checks the data for logic and consistency. This is followed by the PHASES program which computes and tabulates phase splits for each intersection for given trial cycle lengths. The OFFSET program then computes and tabulates the ideal difference in offsets for each network link. This program also computes and tabulates link weights to be used in the optimization procedure.

The OPTMIZ program produces a set of signal offsets for the network which most closely approach the ideal offsets computed for each individual network link. The VALUAT program uses a simplified network traffic simulation to calculate the system propensity to generate delay, stops, and cost for a given set of cycle lengths, phase splits, and offsets. The OUTPUT program presents recommended phase splits and offsets, and prints time-space charts.

The input data requirements have been devised to present traffic engineers with a wide range of flexibility in making decisions. But due to the complexity of the program, perhaps a more solid foundation would be helpful. Intersection data such as number of phases, amber times, and minimum green times are relatively easy to obtain, but determining arrival and discharge headways is more difficult. Similarly volumes, turning movements, speeds, and mode distributions can be easily determined. However, there is a street weighting factor which can cause a lot of trouble.

This factor assigns a weight to certain important streets to allow them to have precedence in the optimization. With no feeling for this factor, the engineer can easily make mistakes.

SIGOP has been used on a trial basis by six cities in this country. Kansas City was the first to finish the trial period of the six that were selected. The other five are Seattle, Washington; San Antonio, Texas; Indianapolis, Indiana; Cincinnati, Ohio; and Miami, Florida. Kansas City felt that the SIGOP program was a useful tool to the traffic engineer. However, the engineers in that city selected link groups from only two-way streets, which proved to be a poor decision. In the collection of the "after" data from the CBD area of downtown Kansas City, Missouri, the results showed that the new offsets, phase splits, and cycle lengths did not improve the system efficiency to any measurable degree. This is possibly due to (1) the limited space available in the CBD, and (2) that the system was operating close to the optimum value due to the many manual calculations that had previously been made.

The SIGOP program does provide a method for developing network timing patterns which is much more economical and faster than any other that has been previously developed. This program allows for improvement of capacity and a decrease in delay without any physical changes in the system, such as street widening. The program is quite lengthy in time due to the tremendous amount of data that must be collected prior to a computer run. But there is no doubt that a program which evaluates an entire network is a great improvement over programs which solve only single street systems.

CONCLUSION

Four different approaches to signal progression problems have been studied in this report. The first three treat only a single arterial, while SIGOP studies an entire network. With the complexity of the SIGOP program, a restriction on time, and since the program is still in the development stages, this program was only explained but not actually run on a computer by this author.

The comparison of three methods for solving progression along an arterial shows advantages for each method. Those three methods are Little's method for either the 1620 (TSS 3) or the 360 (EMTSS 3), the program developed by Chamberlain, and SIGPROG. All three approaches use the number of signals and distances between adjacent intersections as inputs, but variations such as NSIG-1 are used. Distance can be treated as distance between each set of intersections or as the distance from the origin. Therefore, many variations of these programs are possible. A comparison of the three approaches is found in Table 11 below.

Table 11.--Comparison of Methods

	Signals	Cycle	CMIN-CMAX	Distance	Speed	VMIN-VMAX	Green Time	Red Phase	Volumes	Headway	Bandwidth Proportion	Clearance Interval
Little	x	x		x	x			x	x	x		
Chamberlain	x		x	x		x	x		x			
SIGPROG	x		x	x	x		x				x	x

A comparison of Little's method with Chamberlain's method seems to reveal a slight advantage of Chamberlain's method. The main reason for this is that in Little's method the cycle length and directional speeds must be held constant, while Chamberlain uses a range of values for these two inputs from some minimum to a higher maximum. Little uses the red phase for computing offsets while Chamberlain uses green time, and there seems to be little difference in the two. Little uses a headway calculation, which neither Chamberlain or SIGPROG use. Chamberlain's method of output is much easier to read than Little's, but takes longer to run on the computer.

SIGPROG uses a bandwidth proportion in its input which differs somewhat from the volumes used in the other two methods. The other two use volume data, whereas SIGPROG uses a ratio for the two directions which may or may not be actual volume counts. Chamberlain's method provides a little more versatility than SIGPROG in that it uses VMIN and VMAX, while SIGPROG uses only a desired speed and not an actual range. SIGPROG has a big advantage over the other two methods with its output. This program prints out the original data, plots the efficiency for each cycle tried, and gives much more individual information concerning the offsets and bandwidths at each intersection.

Identical data were run for Little's program TSS 3 (1620) and also for EMTSS 3 (360) as reported earlier. Both results gave the same answers, so there seems to be no difference in the two computer programs developed by Little. The data run on Chamberlain's method were from Anderson Avenue in Manhattan, Kansas. It showed that the best cycle was 70 seconds. Similar runs on SIGPROG gave a best cycle length of 65 seconds. However, the Chamberlain method requires some volume inputs for each direction, and in the sample used here figures of 400 vph were used. These figures were not obtained from any counts, but were merely estimated by the author. Therefore the reliability of the results are very questionable.

Many other concepts have been developed to use the signal progression idea. Some other methods include Webster's cycle and split optimization method, the mixed-cycle method, the preferential street method, and Allsop's graph theory method.[9] A study done by Wagner, Barnes, and Gerlough in California found that all methods investigated can improve traffic operation. The results of their study showed that it makes little difference which method should be used. The important thing is to persuade all cities to use one method of their choice, and not merely let the problem lie untouched. Their results also showed that the simple methods are nearly equal in operational improvements to the complex methods. Therefore, a method should be selected which the user can understand, for which he is able to collect data, and for which he is able to use the results.

The author favors SIGPROG only for its completeness of results. Chamberlain's method also presents acceptable results, but not as complete as SIGPROG. Little's method is the easiest to use. Any of the methods will give good results and the one which can be obtained easiest should be used. Further developments need to be made in network optimization done by SIGOP. It is still very costly, and the data collection is very time consuming. However, it is the only method presently developed for network analysis.

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

APPENDIX A - LITTLES METHOD ON 1620

```
C C
C      PROGRAM TSS 3 BANDWIDTH AND OFFSET COMPUTATIONS
C      NEW SYNCRONIZATION TECHNIQUE DEVELOPED BY J.D.C. LITTLE
C      BRIAN V MARTIN JANUARY 1964
C      GREGORY W. HARDIN   KANSAS STATE   SPRING SEMESTER 1971
    1 FORMAT(34H              N      CYCLE       HEADWAY)
    2 FORMAT(33H                     (SEC)        (SEC))
    3 FORMAT(47H    INBOUND      OUTBOUND     POSSIBLE     POSSIBLE)
    4 FORMAT(48H     VOLUME        VOLUME     VOLUME IN    VOLUME OUT)
    5 FORMAT(45H     (VPH)         (VPH)        (VPH)          (VPH))
    6 FORMAT(39H   BANDWIDTH    BANDWIDTH         CRITICAL)
    7 FORMAT(38H    INBOUND      OUTBOUND          SIGNAL)
    8 FORMAT(21H     (SEC)         (SEC))
    9 FORMAT(51H    DISTANCE    RED SPLIT    RHS OF RED START OF CYCLE)
   10 FORMAT(49H     (FEET)        (SEC)         (SEC)        (FRACTION))
   11 FORMAT(48H   FRONT EDGE   REAR EDGE    TOTAL TIME   TOTAL TIME)
   12 FORMAT(47H    OUT BAND     IN BAND       INBOUND      OUTBOUND)
   13 FORMAT(48H   (FRACC.L.)   (FRACC.L.)   (FRACC.L.)   (FRACC.L.))
   14 FORMAT(10H              )
      DIMENSION X(50),RED(50),SPEDI(50),SPEDO(50),Y(50),Z(50),W(50)
      DIMENSION PHASE(50),TIME(50)
C      INPUT ROUTINE
   15 READ,NSIG,CYCLE,VOLIN,VOLOT,HEDWY
      DO 16 I=1,NSIG
      READ,X(I),B
   16 RED(I)=B/CYCLE
   17 PLATI=VOLIN*HEDWY/3600.
      PLATO=VOLOT*HEDWY/3600.
      NSAVE=NSIG-1
      DO 18 I=1,NSAVE
      READ,A,B
      SPEDI(I)=.6818182/A
   18 SPEDO(I)=.6818182/B
C      START COMPUTATIONS
      Y(1)=0.
      Z(1)=0.
      TIME(1)=0.
      B=0.5/CYCLE
      DO 19 I=2,NSIG
      M=I-1
      A=(X(I)-X(M))*B
      SAVE=A*(SPEDO(M)+SPEDI(M))
      TIME(I)=TIME(M)+SAVE
      Y(I)=Y(M)-.5*(RED(I)-RED(M))+SAVE
   19 Z(I)=Z(M)+A*(SPEDO(M)-SPEDI(M))
C      START COMPUTATION FOR EQUAL BANDWIUTH
      BAND=0.
      DO 31 I=1,NSIG
      WMIN=3600.
```

```
      DO 29 J=1,NSIG
      A=Y(J)-Y(I)
      B=A-0.5
      NSAVE=A
      SAVE=NSAVE
      A=A-SAVE
      IF(A) 20,21,21
   20 A=A+1.
   21 NSAVE=B
      SAVE=NSAVE
      B=B-SAVE
      IF(B) 22,23,23
   22 B=B+1.
   23 IF(A-B) 24,25,25
C     LIGHTS IN PHASE FOR MAXIMUM BANDWIDTH
   24 PHASE(J)=0.
      W(J)=1.-A
      GO TO 26
C     LIGHTS OUT OF PHASE FOR MAXIMUM BANDWIDTH
   25 PHASE(J)=.5
      W(J)=1.-B
   26 B=W(J)-RED(J)
C     IS CURRENT BANDWIDTH BETTER THAN PREVIOUS BEST BANDWIDTH
      IF(B-BAND) 31,27,27
   27 IF(WMIN-B) 29,29,28
   28 WMIN=B
   29 CONTINUE
      BAND=WMIN
      LTBST=I
C     SAVE BEST PHASE(J) IN SPEDI(K) AND BEST W(J) IN SPEDO(K)
      DO 30 K=1,NSIG
      SPEDI(K)=PHASE(K)
   30 SPEDO(K)=W(K)
   31 CONTINUE
C     COMPUTE FINAL OFFSETS FOR MAXIMUM EQUAL BANDWIDTH AND STORE
C     IN PHASE(I)
      B=Z(LTBST)
      DO 33 I=1,NSIG
      A=Z(I)-B+SPEDI(I)
      NSAVE=A
      SAVE=NSAVE
      A=A-SAVE
      IF(A) 32,33,33
   32 A=A+1.
   33 PHASE(I)=A
C     FIND LIGHT WITH MINIMUM GREEN(I.E. FIND MAX RED)-CALL IT
C     WMIN
      WMIN=0.
      DO 35 I=1,NSIG
      IF(RED(I)-WMIN) 35,35,34
   34 WMIN=RED(I)
      NSAVE=I
```

```
   35 CONTINUE
      A=1.-RED(NSAVE)
      B=2.*BAND
      SAVE=PLATI+PLATO
C     ASSUME OUTBAND PLATOON IS THE LARGEST
      WMIN=PLATO
C     WHICH PLATOON LENGTH IS IN FACT THE LARGEST
      IF(PLATI-PLATO) 38,36,37
C     PLATOON LENGTHS ARE EQUAL
   36 BIN=BAND
      BOUT=BAND
      GO TO 54
C     CORRECT WMIN SINCE INBOUND PLATOON LENGTH IS LARGEST
   37 WMIN=PLATI
C     FIND BAND WHERE BOUT EQUALS BAND IN DIRECTION OF LARGEST
C     PLATOON
   38 BOUT=B*WMIN/SAVE
      IF(BOUT-WMIN) 39,39,40
   39 BOUT=WMIN
      IF(BOUT-B) 40,41,41
   40 IF(BOUT-A) 42,41,41
   41 BOUT=A
   42 BIN=B-BOUT
      IF(BIN) 43,44,44
   43 BIN=0.
C     WHICH PLATOON LENGTH IS THE LARGEST
   44 IF(PLATI-PLATO) 48,48,45
C     SINCE INBOUND PLATOON IS LARGEST, SWITCH BOUT AND BIN
   45 WMIN=BIN
      BIN=BOUT
      BOUT=WMIN
      DO 47 I=1,NSIG
      A=BIN+RED(I)-SPEDO(I)
      IF(A) 46,47,47
   46 A=0.
   47 SPEDI(I)=A
      GO TO 51
   48 DO 50 I=1,NSIG
      A=BOUT-BAND-1.+SPEDO(I)
      IF(A) 49,50,50
   49 A=0.
   50 SPEDI(I)=A
C     COMPUTE FINAL OFFSETS FOR UNEQUAL BANDWIDTHS
   51 B=PHASE(LTRST)-SPEDI(LTRST)
      DO 53 I=1,NSIG
      A=PHASE(I)-SPEDI(I)-B
      NSAVE=A
      SAVE=NSAVE
      A=A-SAVE
      IF(A) 52,53,53
   52 A=A+1.
   53 PHASE(I)=A
```

```
C     OUTPUT ROUTINE
   54 PLATI=BIN*3600./HEDWY
      PLATO=BOUT*3600./HEDWY
      BIN=BIN*CYCLE
      BOUT=BOUT*CYCLE
      PUNCH 1
      PUNCH 2
      PUNCH,NSIG,CYCLE,HEDWY
      PUNCH 14
      PUNCH 3
      PUNCH 4
      PUNCH 5
      PUNCH,VOLIN,VOLOT,PLATI,PLATO
      PUNCH 14
      PUNCH 6
      PUNCH 7
      PUNCH 8
      PUNCH,BIN,BOUT,LTBST
      PUNCH 14
      PUNCH 9
      PUNCH 10
C     WMIN EQUALS THE TIME FROM THE ORIGIN TO THE RIGHT HAND SIDE
C     OF RED FOR EACH LIGHT.  THE ORIGIN IS THE CENTER OF RED FOR
C     THE REFERENCE LIGHT,LTBST.  ALL OUTPUT IS IN SECONDS EXCEPT
C     FOR X(I), WHICH IS IN FEET.
      DO 57 I=1,NSIG
      SAVE=RED(I)
      WMIN=(PHASE(I)+.5*SAVE)*CYCLE
      IF(WMIN-CYCLE) 56,56,55
   55 WMIN=WMIN-CYCLE
   56 SAVE=SAVE*CYCLE
   57 PUNCH,X(I),SAVE,WMIN,PHASE(I)
      PUNCH 14
      A=TIME(LTBST)+Z(LTBST)
      B=.5*RED(LTBST)
      ALHS=B-A
   58 IF(ALHS) 59,60,60
   59 ALHS=ALHS+1.
      GO TO 58
   60 A=TIME(NSIG)-Z(NSIG)
      SAVE=TIME(LTBST)-Z(LTBST)
      BRHS=SAVE-B
   61 IF(BRHS-A) 62,63,63
   62 BRHS=BRHS+1.
      GO TO 61
   63 B=TIME(NSIG)+Z(NSIG)
C     ALHS INDICATES POSITION OF FRONT EDGE OF OUTBOUND BAND AT
C     FIRST SIGNAL, IN TERMS OF CYCLE LENGTHS FROM ORIGIN.  BRHS
C     INDICATES POSITION OF REAR EDGE OF INBOUND BAND AT FIRST
C     SIGNAL, IN TERMS OF CYCLE LENGTHS FROM ORIGIN.  A AND B
C     INDICATE TOTAL SYSTEM TRAVEL TIME, INBOUND AND OUTBOUND
C     RESPECTIVELY, IN TERMS OF CYCLE LENGTHS.
```

```
PUNCH 11
PUNCH 12
PUNCH 13
PUNCH,ALHS,BRHS,B,A
PUNCH 14
READ,NSAVE,CYCLE
IF(NSAVE) 17,17,15
END
```

