Synchronizing Traffic Signals for Maximal Bandwidth

JOHN D. C. LITTLE and BRIAN V. MARTIN, Respectively, Sloan School of Management and Department of Civil Engineering, Massachusetts Institute of Technology; and
JOHN T. MORGAN, The Australian National University, Canberra City, A. C. T.

Traffic signals can be synchronized so that a car, starting at one end of a street and traveling at preassigned speeds, can go to the other end without stopping for a red light. The portion of a signal cycle for which this is possible is called the bandwidth for that direction. Ordinarily the bandwidth in each direction is single, i. e., is not split into two or more intervals within a cycle. Two problems are solved for this case: (a) given an arbitrary number of signals along a street, a common cycle length, the green and red times for each signal, and specified vehicle speeds in each direction between adjacent signals, synchronize the signals to produce bandwidths that are equal in each direction and as large as possible; and (b) adjust the synchronization to increase one bandwidth to some specified, feasible value and maintain the other as large as is then possible. The method of calculation has been programmed for a 20K IBM 1620.

•TRAFFIC signals prevent chaos at busy intersections, but nobody likes the frequent stops that often occur on streets with many signals. The number of stops can be reduced by proper synchronization of the signals.

Consider a street with a sequence of signals all of which have the same cycle length. The bandwidth along the street will be defined as that portion of a cycle during which a car could start at one end of the street and, by traveling at preassigned speeds (not necessarily all the same), go to the other end without stopping for a red light. Each direction has its own bandwidth. For example, it is an easy matter to synchronize the signals so that, for one direction, a car that passes the first signal just as it turns green passes all others in the same way. We shall call this a complete one-way synchronization. The bandwidth for that direction is as large as possible and equals the shortest of the green times of the signals on the street. Bandwidth in the other direction, however, is likely to be small or zero, unless the distances between signals are particularly fortuitous. Signals synchronized to create a substantial bandwidth are called progression systems.

It is possible to construct examples where the bandwidth in a single cycle in a single direction is split up into two or more intervals separated by very short reds. Since it seems rather unlikely that split bandwidths would often occur in practice, and since the extension of results to cover these cases appears rather cumbersome, we restrict ourselves unless otherwise stated to problems for which the maximal bandwidths are unsplit.

Procedures are given for solving the following two problems. Problem 1: given a common cycle length, green splits for each signal, and specified speeds in each direction between adjacent signals, determine offsets for the signals, so as to produce bandwidths which are equal in each direction and as large as possible. Problem 2: adjust

Paper sponsored by Committee on Traffic Control Devices.

the offsets to favor one direction with a larger bandwidth, if feasible, and give the other direction the largest bandwidth then possible.

The paper is divided into three parts. After an introduction, the first section discusses the background of the problem and describes briefly a computer program that calculates the desired offsets. The second section develops the mathematical theory underlying the solution. The third section describes the computer program and its operation in detail.

BACKGROUND AND COMPUTER PROGRAM

Background

The objective of maximizing bandwidth has an intuitive appeal and is widely used. A more obvious criterion might be trip delay, but almost any kind of synchronization that treats the street as a whole leads to the concept of a planned speed. Once specified, the planned speed tends to determine trip delay (1), unless input flow exceeds street capacity, in which case delay is determined mostly by the amount and duration of the overload. Changes in synchronization tend to produce changes in trip delay which in terms of percentage are small. The stops themselves may be more irritating than the delay. However, the driver's trade-off between stops and delay does not seem to have been much investigated.

In any case, increases in bandwidth usually tend to decrease both stops and delay. For example, in von Stein's (2, 3) approach to traffic control, drivers are encouraged by various signaling devices to form compact platoons which travel nonstop through the system at a preset speed. Insofar as this is successful, trip delay is fixed by the speed. The bandwidth determines the maximal platoon size for which stops can be avoided. A stop forces a driver back into the following platoon with a delay of some fraction of a period. Therefore, the objective studied here is that of maximizing main street bandwidths subject to the constraints imposed by service for the cross streets, pedestrian crossings, etc. For further discussion of signal synchronization and for other approaches to the problem, see Newell (12, 13) and Grace and Potts (14).