4	70.0	990.	1021.	2.
0.	32.2			
380.	24.5			
755.	26.6			
1130.	24.5			
35.0	35.0			
35.0	35.0			
35.0	35.0			
1	1.			

N	CYCLE (SEC)	HEADWAY (SEC)
4	7.0000E+01	2.0000

INBOUND VOLUME (VPH)	OUTBOUND VOLUME (VPH)	POSSIBLE VOLUME IN (VPH)	POSSIBLE VOLUME OUT (VPH)
9.9000E+02	1.0210E+03	3.7904E+01	9.7200E+02

BANDWIDTH INBOUND (SEC)	BANDWIDTH OUTBOUND (SEC)	CRITICAL SIGNAL
1.4740	3.7800E+01	1

DISTANCE (FEET)	RED SPLIT (SEC)	RHS OF RED (SEC)	START OF CYCLE (FRACTION)
0.0000	3.2200E+01	1.6100E+01	0.0000
3.8000E+02	2.4500E+01	2.3503E+01	1.6075E-01
7.5500E+02	2.6600E+01	3.0808E+01	2.5011E-01
1.1300E+03	2.4500E+01	3.0413E+01	2.5947E-01

FRONT EDGE OUT BAND (FRACC.L.)	REAR EDGE IN BAND (FRACC.L.)	TOTAL TIME INBOUND (FRACC.L.)	TOTAL TIME OUTBOUND (FRACC.L.)
2.3000E-01	7.7000E-01	3.1447E-01	3.1447E-01

9	60.0	400.	400.	2.
0.	27.0			
1380.	23.0			
2800.	23.0			
3480.	26.0			
4660.	24.0			
5600.	23.0			
6060.	24.0			
6520.	27.0			
6980.	27.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
20.0	20.0			
20.0	20.0			
20.0	20.0			
20.0	20.0			
1	1.			

N	CYCLE (SEC)	HEADWAY (SEC)
9	6.0000E+01	2.0000

INBOUND VOLUME (VPH)	OUTBOUND VOLUME (VPH)	POSSIBLE VOLUME IN (VPH)	POSSIBLE VOLUME OUT (VPH)
4.0000E+02	4.0000E+02	4.6227E+02	4.6227E+02

BANDWIDTH INBOUND (SEC)	BANDWIDTH OUTBOUND (SEC)	CRITICAL SIGNAL
1.5409E+01	1.5409E+01	5

DISTANCE (FEET)	RED SPLIT (SEC)	RHS OF RED (SEC)	START OF CYCLE (FRACTION)
0.0000	2.7000E+01	1.3500E+01	0.0000
1.3800E+03	2.3000E+01	4.1500E+01	5.0000E-01
2.8000E+03	2.3000E+01	1.1500E+01	0.0000
3.4800E+03	2.6000E+01	4.3000E+01	5.0000E-01
4.6600E+03	2.4000E+01	1.2000E+01	0.0000
5.6000E+03	2.3000E+01	4.1500E+01	5.0000E-01
6.0600E+03	2.4000E+01	4.2000E+01	5.0000E-01
6.5200E+03	2.7000E+01	1.3500E+01	0.0000
6.9800E+03	2.7000E+01	1.3500E+01	0.0000

FRONT EDGE OUT BAND (FRACC.L.)	REAR EDGE IN BAND (FRACC.L.)	TOTAL TIME INBOUND (FRACC.L.)	TOTAL TIME OUTBOUND (FRACC.L.)
4.3485E-01	3.5652	3.0833	3.0833

8	60.0	370.	354.	2.
0.	36.0			
1600.	18.0			
1865.	18.0			
2675.	18.0			
3750.	21.0			
4285.	18.0			
4800.	9.0			
5520.	27.6			
30.0	30.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
1	1.			

N	CYCLE (SEC)	HEADWAY (SEC)
8	6.0000E+01	2.0000

INBOUND VOLUME (VPH)	OUTBOUND VOLUME (VPH)	POSSIBLE VOLUME IN (VPH)	POSSIBLE VOLUME OUT (VPH)
3.7000E+02	3.5400E+02	6.3208E+02	6.0474E+02

BANDWIDTH INBOUND (SEC)	BANDWIDTH OUTBOUND (SEC)	CRITICAL SIGNAL
2.1069E+01	2.0158E+01	1

DISTANCE (FEET)	RED SPLIT (SEC)	RHS OF RED (SEC)	START OF CYCLE (FRACTION)
0.0000	3.6000E+01	1.8000E+01	0.0000
1.6000E+03	1.8000E+01	3.9000E+01	5.0000E-01
1.8650E+03	1.8000E+01	3.8544E+01	4.9241E-01
2.6750E+03	1.8000E+01	9.0000	0.0000
3.7500E+03	2.1000E+01	4.0500E+01	5.0000E-01
4.2850E+03	1.8000E+01	3.9000E+01	5.0000E-01
4.8000E+03	9.0000	4.5000	0.0000
5.5200E+03	2.7600E+01	1.3800E+01	0.0000

FRONT EDGE OUT BAND (FRACC.L.)	REAR EDGE IN BAND (FRACC.L.)	TOTAL TIME INBOUND (FRACC.L.)	TOTAL TIME OUTBOUND (FRACC.L.)
3.0000E-01	2.7000	2.0909	2.0909

APPENDIX B - LITTLES METHOD ON 360

```
//          JOB (FF18C4G7,10,10),-HARDIN-,MSGLEVEL=1
//          EXEC FORTGCLG
//FORT.SYSIN DD *
C     PROGRAM EMTSS3 BANDWIDTH AND OFFSET COMPUTATIONS
C     SYNCHRONIZATION TECHNIQUE DEVELOPED BY J.D.C. LITTLE
C     TRANSFORMED TO FORTRAN IV BY EDMOND T. MILLER DECEMBER 1964
C     PROGRAMMED FOR AN IBM 7094
      DIMENSION PHASE(50),TIME(50),X(50),Y(50),Z(50),W(50),SPEDI(50),RE
     1(50),SPEDO(50),NAME(18),SI(50),SO(50),NAMES(50,8),GO(50),GI(50)
      COMMON PHASE,TIME,X,Y,Z,W,SPEDI,SPEDO,RED,NAME,SI,SO,PLATO,PLATI,
     1LHS,RRHS,VOLIN,VOLOT,HEDWY,CYCLE,LTRST,NSIG,A,B,RIN,ROT,NAMES,HS,
     2S
      DATA KLANKS/4H    /
  100 FORMAT(18A4)
  101 FORMAT(I4,6F10.0,I4)
  102 FORMAT(2F10.0,7A4,A2)
  103 FORMAT(1H118A4)
  104 FORMAT(18H NUMBER OF SIGNALS8X1H=I7/13H CYCLE LENGTH13X1H=F8.2/16
     1 VEHICLE HEADWAY10X1H=F8.2/27H INBOUND VOLUME (INPUT)    =F8.2/27H
     2OUTBOUND VOLUME (INPUT)  =F8.2/27H POSSIBLE INBOUND VOLUME  =F8.2
     327H POSSIBLE OUTBOUND VOLUME =F8.2/18H INBOUND BANDWIDTH8X1H=F8.2
     419H OUTBOUND BANDWIDTH7X1H=F8.2/29H RESTRICTING SIGNAL IS NUMBERI
     55)
  105 FORMAT(1H030X74HSIGNAL  DISTANCE FROM  LENGTH OF   END  OF   OFF
     1ET   PROGRESSION  SPEEDS/10X11HSTREET NAME10X21HNUMBER   THE  ORI
     2IN 2(11H  RED PHASE),19X5H(MPH)/43X6H(FEET)5X51H(SECONDS)  (SECON
     3S)  (CYCLES)  INBOUND    OUTBOUND)
  106 FORMAT(1X7A4,A2,I4,2XF12.2,1X2F11.2,F10.2)
  107 FORMAT(30H0FRONT EDGE OF OUTBOUND BAND =F7.2,8H CYCLES,F8.2,8H SE
     1ONDS / 30H REAR EDGE OF INBOUND BAND   =F7.2,8H CYCLES,F8.2,8H SE
     2ONDS / 30H SYSTEM TRAVEL TIME INBOUND  =F7.2,8H CYCLES,F8.2,8H SE
     3ONDS / 30H SYSTEM TRAVEL TIME OUTBOUND =F7.2,8H CYCLES,F8.2,8H SE
     4ONDS)
  108 FORMAT(2F10.0)
  109 FORMAT(80X2F12.2)
    1 READ(5,100)(NAME(I),I=1,18)
      READ(5,101)NSIG,CYCLE,VOLIN,VOLOT,HEDWY,HS,TS,KEY
      READ(5,102)(X(I),RED(I),(NAMES(I,J),J=1,8),I=1,NSIG)
      DO 2 I=1,NSIG
    2 RED(I)=RED(I)/CYCLE
    3 PLATI=VOLIN*HEDWY/3600.
      PLATO=VOLOT*HEDWY/3600.
      NSAVE=NSIG-1
      IF(KEY.GT.0) GO TO 37
      READ(5,108)(SPEDI(I),SPEDO(I),I=1,NSAVE)
    4 DO 5 I=1,NSAVE
      GO(I)=SPEDO(I)
      GI(I)=SPEDI(I)
      SPEDI(I)=.68181818/SPEDI(I)
```

```
      SI(I)=SPEDI(I)
      SPEDO(I)=.68181818/SPEDO(I)
    5 SO(I)=SPEDO(I)
      Y(1)=0
      Z(1)=0
      TIME(1)=0
      B=0.5/CYCLE
      DO 6 I=2,NSIG
      M=I-1
      A=(X(I)-X(M))*B
      SAVE=(SPEDO(M)+SPEDI(M))*A
      TIME(I)=TIME(M)+SAVE
      Y(I)=Y(M)-.5*(RED(I)-RED(M))+SAVE
    6 Z(I)=Z(M)+A*(SPEDO(M)-SPEDI(M))
      BAND=0
      DO 11 I=1,NSIG
      WMIN=3600.
      DO 9 J=1,NSIG
      A=Y(J)-Y(I)
      B=A-0.5
      A=A-FLOAT(IFIX(A))
      IF(A.LT.0.0)A=A+1.
      B=B-FLOAT(IFIX(B))
      IF(B.LT.0.0)B=B+1.
      IF(A.GT.B)GO TO 7
      PHASE(J)=0
      W(J)=1.-A
      GO TO 8
    7 PHASE(J)=.5
      W(J)=1.-B
    8 B=W(J)-RED(J)
      IF(B.LT.BAND) GO TO 11
      IF(WMIN.GT.B) WMIN=B
    9 CONTINUE
      BAND=WMIN
      LTBST=I
      DO 10 K=1,NSIG
      SPEDI(K)=PHASE(K)
   10 SPEDO(K)=W(K)
   11 CONTINUE
      B=Z(LTBST)
      WMIN=0
      DO 12 I=1,NSIG
      A=Z(I)-B+SPEDI(I)
      A=A-FLOAT(IFIX(A))
      IF(A.LT.0.0)A=A+1.
      PHASE(I)=A
      IF(RED(I).LE.WMIN) GO TO 1
      WMIN=RED(I)
      NSAVE=I
   12 CONTINUE
      A=1.-RED(NSAVE)
```

```
      B=2.*BAND
      WMIN=PLATO
      IF(PLATI-PLATO)15,13,14
   13 BIN=BAND
      BOT=BAND
      GO TO 24
   14 WMIN=PLATI
   15 BOT=B*WMIN/(PLATI+PLATO)
      IF (BOT.GT.WMIN) GO TO 16
      BOT=WMIN
      IF(BOT.GE.B)GO TO 17
   16 IF(BOT.LT.A)GO TO 18
   17 BOT=A
   18 BIN=B-BOT
      IF(BIN.LT.0.0)BIN=0.
      IF(PLATI.LE.PLATO)GO TO 20
      SAVE=BIN
      BIN=BOT
      BOT=SAVE
      DO 19 I=1,NSIG
      A=BIN+RED(I)-SPEDO(I)
      IF(A.LT.0.0)A=0.
   19 SPEDI(I)=A
      GO TO 22
   20 DO 21 I=1,NSIG
      A=BOT-BAND-1.+SPEDO(I)
      IF(A.LT.0.0)A=0.
   21 SPEDI(I)=A
   22 B=PHASE(LTBST)-SPEDI(LTBST)
      DO 23 I=1,NSIG
      A=PHASE(I)-SPEDI(I)-B
      A=A-FLOAT(IFIX(A))
      IF(A.LT.0.0)A=A+1.
   23 PHASE(I)=A
   24 PLATI=BIN*3600./HEDWY
      PLATO=BOT*3600./HEDWY
      BIN=BIN*CYCLE
      BOT=BOT*CYCLE
      WRITE(6,103)(NAME(I),I=1,18)
      WRITE(6,104)NSIG,CYCLE,HEDWY,VOLIN,VOLOT,PLATI,PLATO,BIN,BOT,LTBST
      WRITE(6,105)
      DO 25 I=1,NSIG
      WMIN=(PHASE(I)+.5*RED(I))*CYCLE
      IF(WMIN.GT.CYCLE)WMIN=WMIN-CYCLE
      A=RED(I)*CYCLE
      W(I)=A
      WRITE(6,106)(NAMES(I,K),K=1,8),I,X(I),A,WMIN,PHASE(I)
      IF(I.LT.NSIG)WRITE(6,109)GI(I),GO(I)
   25 CONTINUE
      ALHS=.5*RED(LTBST)-TIME(LTBST)-Z(LTBST)
   26 IF(ALHS)27,28,28
   27 ALHS=ALHS+1.
```

```
      GO TO 26
   28 A=TIME(NSIG)-Z(NSIG)
      BRHS=TIME(LTBST)-Z(LTBST)-.5*RED(LTBST)
   29 IF(BRHS-A)30,31,31
   30 BRHS=BRHS+1.
      GO TO 29
   31 B=TIME(NSIG)+Z(NSIG)
      ASEC=A*CYCLE
      BSEC=B*CYCLE
      FSEC=ALHS*CYCLE
      RSEC=BRHS*CYCLE
      WRITE(6,107)ALHS,FSEC,BRHS,RSEC,A,ASEC,B,BSEC
      READ(5,101)NSAVE,C2
      GO TO (1,34,32,33,35,35),NSAVE
   32 CONTINUE
C     CALL DRAW(NSAVE)
      GO TO 1
   33 CONTINUE
C     CALL DRAW(NSAVE)
   34 CYCLE=C2
      READ(5,100)(NAME(I),I=1,18)
      GO TO 3
   35 CONTINUE
C     CALL DRAW(NSAVE)
   36 STOP
   37 READ(5,108) S1,S2
      DO 38 I=1,NSAVE
      SPEDI(I)=S1
   38 SPEDO(I)=S2
      GO TO 4
      END




      SUBROUTINE DRAW(NSAVE)
      DIMENSION P(50),T(50),X(50),Y(50),Z(50),R(50),SPI(50),RED(50),SPO
     1(50),NAME(12),SI(50),SO(50),CT(99),CD(99),NAMES(50,5),KCD(5)
      COMMON P,T,X,Y,Z,R,SPI,SPO,RED,NAME,SI,SO,PO,PI,FOB,RIB,VIN,VOT,H
     1Y,C,LT,N,T1,TO,BIN,BOT,NAMES,HS,TS
      DATA JACK,KLANKS/0,4H    /
      IF(NSAVE.EQ.5)IF(JACK)54,54,53
      IF(JACK.EQ.0) GO TO 52
   50 JACK=1
   51 PLT=RIB*C
      IF(RIB-T1-BIN/C)55,55,56
   52 CONTINUE
C     CALL PLOT(0.,0.,2)
C     CALL PLOT(0.,0.,3)
```

```
C     CALL PLOT(0.,-3.,-3)
C     CALL PLOT(0.,1.0,-3)
      GO TO 50
   53 CONTINUE
C     CALL PLOT(0.,0.,99)
   54 RETURN
   55 RIB=RIB+1.
      GO TO 51
   56 HOLD=BOT+(FOB+TO)*C
      IF(PLT.LT.HOLD)PLT=HOLD
      SCT=(IFIX(PLT/90.)+1)*10
      IF(TS.GT.SCT)SCT=TS
      SMIN=SCT/2.
      SCD=HS
      IF(SCD.EQ.0.)SCD=1000.
      DMIN=SCD/2.
      DO 57 I=1,N
   57 P(I)=P(I)*C
      MXD=X(N)/SCD+1.
      MXT=9
      TMX=9.*SCT
      DMX=FLOAT(MXD)*SCD
      XAXIS=MXD
   58 IF((RIB+1.)*C.GT.TMX) GO TO 59
      RIB=RIB+1.
      GO TO 58
   59 CONTINUE
C     CALL AXIS1(.5,0.,15HDISTANCE (FEET),-15,XAXIS,0.,0.,SCD,10.0)
C     CALL AXIS1(0.,.5,14HTIME (SECONDS),14,9.,89.99,0.,SCT,10.)
C     CALL LINE1(DMX,-DMIN,TMX,-SMIN,SCD,SCT)
      DO 71 J=1,N
      CT(1)=0
      CT(2)=P(J)-R(J)/2.
      CH=CT(2)
      IF(CT(2))62,60,60
   60 IF(CT(2)+R(J)-C)63,63,61
   61 CH=CH-C
   62 CT(2)=0
   63 CT(3)=CT(2)
      CT(4)=CH+R(J)
      CT(5)=CT(4)
      CT(6)=CT(2)
      CT(7)=CT(2)
      CT(8)=CH+C
      CT(9)=CT(8)
      CT(10)=CT(9)+R(J)
      CT(11)=CT(10)
      CT(12)=CT(9)
      CT(13)=CT(9)
      DO 64 I=14,95
      CT(I)=CT(I-6)+C
      IF(CT(I)-TMX)64,64,65
```

```
   64 CONTINUE
   65 K=(((I-2)/6)*6)+2
      IF(I-K)66,66,67
   66 CT(I)=TMX
      K=I
      GO TO 68
   67 CT(I)=TMX
      CT(I+1)=TMX
      CT(I+2)=CT(I-1)
      CT(I+3)=CT(I+2)
      CT(I+4)=TMX
      K=I+4
   68 CD(1)=X(J)
      CD(2)=CD(1)
      DSHIFT=SCD*0.05
      DO 69 I=3,K,6
      CD(I)=CD(1)-DSHIFT
      CD(I+1)=CD(I)
      CD(I+2)=CD(1)+DSHIFT
      CD(I+3)=CD(I+2)
      CD(I+4)=CD(1)
   69 CD(I+5)=CD(1)
      DIST=X(J)/SCD+.42
      DO 70 I=1,8
   70 KCD(I)=NAMES(J,I)
C     CALL SYMBOL(DIST,1.0,0,1,KCD,90.0,30)
   71 CONTINUE
C     CALL LINE2(CD,CT,K,0,0,1)
      NP1=N+1
      N2=N*2
      DO 72 I=1,N
   72 CD(I)=X(I)
      DO 73 I=NP1,N2
      II=N2-I+1
   73 CD(I)=CD(II)
      CT(I)=FOB*C
      DO 74 I=2,N
   74 CT(I)=CT(I-1)+SO(I-1)*(X(I)-X(I-1))
      DO 75 I=NP1,N2
      IN=N2-I+1
   75 CT(I)=CT(IN)+BOT
C     CALL LINE2(CD,CT,N2,0,0,1)
      CT(1)=RIB*C
      DO 76 I=2,N
   76 CT(I)=CT(I-1)-SI(I-1)*(X(I)-X(I-1))
      DO 77 I=NP1,N2
      IN=N2-I+1
   77 CT(I)=CT(IN)-BIN
C     CALL LINE2(CD,CT,N2,0,0,1)
      SKIP=MXD+4
      IF(SKIP.LT.8.5)SKIP=8.5
      DO 78 I=1,18
```

```
      II=13-I
      IF(NAME(II).NE.KLANKS) GO TO 79
   78 II=II-1
   79 KT=II*6
      XAXIS=XAXIS+.5
      XAXIS=(XAXIS-FLOAT(KT)*.10714)*.5
      IF(XAXIS.LT.0.) XAXIS=0.
C     CALL SYMBOL(XAXIS,-.625,.125,NAME,0.,KT)
C     CALL PLOT(SKIP,0.,-3)
C     IF(NSAVE.EQ.6) CALL PLOT(0.,0.,99)
      RETURN
      END
/*
//GO.SYSIN DD *
```