The literature on bandwidth contains a number of methods, mostly graphical, for solving special cases of Problem 1. Matson, Smith and Hurd (4) consider primarily signals with constant spacing. Bruening (5) and Petterman (6) approach the problem by trial and error. Raus (7) treats a limited class of problems algebraically.

Bowers (8) gives a graphical method for maximizing bandwidth when the green times are all the same and speed is a constant. His standard procedure involves solving the problem for a range of (speed) × (period) and identifying those values which yield the largest bandwidth as a percentage of period. Evans (9) presents Bowers' method. Davidson (10) also uses this method, but redefines the problem slightly by taking the bandwidth for the main street as given and seeking to maximize the smallest percentage of green assigned to any cross street. This criterion determines green splits for a few critical signals with the rest given the largest cross street green consistent with the specified main street bandwidth. The resulting synchronization is the same as that of Bowers' method.

Our method solves the foregoing cases and handles two generalizations which have not, to our knowledge, been handled previously in any formal way: (a) arbitrary planned speeds are permitted in either direction between any two adjacent signals; and (b) a device is given for apportioning bandwidth between directions on the basis of platoon size. In addition, the method is designed for machine computation and has been programmed for an IBM 1620.

Computer Program

The calculation of offsets to give maximal bandwidth is an easy job, thanks to computers. A program, called TSS3, has been written for a 20K IBM 1620. The machine language object deck* will run on any basic 1620 installation. If the installation has a

*Obtainable from Director, Civil Engineering Systems Laboratory, M.I.T., Cambridge, Massachusetts, 02139.

Cal-Comp Digital Plotter available for use on a line, a further program, TSS4, will take the output of TSS3 and plot a space-time diagram for the final signal settings.

The data required to operate the program are as follows: number the signals 1, 2, 3,... in the direction of increasing distance from an origin at one end of the street. This direction will be called outbound, the opposite direction inbound. To find the maximal equal bandwidths, the program requires as input: (a) number of signals, (b) cycle length (sec), (c) distance of each signal from chosen origin (ft), (d) red phase of each signal (sec), and (e) vehicle speed in each direction between each pair of adjacent signals (mph). For the case of unequal volumes in the two directions, the program will adjust the bandwidths to favor the heavy volume direction. For this purpose, the program requires: (a) inbound volume (veh/hr), (b) outbound volume (veh/hr), and (c) headway between vehicle (sec).

The output of the program consists of: (a) offsets for each signal with respect to a reference signal, (b) number of reference signal, (c) inbound and outbound bandwidths, and (d) largest volumes that will fit unimpeded through the inbound and outbound green bands. The program also produces certain other information useful to the plotting program.

The treatment of volumes is based on the idea of platoons. A given volume and cycle length together imply some number of vehicles per cycle through each signal. Under suitable conditions these vehicles move as a fairly compact platoon through the system. The average headway between vehicles determines the time-length of the platoon. The computer program tries to arrange bandwidths so that both inbound and outbound platoons fit into their green bands. However, a number of special cases come up. Whenever the two platoons are equal, equal bandwidths are given each direction. If the sum of the two bandwidths is greater than the sum of the two platoon lengths, the individual bandwidths are made proportional to platoon lengths, as far as possible. If the sum of the bandwidths is less than the sum of the platoon lengths, the larger platoon is accommodated, if possible, and, thereafter, as much bandwidth as can be arranged is given to the direction with the smaller platoon. The final results are summarized by printing out the inbound and outbound volumes that would be obtained by putting through the largest platoons that fit unimpeded into the green bands.

The time to solve a 10-signal problem is only about a minute. Thus it is a reasonable task to explore a range of cycle lengths to look for particularly large bandwidths or to make sensitivity tests on other constants of the system.

Applications of the method have been made in Cleveland and, more recently, by Hesketh (15) outside Providence.