ROSEDALE AVENUE, FORT WORTH, TEXAS---OFF PEAK
NUMBER OF SIGNALS = 8
CYCLE LENGTH = 60.00
VEHICLE HEADWAY = 2.00
INBOUND VOLUME (INPUT) = 370.00
OUTBOUND VOLUME (INPUT) = 354.00
POSSIBLE INBOUND VOLUME = 632.08
POSSIBLE OUTBOUND VOLUME = 604.74
INBOUND BANDWIDTH = 21.07
OUTBOUND BANDWIDTH = 20.16
RESTRICTING SIGNAL IS NUMBER 1

STREET NAME	SIGNAL NUMBER	DISTANCE FROM THE ORIGIN (FEET)	LENGTH OF RED PHASE (SECONDS)	END OF RED PHASE (SECONDS)	OFFSET (CYCLES)	PROGRESSION SPEEDS (MPH) INBOUND	OUTBOUND
8TH	1	0.0	36.00	18.00	0.0		
						30.00	30.00
5TH	2	1600.00	18.00	39.00	0.50		
						30.00	30.00
HENDERSO	3	1865.00	18.00	39.54	0.49		
						30.00	30.00
COLLEGE	4	2675.00	18.00	9.00	0.0		
						30.00	30.00
HEMPHIL	5	3750.00	21.00	40.50	0.50		
						30.00	30.00
JENNINGS	6	4285.00	18.00	39.00	0.50		
						30.00	30.00
ST.LOUIS	7	4800.00	9.00	4.50	0.0		
						30.00	30.00
SO. MAIN	8	5520.00	27.50	13.80	0.0		

FRONT EDGE OF OUTBOUND BAND = 0.30 CYCLES, 18.00 SECONDS
REAR EDGE OF INBOUND BAND = 2.70 CYCLES, 162.00 SECONDS
SYSTEM TRAVEL TIME INBOUND = 2.09 CYCLES, 125.45 SECONDS
SYSTEM TRAVEL TIME OUTBOUND = 2.09 CYCLES, 125.45 SECONDS

```
  DATE        JOB      ACCNT
16 APR 71 A00068C2 FF18C4Q7        HARDIN

TIME ON    3.787   TIME PRINTED      4.081
TIME OFF   3.834   MINUTES EXECUTION TIME       .78

        SYSTEM TIME     .013  COST $*****2.60
          (RATE = $200/HR.)

        READ                          $******.12
        READ                          $******.28

                        TOTAL COST $*****3.00
     45.20     DOLLARS REMAINING ON ACCOUNT NUMBER



HASP-II JOB STATISTICS --      340 CARDS READ --  486 LINES PRINTED
                       0 CARDS PUNCHED --    2.85 MINUTES RESIDENT TIME
```

APPENDIX C - CHAMBERLAINS METHOD

```
C     THE SYSTEM CONFIGURATION FOR THIS COMPUTER IS AN 1130 COMPUTER
C     WITH 4096 CORES, A 1442 CARD READ-PUNCH, AND AN 1132 PRINTER.
C     SMALLER CONFIGURATIONS REQUIRE IOCS, WRITE, AND READ STMT. CHANGE
      DIMENSION DIST(50), TJ(50),ALFT1(51,2),RT1(51,2),XN1(50,2)
      DIMENSION IT3(50,2),KTR2(2),KTR3(2),XMIN(2),SUM(50,2),OJ(50)
      DIMENSION IT2(50,2)
      KAW=1
    1 READ(1,200)CMIN,CMAX,TO,XL,VMIN,VMAX,Z,FO,FI
      L=XL
      READ (1,201) (DIST(II), TJ(II), II = 1,L)
      MOZ=0
      I=0
      ZZ = 0.0
      IF(L-51) 31,30,30
   30 WRITE(3,202)
      GO TO 1
   31 M = ((((CMAX/CMIN) * VMAX) - VMIN) / Z) + 1.
      VEL=VMIN
      K=0
      NIX=0
C     CALCULATE INDIVIDUAL INTERFERENCES ON BOTH LEFT AND RIGHT
      DO 66 KM=1,M
      IF(KM-2)32,32,33
   32 K=K+1
      NIX=NIX+1
   33 IDX1=0
      IDX2=0
      KTR3(K)=0
      VEL1=VEL*1.4666667
      DO 40 J=1,L
      TIME= 2.*DIST(J)/VEL1
      N1=ABS (TIME)/CMIN
      XN1(J,K)=N1
      TDIFR=TO-TJ(J)
      IF(DIST(J))34,35,35
   34 PART1=TIME+CMIN*XN1(J,K)
      ALFT1(J,K)=ABS (PART1)+TDIFR
      PART1=TIME+CMIN*(XN1(J,K)+1.*
      RT1(J,K)=ABS (PART1)+TDIFR
      GO TO 36
   35 PART1=CMIN*(XN1(J,K)+1.)-TIME
      ALFT1(J,K)=ABS (PART1)+TDIFR
      PART1=CMIN*XN1(J,K)-TIME
      RT1(J,K)=ABS (PART1)+TDIFR
   36 PROD=ALFT1(J,K)*RT1(J,K)
      IF (PROD) 38,38,37
   37 IF (ALFT1(J,K)) 38,38,39
   38 IDX2=IDX2+1
      IT3(IDX2,K)=J
```

```
      KTR3(K)=IDX2
      GO TO 40
   39 IDX1=IDX1+1
      IT2(IDX1,K)=J
      KTR2(K)=IDX1
   40 CONTINUE
      RT1(L+1,K)=VEL
      IT2(IDX1+1,K)=L+1
      ALFT1(L+1,K)=0.
      KTR2(K)=IDX1+1
      VEL=VEL+Z
C     SORT LEFT INTERFERENCES IN DESCENDING ORDER OF MAGNITUDE
      MN=KTR2(K)-2
      N=MN
      DO 45 I=1,MN
      IDX1=0
      DO 43 J=1,N
      KI=IT2(J,K)
      LI=IT2(J+1,K)
      IF(ALFT1(KI,K)-ALFT1(LI,K))41,42,42
   41 IT2(J,K)=LI
      IT2(J+1,K)=KI
      GO TO 43
   42 IDX1=IDX1+1
   43 CONTINUE
      IF(IDX1-N)44,46,46
   44 N=N-1
   45 CONTINUE
C     COMPUTE MINIMUM INTERFERENCE
   46 KTR1=KTR2(K)-1
      I=1
      INDX=IT2(I,K)
      A1=ALFT1(INDX,K)
      B1=RT1(INDX,K)
   47 INDX=IT2(I+1,K)
      SUM(I,K)=ALFT1(INDX,K)+B1
      IF(A1-SUM(I,K))49,49,48
   48 A1=SUM(I,K)
   49 I=I+1
      IF(RT1(INDX,K)-B1)51,51,50
   50 B1=RT1(INDX,K)
   51 IF(I-KTR1)47,52,52
   52 SUM(I,K)=B1
      IF(A1-B1)53,53,54
   53 XMIN(K)=A1
      GO TO 55
   54 XMIN(K)=B1
   55 IF(MOZ)57,56,57
   56 MOZ=1
   10 WRITE(3,203)(J,DIST(J),TJ(J),J=1,L)
   11 WRITE(3,204) CMIN,CMAX,TO,XL,VMIN,VMAX,Z
   12 WRITE (3,205)
```

```
C     CALL DATSW(3,KAW)
   57 IF(ZZ)16,13,16
   13 WRITE(3,206) KM,XMIN(K)
      GO TO (58,59),KAW
   58 DO 15 J=1,L
   14 WRITE(3,207)XN1(J,K)
   15 WRITE(3,208) ALFT1(J,K),RT1(J,K)
   59 ZZ=ZZ+1.0
   60 GO TO (66,63),NIX
   16 WRITE(3,209) KM,XMIN(K)
C     CALL DATSW(3,KAW)
      GO TO (61,62),KAW
   61 DO 18 J=1,L
   17 WRITE (3,207)XN1(J,K)
   18 WRITE (3,208) ALFT1(J,K),RT1(J,K)
   62 GO TO 60
   63 IF(XMIN(1)-XMIN(2))64,64,65
   64 K=2
      GO TO 66
   65 K=1
   66 CONTINUE
      IF(M-1)70,70,67
   67 IF(K-1)68,68,69
   68 K=2
      KI=1
      GO TO 71
   69 K=1
   70 KI=2
   71 VOPT=RT1(L+1,K)
      WRITE(3,215)VOPT
      IF(VOPT-VMAX)72,72,73
   72 COPT=CMIN
      S=1.
      GO TO 74
   73 S=VOPT/VMAX
      COPT=S*CMIN
      VOPT=VMAX
   74 BW=S*(2.*TO-XMIN(K))
      KTR1=KTR2(K)-1
      KTR4=KTR2(K)
      DO 75 I=1,KTR4
      IF(XMIN(K)-SUM(I,K))75,86,75
   75 CONTINUE
C     THE LARGEST INTERFERENCE ON THE LEFT IS THE MINIMUM
C     INTERFERENCE
      ALM=XMIN(K)*.5
      DO 78 I=1,KTR1
      IBO=IT2(I,K)
      ALFT1(IBO,KI)=ALFT1(IBO,K)
      RT1(IBO,KI)=0.
      IF(DIST(IBO))76,77,77
   76 XNJ=XN1(IBO,K)
```

```
      GO TO 78
   77 XNJ=XN1(IBO,K)+1.
   78 OJ(IBO) = ( .5 * (XNJ * CMIN + TO - TJ(IBO))) * S
   79 IF(KTR3(K))20,20,80
   80 KTR = KTR3(K)
      DO 85 I = 1,KTR
      IBO=IT3(I,K)
      ALFT1(IBO,KI)=0.
      RT1(IBO,KI)=0.
      IF(ALFT1(IBO,K))81,81,82
   81 IF(DIST(IBO))84,83,83
   82 IF(DIST(IBO))83,84,84
   83 XNJ=XN1(IBO,K)+1.
      GO TO 85
   84 XNJ=XN1(IBO,K)
   85 OJ(IBO) = ( .5 * (XNJ * CMIN + TO - TJ(IBO))) * S
      GO TO 20
   86 I=I+1
      IBO=IT2(I,K)
      IF(ALFT1(IBO,K))94,94,87
C     COMPUTE OFFSETS BASED ON INTERFERENCE ON THE LEFT
   87 ALM=ALFT1(IBO,K)*.5
      RTO=(XMIN(K)-ALFT1(IBO,K))*.5
      DO 90 J=1,KTR1
      IBO=IT2(J,K)
      ALFT1(IBO,KI)=ALFT1(IBO,K)
      RT1(IBO,KI)=0.
      IF(DIST(IBO))88,89,89
   88 XNJ=XN1(IBO,K)
      GO TO 90
   89 XNJ=XN1(IBO,K)+1.
   90 OJ(IBO) = ( .5 * (XNJ * CMIN + TO - TJ(IBO))) * S
C     COMPUTE OFFSETS BASED ON INTERFERENCE ON THE RIGHT
      I=I-1
      DO 93 J=1,I
      IBO=IT2(J,K)
      ALFT1(IBO,KI)=0.
      RT1(IBO,KI)=RT1(IBO,K)
      IF(DIST(IBO))91,92,92
   91 XNJ=XN1(IBO,K)+1.
      GO TO 93
   92 XNJ=XN1(IBO,K)
   93 OJ(IBO) = ( .5 * (XNJ * CMIN + TO - TJ(IBO))) * S
      GO TO 79
C     THE LARGEST INTERFERENCE ON THE RIGHT IS THE MINIMUM
C     INTERFERENCE
   94 RTO=XMIN(K)*.5
      DO 98 I=1,KTR1
      IBO=IT2(I,K)
      ALFT1(IBO,KI)=0.
      RT1(IBO,KI)=RT1(IBO,K)
      IF(DIST(IBO))95,96,96
```

```
 95 XNJ=XN1(IBO,K)+1.
    GO TO 97
 96 XNJ=XN1(IBO,K)
 97 T=TJ(IBO)
 98 OJ(IBO)=(.5*(XNJ*CMIN+TO-T))*S
    GO TO 79
 20 WRITE(3,204)CMIN,CMAX,TO,XL,VMIN,VMAX,Z
 21 WRITE(3,203)(J,DIST(J),TJ(J),J=1,L)
    VAL=COPT/2.0
    DO 101 J=1,L
    DUMB = 0.
 99 IF(OJ(J)-VAL)101,101,100
100 DUMB = OJ(J) - COPT
    OJ(J) = DUMB
    GO TO 99
101 CONTINUE
 22 WRITE(3,210)VOPT,COPT,BW,(J,OJ(J),J=1,L)
    DBWO=.5*BW*(FO-FI)/(FO+FI)
    B1=.5*BW+ABS (DBWO)
    T=TO*S
    IF(B1-T )102,102,25
102 IF(DBWO)103,24,104
103 DBWO=-DBWO
    KIP=1
    GO TO 105
104 KIP=0
105 DA=S*ALM-DBWO
    IF(DA)107,106,106
106 ALM=DA
    MAP=1
    GO TO 108
107 ALM=0.
    MAP=2
108 DO 112 I=1,L
    AL1=.5*ALFT1(I,KI)*S
    IF(ALM-AL1)109,112,112
109 DP=AL1-ALM
    IF(KIP)110,110,111
110 OJ(I)=OJ(I)-DP
    GO TO 112
111 OJ(I)=OJ(I)+DP
112 CONTINUE
    GO TO (23,113),MAP
113 RTO=S*RTO+DA
    DO 117 I=1,L
    RT2=.5*RT1(I,KI)*S
    IF(RTO-RT2)114,117,117
114 DP=RT2-RTO
    IF(KIP)116,116,115
115 OJ(I)=OJ(I)-DP
    GO TO 117
116 OJ(I)=OJ(I)+DP
```

```
117 CONTINUE
 23 WRITE(3,211)(I,OJ(I),I=1,L)
 24 WRITE(3,212)FO,FI
    GO TO 26
 25 WRITE(3,213)
    ALPHA=B1-T
    IF(DBWO)118,24,119
118 DBWO=DBWO+ALPHA
    GO TO 103
119 DBWO=DBWO-ALPHA
    GO TO 104
 26 WRITE(3,214)
    GO TO 1
200 FORMAT (12F6.1)
201 FORMAT (2F10.2)
202 FORMAT (1H ,5X,54HERROR***NUMBER OF INTERSECTIONS EXCEEDS 50-JOB D
   1ELETED)
203 FORMAT (1H0        ,1HJ,5X,11HDISTANCE TO,5X,13HGREEN TIME AT/1H ,4
   1X,14HINTERSECTION J,3X,14HINTERSECTION J///(1H ,I2,6X,F8.1,10X,F6.
   22))
204 FORMAT (1H ,3X,18H*** INPUT DATA ***///1H ,2X,4HCMIN,10X,4HCMAX,10
   1X,2HTO,7X,2HXL,8X,4HVMIN,9X,4HVMAX,7X,1HZ//(1H ,F6.1,8X,F6.1,6X,F6
   2.1,4X,F6.1,5X,F6.1,6X,F6.1,5X,F6.3))
205 FORMAT (1H //1H ,2X,21HMINIMUM INTERFERENCES//1H )
206 FORMAT (1H ,10H ITERATION,5X,12HINTERFERENCE/1H ,2X,6HNUMBER,9X,8H
   1IN SECS.//(1H ,4X,I3,12X,F7.3))
207 FORMAT(1H0,10X,F6.1)
208 FORMAT(1H0,10X,8HLEFT INT,F6.2,10X,9HRIGHT INT,F6.2)
209 FORMAT (1H ,4X,I3,12X,F7.3)
210 FORMAT (1H0,19HOPTIMUM VELOCITY IS,F6.1,2X,3HMPH/  1H ,23HOPTIMUM
   1CYCLE LENGTH IS,F6.1,2X,4HSEC./  1H ,12HBANDWIDTH IS,F6.2,3X,4HSEC
   2. //1H ,10X,6HOFFSET/1H ,10X,7HSECONDS/  (1H ,3X,I2,F11.1))
211 FORMAT (1H0,7X,11HNEW OFFSETS///(3X,I2,F11.1))
212 FORMAT (1H3,5X,17HOUTBOUND FLOW IS ,F6.1,13H VEHICLES/HR.///6X,17H
   1INBOUND FLOW IS  ,F6.1,13H VEHICLES/HR.)
213 FORMAT (1H0,3X,60HNEW BW EXCEEDS TO*****  BW=TO  IN DIRECTION OF H
   1EAVIEST FLOW)
214 FORMAT (1H ///1H ,19H *** END OF JOB *** /1H1)
215 FORMAT (1H0,7HVOPT IS,F6.2,1X,3HMPH/)
    END
```

J	DISTANCE TO INTERSECTION J	GREEN TIME AT INTERSECTION J
1	395.0	31.00
2	1398.0	25.00
3	2037.0	25.00
4	2990.0	25.00

*** INPUT DATA ***

CMIN	CMAX	TC	XL	VMIN	VMAX	Z
50.0	80.0	19.0	4.0	20.0	30.0	1.00

MINIMUM INTERFERENCES

ITERATION NUMBER	INTERFERENCE IN SECS.
1	11.068

0.0

LEFT INT 11.07 RIGHT INT 14.93

1.0

LEFT INT -1.32 RIGHT INT 39.32

2.0

LEFT INT 5.11 RIGHT INT 32.89

4.0

LEFT INT 40.14 RIGHT INT -2.14

2 12.351

0.0

LEFT INT 12.35 RIGHT INT 13.65

1.0

LEFT INT 3.22 RIGHT INT 34.78

2.0

LEFT INT 11.73 RIGHT INT 26.27

```
     28              10.350

            0.0

          LEFT  INT 26.54           RIGHT  INT -0.54

            0.0

          LEFT  INT  3.44           RIGHT  INT 34.56

            1.0

          LEFT  INT 34.90           RIGHT  INT  3.10

            1.0

          LEFT  INT  7.25           RIGHT  INT 30.75

VOPT IS 44.00 MPH

   *** INPUT DATA ***
```

CMIN	CMAX	TC	XL	VMIN	VMAX	Z
50.0	80.0	19.0	4.0	20.0	30.0	1.000

J	DISTANCE TO INTERSECTION J	GREEN TIME AT INTERSECTION J
1	395.0	31.00
2	1398.0	25.00
3	2037.0	25.00
4	2990.0	25.00

```
OPTIMUM VELOCITY IS  30.0  MPH
OPTIMUM CYCLE LENGTH IS  73.3  SEC.
BANDWIDTH IS 43.32   SEC.
```

	OFFSET SECONDS
1	-8.8
2	32.3
3	32.3
4	-4.4

```
    OUTBOUND FLOW IS  400.0 VEHICLES/HR.

    INBOUND FLOW IS   400.0 VEHICLES/HR.

*** END OF JOB ***
```

```
  DATE        JOB       ACCNT
23 APR 71 A0009817 FF18C4Q7        HARDIN

TIME ON   21.790    TIME PRINTED     21.948
TIME OFF 21.873    MINUTES EXECUTION TIME        .84

          SYSTEM TIME     .014  COST $*****2.80
            (RATE = $200/HR.)

          READ                          $******.10
          READ                          $******.40

                         TOTAL COST $*****3.30
     25.04     DOLLARS REMAINING ON ACCOUNT NUMBER



HASP-II JOB STATISTICS --     306 CARDS READ --  733 LINES PRINTED
                       0 CARDS PUNCHED --    4.95 MINUTES RESIDENT TIME
```