THEORY

Definitions and Notation

Consider a two-way street having n traffic signals. Directions on the street will be identified as outbound and inbound. The signals will be denoted S_1, S_2, ..., S_n with the subscript increasing in the outbound direction. Let

- C = cycle length of the signals (sec);
- r_i = red time of S_i on street under study (cycles);
- b $(\bar{b})$ = outbound (inbound) bandwidth (cycles);
- t_{ij} $(\bar{t}_{ij})$ = travel time from S_i to S_j in the outbound (inbound) direction (cycles); and
- θ_{ij} = relative phase, or offset, of S_i and S_j, measured as time from center of a red of S_i to next center of red of S_j (cycles); by convention $0 \leq \theta_{ij} < 1$. (See Fig. 1.)

Any time quantity can be expressed in cycles by dividing by C. "Red time" is used as shorthand for "unusable time." A set of θ_{ij}, j = 1, ..., n for any i will be called a synchronization of the signals.

Travel times between adjacent signals are presumed known and fixed. Then all t_{ij} may be calculated from the following:

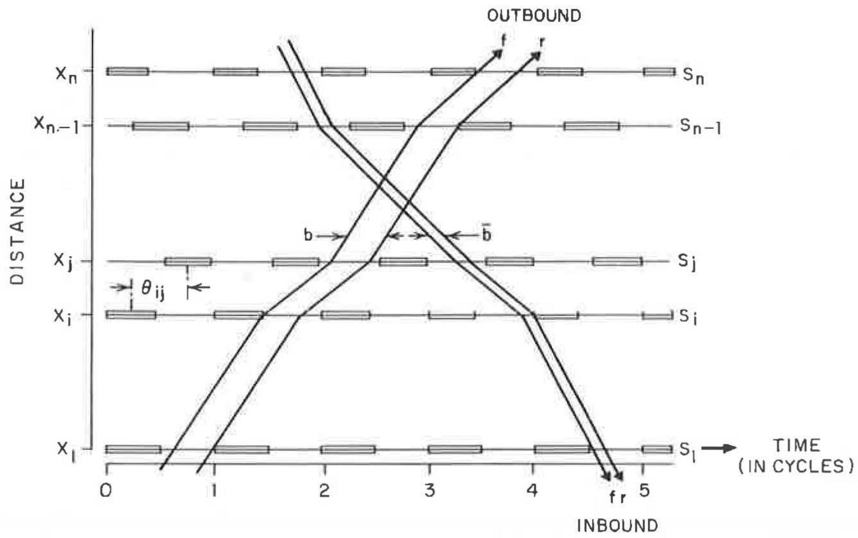


Figure 1. Space-time diagram showing outbound and inbound green bands; signals S_1 and S_i are critical signals.

$$t_{ij} = \begin{cases} \displaystyle\sum_{k=i}^{j-1} t_{k,\,k+1} & j > i \\ 0 & j = i \\ \displaystyle -\sum_{k=j}^{i-1} t_{k,\,k+1} & j < i \end{cases}$$

and all $\bar{t}_{ij}$ from corresponding expressions with each t replaced by $\bar{t}$. Although t_{ij} and $\bar{t}_{ij}$ are the basic inputs to the calculation, it is frequently more convenient to think in terms of speeds and distances. Let

x_i = position of S_i on the street (ft), and
v_i ($\bar{v}_i$) = outbound (inbound) speed between S_i and S_{i+1} (ft/sec).

Then

$$t_{i,\,i+1} = \frac{x_{i+1} - x_i}{v_i C}$$

$$\bar{t}_{i,\,i+1} = \frac{x_i - x_{i+1}}{\bar{v}_i C} \qquad (1)$$

Most previous work has assumed $v_i = \bar{v}_i = v$, in which case $t_{ij} = -\bar{t}_{ij} = (x_j - x_i)/vC$, but this work is not so restricted.

Figure 1 shows a space-time diagram for travel on the street. Heavy horizontal lines indicate when the signals are red. The zig-zag lines represent trajectories of cars passing unimpeded along the street in the directions indicated. Changes in slope correspond to changes in speed. The set of possible unimpeded trajectories in a given direction forms a green band whose horizontal width is the bandwidth for that direction.