APPENDIX D - SIGPROG

```
      INTEGER OPTION, SCAN
      DIMENSION  TT(2,99), FEET(2,99), SPEED(2,99), OFFSET(100),
     1  OFF PB(2,100), TITLE(10), SUB TIT(10), OUTFIT(2,10),
     2 VELCTY(8,99), NAME(3,100), DIR 1(2), DIR 2(2), CYCLES(4), BW(4)
      DIMENSION ID 1 (10),  ID 2 (10)
      DIMENSION  IFMT(14), IX(101)
      COMMON  STA(100), SPLIT(100), CUT R(100), CUT L(100), STAT,
     1  LEAVE, SLICE R, SLICE L, NO SIGS, CYCLE, BANDW, IBASE,
     2  SPLIT P(100), YELLOW(100), PED X(100)
      DATA  MARK E, NO, ASTK, BLANK, IYES / 1HE, 2HNO, 1H*, 1H , 3HYES
      DATA  MESS A, MESS B, SCAN  / 3HMPH, 3HSEC, 4HSCAN  /
      DATA IFMT/ (1H  , 2HEF , 2HF  , 1H=F , 7.3, ,-1X, , 2HAT , F5.0 ,
     1 ,1X, , 2HSE , 2HCS ,   , X,I3 , ) /
      DATA IX/ 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 ,
     1 14 , 15 , 16 , 17 , 18 , 19 , 20 , 21 , 22 , 23 , 24 , 25 , 26 ,
     2 27 , 28 , 29 , 30 , 31 , 32 , 33 , 34 , 35 , 36 , 37 , 38 , 39 ,
     3 40 , 41 , 42 , 43 , 44 , 45 , 46 , 47 , 48 , 49 , 50 , 51 , 52 ,
     4 53 , 54 , 55 , 56 , 57 , 58 , 59 , 60 , 61 , 62 , 63 , 64 , 65 ,
     5 66 , 67 , 68 , 69 , 70 , 71 , 72 , 73 , 74 , 75 , 76 , 77 , 78 ,
     6 79 , 80 , 81 , 82 , 83 , 84 , 85 , 86 , 87 , 88 , 89 , 90 , 91 ,
     7 92 , 93 , 94 , 95 , 96 , 97 , 98 , 99 , 100 , 101 /
C     READ TITLE OF OUTFIT
      READ(5,5)  ((OUTFIT(I,J), J=1,10),I=1,2)
    5 FORMAT(20A4)
C
C     READ IDENTIFICATION OF NEXT PROBLEM AND BASIC INFORMATION
   10 WRITE(6,11)
   11 FORMAT(1H1 / 1H0, 9X, 32X, 17HINPUT INFORMATION / 1X )
      WRITE(6,6)  ((OUTFIT(I,J), J=1,10),I=1,2)
    6 FORMAT(1X,20A4/)
      READ(5,15)  ID 1, TITLE, ID 2, SUB TIT
   15 FORMAT(20A4 / 20A4 )
      WRITE(6,16)  ID1, TITLE, ID2, SUB TIT
   16 FORMAT(10X, 20A4 / 10X, 20A4 )
      READ(5,20)  ID 1, NO SIGS
   20 FORMAT(10A4, I5 )
      WRITE(6,21) ID1, NO SIGS
   21 FORMAT(10X, 10A4, I5 )
      IF(NO SIGS .LE. 100)  GO TO 30
      WRITE(6,25)  TITLE, SUB TIT, NO SIGS
   25 FORMAT(1H1, 10A4, 2X, 10A4 / 20HONUMBER OF SIGNALS =, I5 /
     1  44H MAXIMUM NUMBER ACCEPTABLE TO PROGRAM IS 100  /
     2  22HOPROCESSING TERMINATED  )
      STOP
   30 READ(5,35) ID 1, CYCLE A, CYCLE Z
   35 FORMAT(10A4, F7.2, F10.2 )
      WRITE(6,36)  ID 1, CYCLE A, CYCLE Z
   36 FORMAT(10X, 10A4, F7.2, F10.2 )
      READ(5,40)  ID 1, SPD TOL
```

```
   40 FORMAT(10A4, F7.2 )
      WRITE(6,41)  ID 1, SPD TOL
   41 FORMAT(10X, 10A4, F7.2 )
      READ(5,40) ID1, BASE OF
      WRITE(6,41)  ID1, BASE OF
      READ(5,43) ID 1, ID 2
   43 FORMAT(20A4 )
      WRITE(6,44)  ID 1, ID 2
   44 FORMAT(10X, 20A4 )
      READ(5,45)  ID 1, DIR 1, DIR 2
   45 FORMAT(10A4, 2A4, 2X, 2A4 )
      WRITE(6,46)  ID 1, DIR 1, DIR 2
   46 FORMAT(10X, 10A4, 2A4, 2X, 2A4 )
      READ(5,50)  ID 1, BW PCT 1, BW PCT 2
   50 FORMAT(10A4, F7.2, F10.2 )
      WRITE(6,51)  ID 1, BW PCT 1, BW PCT 2
   51 FORMAT(10X, 10A4, F7.2, F10.2 )
      BW PCT 1 = BW PCT 1 / (BW PCT 1 + BW PCT 2) * 100.
      BW PCT 2 = 100. - BW PCT 1
      I BW 1 = BW PCT 1 + .50001
      I BW 2 = 100 - I BW 1
C
C     INITIALIZE ACCUMULATORS AND INDICATORS
      STA(1) = 100.0
      SPD MAX = 0.0
      LEAVE = 0
C
C     READ SIGNAL SPLIT CONTROL INFORMATION
      DO 90 I = 1,NO SIGS
      READ(5,55) ID 1, (NAME(J,I),J=1,3), SPLIT P(I)
   55 FORMAT(10A4, 20X, 3A4, F8.3 )
      WRITE(6,56)  ID 1, (NAME(J,I),J=1,3), SPLIT P(I)
   56 FORMAT(10X, 10A4, 20X, 3A4, F8.2 )
      READ(5,57)  ID 1, PED X(I), YELLOW(I)
   57 FORMAT(10A4, 20X, F12.5, F8.3 )
      WRITE(6,58)  ID 1, PED X(I), YELLOW(I)
   58 FORMAT(10X, 10A4, 20X, F9.2, F11.2 )
   60 IF(I .EQ. NO SIGS)  GO TO 90
C
C     READ SIGNAL SPACING INFORMATION
      READ(5,65) MARK, ID1, (FEET(J,I),J=1,2)
   65 FORMAT(A1, 9A4, A3,  F7.2, F10.2 )
      IF(FEET(2,I ).EQ. 0. ) FEET(2,I) = FEET(1,I)
      WRITE(6,66) MARK, ID1, (FEET(J,I),J=1,2)
   66 FORMAT(10X, A1, 9A4, A3, F7.2, F10.2 )
      IF(MARK .NE. MARK E)  GO TO 70
C     HAVE JUST READ OPTIONAL TRAVEL TIME SPACING CARD INSTEAD
      EXTRA=1.
      TT(1,I) = FEET(1,I)
      TT(2,I) = FEET(2,I)
      SPEED(1,I) = 0.
      SPEED(2,I) = 0.
```

```
      SPD MAX = 50.0
      GO TO 85
   70 READ(5,75)  ID 1, (SPEED(J,I),J=1,2 )
   75 FORMAT(10A4, F7.2, F10.2 )
      EXTRA=0.
      IF(SPEED(2,I) .EQ. 0.)  SPEED(2,I) = SPEED(1,I)
      WRITE(6,76)  ID 1, (SPEED(J,I),J=1,2 )
   76 FORMAT(10X, 10A4, F7.2, F10.2 )
      TT(1,I) = FEET(1,I) / SPEED(1,I) * 60. / 88.
      TT(2,I) = FEET(2,I) / SPEED(2,I) * 60. / 88.
C     FIND MAXIMUM PROGRESSIVE SPEED DESIRED
      DO 80 J=1,2
      IF(SPEED(J,I) .LE. SPD MAX)  GO TO 80
      SPD MAX = SPEED(J,I)
   80 CONTINUE
C
C     CALCULATE TRAVEL TIME STATION OF NEXT SIGNAL
   85 STA(I+1) = STA(I) + (TT(1,I) + TT(2,I)) / 2.
   90 CONTINUE
      READ(5,92)  ID 1, OPTION
   92 FORMAT(10A4, 1X, A4 )
      WRITE(6,93)  ID 1, OPTION
   93 FORMAT(10X, 10A4, 1X, A4 )
      READ(5,94) ID1,IPCH,ISIZE
   94 FORMAT(10A4,2X,A3,2X,I1)
      WRITE(6,95) ID1,IPCH,ISIZE
   95 FORMAT(10X,10A4,2X,A3,2X,I1)
      READ(5,96)  ID 1, ID 2(1), IWORD
   9L FORMAT(10A4, A3, A2 )
      WRITE(6,97)  ID 1, ID 2(1), IWORD
   97 FORMAT(10X, 10A4, A3, A2 )
C
C     SCAN BETWEEN CYCLE LIMITS FOR BEST INTEGER FIT
      WRITE(6,100) TITLE, SUB TIT
  100 FORMAT(1H1, 9X, 23HCYCLE SCAN FOR BEST FIT / 1H0, 9X, 10A4 /
     1  10X, 10A4  / 1X )
      EFF MAX = 0.
      LIM 1 = CYCLE A * (SPD MAX - SPD TOL) / SPD MAX + .99999
      LIM 2 = CYCLE Z * (SPD MAX + SPD TOL) / SPD MAX
      JUMP = 1
      IF(SPD TOL .EQ. 0.)  JUMP = 5
      DO 110 I=LIM 1,LIM 2,JUMP
      CYCLE = I
      IF(CYCLE A .EQ. CYCLE Z .AND. SPD TOL .EQ. 0.) CYCLE = CYCLE A
      CALL OPTIM
      EFFIC = BAND W / CYCLE * 100.
      IF(EFFIC .LE. 0.)  EFFIC = 0.0001
      ISUB = EFFIC + 1.5
      IF(ISUB .GT. 0)  GO TO 105
      ISUB = 1
  105 INDEX = ISUB - 1
      IFMT(12) = IX(ISUB)
```

```
      WRITE(6,IFMT)  EFFIC, CYCLE, INDEX
      IF(EFFIC .LE. EFF MAX)  GO TO 110
      EFF MAX = EFFIC
      CY OPT = CYCLE
  110 CONTINUE
      WRITE(6,115)  CY OPT, EFF MAX
  115 FORMAT(1H0, 9X, 12HBEST FIND IS / 10X, 11H  CYCLE OF, F5.0, 2X,
     1  12HEFFICIENCY =, F8.3 )
      IF(OPTION .EQ. SCAN)  GO TO 325
      IF(LIM 1 .GE. LIM 2)  GO TO 145
      WRITE(6,120)
  120 FORMAT( 1H0, 9X, 22HITERATION IMPROVEMENTS )
C
C     ZERO IN ON BEST CYCLE BY SUCCESSIVE ITERATIONS
      DO 135 J=1,10
      EXP = 2. ** (J-1)
      A = 1. / EXP
      LIM 1 = (CY OPT - A) * EXP * 10. + .00000001
      LIM 2 = (CY OPT + A) * EXP * 10. + .00000001
      DO 125 I=LIM1,LIM2,5
      CYCLE = I
      CYCLE = CYCLE / (EXP * 10.)
      CALL OPTIM
      EFFIC = BAND W / CYCLE * 100.
      IF ( EFFIC .LE. EFF MAX)  GO TO 125
      EFF MAX = EFFIC
      CY OPT = CYCLE
  125 CONTINUE
      WRITE(6,130)  CY OPT, EFF MAX
  130 FORMAT(21X, F8.3, 11X, F8.3 )
  135 CONTINUE
  145 CONTINUE
C
C     DETERMINE BASIC OFFSETS
      CYCLE = CY OPT
      CALL OPTIM
      LEAVE = 1
      CALL OPTIM
      OFF I PB =(CYCLE * SPLIT(IBASE) / 200. - SLICE R - BAND W / 2. )
     1  * 100. / CYCLE + BASE OF
      DO 150 I=1,NO SIGS
      J = (STA(I)-STAI) / CYCLE * 2. + .00001
      HALF CY = J
      OFFSET(I) = 50. * (HALF CY + 1.) + BASE OF
      IF(SLICE R .GE. CUT R(I)) OFFSET(I) = 50. * HALF CY + BASE OF
      OFF PB(1,I) = OFF I PB + (STA(I)-STAI) / CYCLE * 100.
      OFF PB(2,I) = -OFF PB(1,I) + 2. * BASE OF
  150 CONTINUE
C
C     PREPARE CYCLE LENGTHS FOR PRINT-OUT
      CYCLES(2) = FLOAT(INT(CY OPT/5.+.00001)) * 5.
      CYCLES(1) = CYCLES(2) - 5.
```

```
      CYCLES(3) = CYCLES(2) + 5.
      CYCLES(4) = CYCLES(2) + 10.
      ITEST = 3
      IF(INT(CY OPT/5.+.00001) .EQ. INT((CY OPT+2.5)/5.+.00001))ITEST=2
      IF(ITEST .EQ. 2)  GO TO 151
      HOLD = CYCLES(2)
      CYCLES(2) = CYCLES(3)
      CYCLES(3) = HOLD
      GO TO 152
  151 CYCLES(4) = CYCLES(3)
      CYCLES(3) = CYCLES(1)
  152 CYCLES(1) = CY OPT
C
C     ADJUST BANDWIDTH TO FAVOR BY DIRECTION AS REQUESTED
      BW 1 = CYCLE * EFF MAX / 100.
      BW 2 = BW 1
      IF(SLICE L + SLICE R .EQ. 0.0)  GO TO 499
      DIR = 1.0
      IF(BW PCT 1 .EQ. 50.0)  GO TO 499
      IF(BW PCT 1 .LT. 50.0)  DIR = -1.0
      DECR = BW 1 * (BW PCT 1 - BW PCT 2) / 100. * DIR
      IF(DECR .GE. SLICE R + SLICE L)  GO TO 420
C     REQUIRED CUT DECREASES ARE LESS THAN TOTAL CUTS
      IF(DECR .GE. BW 2 - 5.)  GO TO 430
      BW 1 = BW 1 + DECR * DIR
      BW 2 = BW 2 - DECR * DIR
      SLICE M = SLICE L
      SLICE S = SLICE R - DECR
      IF(SLICE S .GE. 0.0)  GO TO 450
      SLICE M = SLICE L + SLICE S
      SLICE S = 0.0
      GO TO 450
C     SHIFT TO MAXIMUM WIDTH IN FAVORED DIRECTION
  420 BW 1 = BW 1 + (SLICE R + SLICE L) * DIR
      BW 2 = BW 2 - (SLICE R + SLICE L) * DIR
      GO TO 440
  430 BW 1 = BW 1 + SLICE R + SLICE L
      BW 2 = 0.0
      IF(DIR .GT. 0.0)  GO TO 440
      BW 1 = 0.0
      BW 2 = BW 2 + SLICE R + SLICE L
  440 SLICE S = 0.0
      SLICE M = 0.0
C     ADJUST OFFSETS
  450 DO 470 I=1,NO SIGS
      IF(SLICE R .GE. CUT R(I)) GO TO 460
      IF(SLICE M .GE. CUT L(I)) GO TO 470
      SHIFT = (CUT L(I) - SLICE M) / CYCLE * 100. * DIR
      OFFSET(I) = OFFSET(I) - SHIFT
      GO TO 470
  460 IF(SLICE S .GE. CUT R(I)) GO TO 470
      SHIFT = (CUT R(I) - SLICE S) / CYCLE * 100. * DIR
```

```
      OFFSET(I) = OFFSET(I) + SHIFT
  470 CONTINUE
      B RIGHT = (SLICE R - SLICE S) / CYCLE * 50. * DIR
      B LEFT = (SLICE L - SLICE M) / CYCLE * 50. * DIR
      DO 480 I = 1,NO SIGS
      OFF PB(2,I) = OFF PB(2,I) + B RIGHT - B LEFT
      OFF PB(1,I) = OFF PB(1,I) + B RIGHT - B LEFT
  480 CONTINUE
  499 EFF 1 = BW 1 / CYCLE * 100.
      EFF 2 = BW 2 / CYCLE * 100.
C
C     ADJUST OFFSETS FOR DIRECTIONAL DIFFERENCES
      SHIFT = 0.
      DO 155 I=1,NO SIGS
      J = I - 1
      IF(J .LE. 0)  GO TO 153
      AVG = (TT(1,J) + TT(2,J)) / 2.
      SHIFT = SHIFT + (AVG - TT(2,J)) * 100. / CYCLE
  153 X = OFFSET(I) + SHIFT + 200.
      OFFSET(I) = AMOD(X,100.)
      X = OFF PB(1,I) + SHIFT + 200.
      OFF PB(1,I) = AMOD(X,100.)
      X = OFF PB(2,I) + SHIFT + 100000.
      OFF PB(2,I) = AMOD(X,100.)
  155 CONTINUE
C
C     CALCULATE PROGRESSIVE SPEEDS ACHIEVED AT EACH CYCLE LENGTH
      DO 165 J=1,4
      FACTOR = CY OPT / CYCLES(J)
      DO 160 I = 2,NO SIGS
      VELCTY(J,I-1) = SPEED(1,I-1) * FACTOR
      VELCTY(J+4,I-1) = SPEED(2,I-1) * FACTOR
      IF(SPEED(1,I-1) .NE. 0. )  GO TO 160
      VELCTY(J,I-1) = TT(1,I-1) / FACTOR
      VELCTY(J+4,I-1) = TT(2,I-1) / FACTOR
  160 CONTINUE
  165 CONTINUE
C
C     PRINT TABLE OF TIMING PLANS - HEADING
  200 ITEMS = 8
      IPAGE = 1
      WRITE(6,205) TITLE, (OUTFIT(1,J),J=3,10), SUB TIT,
     1                    (OUTFIT(2,J),J=3,10)
  205 FORMAT(1H1, 27X, 34HTRAFFIC SIGNAL SYSTEM TIMING PLANS  / 1H0,
     1  8X, 10A4, 8A4, / 9X, 10A4, 8A4  )
      IF(IPCH.EQ.IYES) WRITE(7,204) ISIZE,EXTRA
  204 FORMAT(I1,1X,F2.0)
      IF(IPCH .EQ. IYES)  WRITE(7,206) TITLE, SUB TIT
  206 FORMAT(20A4)
      WRITE(6,210)  CY OPT, I BW 1, I BW 2, (CYCLES(J),J1,4)
  210 FORMAT(1H0, 8X, 4HPLAN, F6.1, 2H (, I3, 1H-, I3, 1H), 4X,
     1  22HCYCLE LENGTH (SECONDS), 2X, 4F6.1 )
```

```
      WRITE(6,215)  DIR 1
      DO 220 I=1,4
  215 FORMAT(1H0, 12X, 16H****************, 4X, 2A4, 10H DIRECTION )
      BW(I) = EFF 1 * CYCLES(I) / 100.
  220 CONTINUE
      BW1 = BW(1)
      WRITE(6,225)  BW
  225 FORMAT(13X, 1H*, 14X, 1H*, 8X, 20HBAND WIDTH (SECONDS), 4F6.1 )
      WRITE(6,230)  EFF1, EFF1, EFF1, EFF1
  230 FORMAT(13X, 16H* LISTING IS *, 8X, 21HEFFICIENCY (PER CENT),
     1  F5.1, 3F6.1  )
      WRITE(6,235)  DIR1, DIR2
  235 FORMAT(13X, 6H*  IN , 2A4, 2H * / 13X, 16H* DIRECTION  *, 4X,
     1  2A4, 10H DIRECTION )
      DO 240 I=1,4
      BW(I) = EFF2 * CYCLES(I)  / 100.
  240 CONTINUE
      BW2 = BW(1)
      WRITE(6,245) BW
  245 FORMAT(13X, 1H*, 14X, 1H*, 8X, 20HBAND WIDTH (SECONDS), 4F6.1 )
      WRITE(6,250)  EFF2, EFF2, EFF2, EFF2
  250 FORMAT(13X, 16H****************, 8X, 21HEFFICIENCY (PER CENT),
     1  F5.1, 3F6.1 )
      IF(IPCH .EQ. IYES)  WRITE(7,251)  DIR 1, DIR 2, CY OPT, BW 1,
     1  BW 2, EFF 1, EFF 2
  251 FORMAT(4A4, 5F6.1)
  255 WRITE(6,260)
  260 FORMAT(   1H0, 8X, 72HSIGNAL          OFFSETS-PER CENT  INTERVAL  S
     1GNAL  OPT.  BEST  --OTHERS--  )
      MESS = MESS A
      IF(SPEED(1,1) .EQ. 0.0)  MESS = MESS B
      WRITE(6,270)  MESS
  270 FORMAT(9X, 69HLOCATION        BEG.  MID.  END.  G-PCT.-Y  SPCNG.  P
     1OGRESSIVE SPEEDS-,A3 / 9X, 72H************  ****************  ***
     2****  ******  *********************  )
C     PRINT BOTTOM PART OF TABLE
      ITEST = 1
      IF(SPEED(1,1) .EQ. 0.)  ITEST = 2
      DO 320 I=1,NO SIGS
      IF(OFF BEG .LT. 0.)  OFF BEG = OFF BEG + 100.
      OFF BEG = OFFSET(I   ) - SPLIT(I   ) / 2.
      OFF END = OFFSET(I   ) + SPLIT(I   ) / 2.
      IFF(OFF END .GE. 100.)  OFF END = OFF END - 100.
      OFF 1 = OFF BEG
      OFF 3 = OFF END
      AMBER = YELLOW(I) / CYCLES(1) * 100.
      WRITE(6,300) (NAME(J,I),J=1,3), OFF BEG, OFFSET(I), OFF END,
     1  SPLIT(I), AMBER
  300 FORMAT(1H0, 8X, 3A4, 3F6.1, F6.1, F4.1  )
      A = OFF BEG
      B = OFFSET(I)
      C = OFF END
```

```
      D = SPLIT(I) + AMBER
      OFF BEG = OFF PB(1,I) - EFF 1 / 2.
      IF(OFF BEG .LT. 0.)  OFF BEG = OFF BEG + 100.
      OFF END = OFF PB(1,I) + EFF 1 / 2.
      IF(OFF END .GE. 100.)  OFF END = OFF END - 100.
      X MARK = BLANK
      IF(SPLIT(I) - .01 .LE. EFF 1)  X MARK = ASTK
      E = OFF BEG
      F = OFF END
      IF(I .EQ. NO SIGS)  GO TO 320
      G = (FEET(1,I) + FEET(2,I)) / 2.
      H = VELCTY(1,I)
      WRITE(6,305) X MARK, OFF BEG, OFF END, DIR 1, FEET(1,I),
     1    (VELCTY(J,I),J=1,4)
  305 FORMAT(1H0,11XA1,2X,6HP.BAND,F6.1,6X,F6.1, 2X, 2A4, F8.1, 4F6.1 )
      OFF BEG = OFF PB(2,I) - EFF 2 / 2.
      IF(OFF BEG .LT. 0.)  OFF BEG = OFF BEG + 100.
      OFF END = OFF PB(2,I) + EFF 2 / 2.
      IF(OFF END .GE. 100.)  OFF END = OFF END - 100.
      X MARK = BLANK
      IF(SPLIT(I) - .01 .LE. EFF 2)  X MARK = ASTK
      O = OFF BEG
      P = OFF END
      Q = VELCTY(5,I)
      WRITE(6,310) X MARK, OFF BEG, OFF END, DIR 2, FEET(2,I),
     1    (VELCTY(J,I),J=5,8)
  310 FORMAT(12X, A1, 2X, 6HP.BAND,F6.1,6X,F6.1, 2X, 2A4, F8.1, 4F6.1 )
      IF(IPCH .EQ. IYES) WRITE(7,311) (NAME(J,I),J=1,3),A,B,C,D,E,F,G,
     1 H,O,P,Q,ITEST
  311 FORMAT(3A4, 11F6.1, I2 )
      ITEMS = ITEMS - 1
      IF(ITEMS .GT. 0)  GO TO 320
      ITEMS = 10
      IPAGE = IPAGE + 1
      OFF BEG = OFFSET(I+1) - SPLIT(I+1) / 2.
      IF(OFF BEG .LT. 0.)  OFF BEG = OFF BEG + 100.
      OFF END = OFFSET(I+1) + SPLIT(I+1) / 2.
      IF(OFF END .GE. 100.)  OFF END = OFF END - 100.
      WRITE(6,300) (NAME(J,I+1),J=1,3), OFF BEG, OFFSET(I+1), OFF END,
     1 SPLIT(I+1), AMBER
      WRITE(6,315) TITLE, IPAGE, SUB TIT
  315 FORMAT(1H1, 8X, 10A4, 25X, 4HPAGE, I3 / 9X, 10A4 )
      WRITE(6,260)
      WRITE(6,270)    MESS
  320 CONTINUE
      WRITE(6,305) X MARK, OFF BEG, OFF END, DIR 1
      I = NO SIGS
      OFF BEG = OFF PB(2,I) - EFF 2 / 2.
      IF(OFF BEG .LT. 0.)  OFF BEG = OFF BEG + 100.
      OFF END = OFF PB(2,I) + EFF 2 / 2.
      IFF(OFF END .GE. 100.)  OFF END = OFF END - 100.
      X MARK = BLANK
```

```
      IF(SPLIT(I) - .01 .LE. EFF 2)  X MARK = ASTK
      G = 0.
      H = 0.
      O = OFF BEG
      P = OFF END
      Q = 0.
      WRITE(6,310) X MARK, OFF BEG, OFF END, DIR 2
      IF(IPCH .EQ. ITES) WRITE(7,311) (NAME(J,I),J=1,3),A,B,C,D,E,F,G,
     1  H,O,P,Q,ITEST
  325 IF(IWORD .EQ. NO)  GO TO 330
      GO TO 10
  330 WRITE(6,335)
  335 FORMAT(1H1, 1X )
      CALL EXIT
      STOP
      END
```

INPUT INFORMATION

KANSAS STATE UNIVERSITY CIVIL ENGINEERING DEPARTMENT

1	NAME OF SYSTEM	05 RADICS ROAD SIGNAL SYSTEM			
2	SUB-TITLE	ANDERSON AVENUE			
3	NUMBER OF SIGNALS IN SYSTEM	5			
4	MIN AND MAX CYCLES (SECS)	35.00	65.00		
5	SUGGESTED MAX SPEED TOL.(MPH)	0.0			
6	SYSTEM OFFSET TRANSPOSITION (CCT)	0.0			
7	COLUMN HEADINGS	DIR. 1	DIR. 2	NAME/PED-X	SPLIT/CI
8	DIRECTION IDENTIFICATION	E-BOUND	W-BOUND		
9	BAND WIDTH PROPORTIONMENT	50.00	50.00		
A	LOCATION NAME + PHASE SPLIT (PCT)			SUNSET	56.00
B	PED X-ING + CLEARENCE TIMES (SECS)			15.00	3.00
C	TRAVEL DISTANCE (FEET)	953.00	953.00		
D	PROGRESSIVE SPEED DESIRED (MPH)	30.00	30.00		
A	LOCATION NAME + PHASE SPLIT (PCT)			DENISON	56.00
B	PED X-ING + CLEARENCE TIMES (SECS)			15.00	3.00
C	TRAVEL DISTANCE (FEET)	639.00	639.00		
D	PROGRESSIVE SPEED DESIRED (MPH)	30.00	30.00		
A	LOCATION NAME + PHASE SPLIT (PCT)			SEVENTEENTH	56.00
B	PED X-ING + CLEARENCE TIMES (SECS)			20.00	3.00
C	TRAVEL DISTANCE (FEET)	1003.00	1003.00		
D	PROGRESSIVE SPEED DESIRED (MPH)	30.00	30.00		
A	LOCATION NAME + PHASE SPLIT (PCT)			CROSSWALK	68.00
B	PED X-ING + CLEARENCE TIMES (SECS)			15.00	3.00
C	TRAVEL DISTANCE (FEET)	395.00	395.00		
D	PROGRESSIVE SPEED DESIRED (MPH)	30.00	30.00		
A	LOCATION NAME + PHASE SPLIT (PCT)			FOURTEENTH	44.00
B	PED X-ING + CLEARENCE TIMES (SECS)			15.00	3.00
10	PROCESSING INSTRUCTIONS TO COMPUTER	RUN			
11	PUNCHED OUTPUT? IF SO, SIZE	NO 0			
12	ADDITIONAL RUNS FOLLOW	YES			

```
CYCLE SCAN FCR BEST FIT

05  RADICS RCAD SIGNAL SYSTEM
ANCERSON AVENUE

EFF = 17.649 AT  35. SECS                        18
EFF = 10.068 AT  40. SECS              10
EFF = 22.505 AT  45. SECS                              23
EFF = 22.364 AT  50. SECS                             22
EFF = 15.099 AT  55. SECS                     15
EFF = 17.841 AT  60. SECS                        18
EFF = 24.161 AT  65. SECS                               24

BEST FIND IS
   CYCLE CF  65.  EFFICIENCY =  24.161

ITERATION IMPRCVEMENTS
             66.000              25.310
             66.500              25.872
             66.750              26.149
             66.875              26.287
             66.938              26.356
             66.969              26.390
             66.984              26.408
             66.992              26.416
             66.996              26.420
             66.998              26.423
```

```
                    TRAFFIC SIGNAL SYSTEM TIMING PLANS

05  RADIOS ROAD SIGNAL SYSTEM             KANSAS STATE UNIVERSITY
ANDERSON AVENUE                           ENGINEERING DEPARTMENT

PLAN  67.0 ( 50- 50)   CYCLE LENGTH (SECONDS)     67.0  65.0  60.0  70.0

   ****************    E-BOUND  DIRECTION
   *              *       BAND WIDTH (SECONDS)  17.7  17.2  15.9  18.5
   *  LISTING IS  *       EFFICIENCY (PER CENT) 26.4  26.4  26.4  26.4
   *  IN E-BOUND  *
   *  DIRECTION   *    W-BOUND  DIRECTION
   *              *       BAND WIDTH (SECONDS)  17.7  17.2  15.9  18.5
   ****************       EFFICIENCY (PER CENT) 26.4  26.4  26.4  26.4

SIGNAL        OFFSETS-PER CENT  INTERVAL  SIGNAL  OPT.  BEST  --OTHERS--
LOCATION      BEG.  MID.  END.  G-PCT.-Y  SPCNG.  PROGRESSIVE SPEEDS-MPH
***********   ***************   ********  ******  ********************

SUNSET       -25.8   0.0  25.8  51.5 4.5

      P.BAND  91.9        18.3  E-BOUND    953.0  30.0  30.9  33.5  28.7
      P.BAND  81.7         8.1  W-BOUND    953.0  30.0  30.9  33.5  28.7

DENISON       24.2  50.0  75.8  51.5 4.5

      P.BAND  24.2        50.7  E-BOUND    639.0  30.0  30.9  33.5  28.7
      P.BAND  49.3        75.7  W-BOUND    639.0  30.0  30.9  33.5  28.7

SEVENTEENTH   24.2  50.0  75.8  51.5 4.5

      P.BAND  45.9        72.3  E-BOUND   1003.0  30.0  30.9  33.5  28.7
      P.BAND  27.7        54.1  W-BOUND   1003.0  30.0  30.9  33.5  28.7

CROSSWALK     68.2 100.0  31.8  63.5 4.5

      P.BAND  79.9         6.4  E-BOUND    395.0  30.0  30.9  33.5  28.7
      P.BAND  93.6        20.0  W-BOUND    395.0  30.0  30.9  33.5  28.7

FOURTEENTH    80.2 100.0  19.8  39.5 4.5

      P.BAND  93.3        19.8  E-BOUND
      P.BAND  80.2         6.6  W-BOUND
```

REFERENCES

1. Bleyl, Robert L., "A Practical Computer Program for Designing Traffic-Signal-System Timing Plans," Highway Research Record Number 211, Washington, D.C. (1967).

2. Bleyl, R.L., "SIGPROG, Progressive Signal System Timing Program," a program instruction report written for the Bureau of Public Roads, Washington, D.C.

3. Chamberlain, Robert N., "Traffic Progression Program."

4. Gerlough, D.L. and Wagner, F.A., "Improved Criteria for Traffic Signals at Individual Intersections--Interim Report," NCHRP Report 3, Washington, D.C. (1964).

5. Gerlough, D.L. and Wagner, F.A., "Improved Criteria for Traffic Signals at Individual Intersections," NCHRP Report 32, Washington, D.C. (1967).

6. Jacks, M., "Optimum Signal System Timing (SIGOP)," a paper presented at the A.S.C.E. Conference on Transportation Demonstration Projects, November 19-20, 1968, in Washington, D.C.

7. Little, J.D.C., Martin, B.V., and Morgan, J.T., Synchronizing Traffic Signals for Maximal Bandwidth, Report $64-08, Department of Civil Engineering, MIT (1964).

8. Wagner, F.A., Gerlough, D.L., and Barnes, F.C., "Improved Criteria for Traffic Signals Systems on Urban Arterials," NCHRP Report 73, Washington, D.C. (1969).

9. Wagner, F.A., Barnes, F.C., and Gerlough, D.L., "Improved Criteria for Traffic Signal Systems in Urban Networks," NCHRP Project 3-5/2, Los Angeles, California (1969).

USE OF COMPUTER PROGRAMMING TO SOLVE TRAFFIC PROGRESSION PROBLEMS

by

GREGORY WARREN HARDIN

B.S., Kansas State University, 1970

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirement for the degree

MASTER OF SCIENCE

Department of Civil Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

The purpose of this paper was to examine the development of various ideas and methods used in traffic signal progression systems. This examination was limited to methods which could be easily applied by traffic engineers.

A secondary purpose of this paper was to explain the corresponding computer programs so that they can easily be used by traffic engineers.

Traffic progression problems are basically either single-street systems or complex networks. Four programs were studied in the report, with three of the four concerned with single streets only.

The first program studied was developed by J.D.C. Little. He proved that each signal system has half-integer synchronization, i.e. that offsets are either at zero or half a cycle. This also made the bandwidths in each direction equal. His process was developed on both the 1620 and 360 computers, and an analysis showed that the two computers gave the same degree of accuracy.

The next program studied was written by Robert Chamberlain. This program looked for the optimum cycle length by first solving for the optimum speed. This speed was the speed yielding the smallest minimum interference, and was found by searching the range of allowable cycle lengths.

The last single-street program studied was SIGPROG, which converted all time and distance units to travel time units. This altered the conventional time-space diagram to a time-travel time diagram. The maximum progressive bandwidths were then found by using the appropriate algorithms on the computer.

The final program examined was SIGOP, which was developed to solve progression problems for an entire network of signals. This program consists of six "blocks," and thus is very lengthy. The program is currently being tried in six pilot cities in the United States and the results have not been completed.

Traffic data were gathered from four cities and used in the report. A comparison of the three single-street programs was made with the data obtained. This comparison indicated that there is very little difference in the three methods. It was recommended that a method be selected which will be the easiest to use for the problem in question.

For the convenience of traffic engineers, the three single-street computer programs are included.