The trajectory forming the front edge (earlier in time) of a band and the one forming the rear edge (later in time) have been marked f and r, respectively. Although the green bands are only drawn once, they appear once per cycle in parallel bands across the diagram.

Basis for Method

The basis for the method is developed in a sequence of lemmas and theorems. Before starting, let us examine the objectives. We want to maximize bandwidths but there are bandwidths in each direction, b and $\bar{b}$. We could maximize $b + \bar{b}$, but possibly this would produce an undesirable division of the total between b and $\bar{b}$; for example, one of them might be zero. To unravel the situation, consider the following three problems:

(1) Max $(b + \bar{b})$.
(2) Max $(b + \bar{b})$ subject to $b = \bar{b}$.
(3) Max $(b + \bar{b})$ subject to $b > 0$ and $\bar{b} > 0$.

This work shows that there is usually a whole class of synchronizations which solve (3) and, of these, at least one solves (2). Moreover, the max $(b + \bar{b})$ found in (2) and (3) is a constant which can, within certain limits, be divided arbitrarily between b and $\bar{b}$. However, in some cases, the constant will be less than the $(b + \bar{b})$ found in (1). The reason is fairly simple. Under sufficiently awkward red times and signal spacings the max $(b + \bar{b})$ of (2) and (3) can become quite small, even zero. On the other hand, no matter how awkward the spacing, we can always set up a complete one-way synchronization and obtain a $(b + \bar{b})$ at least as large as the smallest green time.

In any case, this work solves all three problems. The central problem is (2), which will be called the problem of finding maximal equal bandwidths and is solved by theorem 3. Theorem 4 expresses the solution of all three problems in what seems to be an operationally useful way.

Definition. —A signal S_j is said to be a critical signal if one side of S_j's red touches the green band in one direction and the other side touches the green band in the other direction. Thus, in Figure 1, signals S_1 and S_j are critical, but no others are.

Lemma 1. —If a synchronization maximizes $(b + \bar{b})$ subject to $b > 0$ and $\bar{b} > 0$, then:

(a) There exists at least one critical signal.

(b) The red time of any critical signal will touch the front edge of one green band and the rear edge of the other.

(c) All critical signals can be divided into two groups: Group 1 consists of signals whose reds touch the front of outbound and the rear of inbound and Group 2 of signals whose reds touch the front of inbound and the rear of outbound.

Proof. —Consider the set of signals whose reds touch a given side of the green band in one direction. Part (a) must be true or else all these signals could be shifted to in-

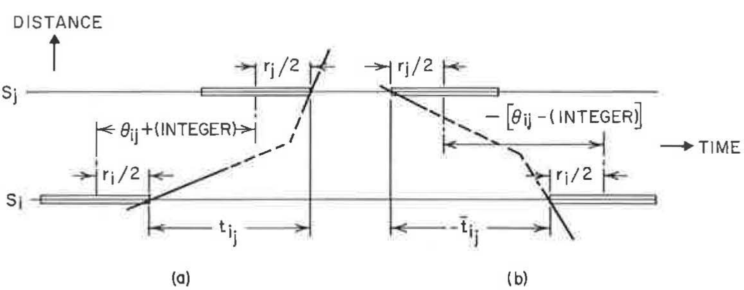


Figure 2. Geometry when two Group 1 signals limit the green band.

crease bandwidth in the one direction without reducing it in the other. Part (b) is a consequence of the definition of critical signal: since the right side of red can only touch front edges and the left side only rear edges, a critical signal must touch (at least) one of each.

Part (c) follows immediately from (b) since there are only two choices. Possibly a signal fits into both groups, in which case it will be considered to be in both. Possibly there is only one critical signal, but then it fits into both groups. This completes the proof.

Suppose two signals, S_i and S_j, are in the same group, for instance, 1. For each signal, the right-hand side of red touches the front of the outbound band and left-hand side touches the rear of the inbound band. Figure 2 shows the geometry for this situation. The quantities are presented in such a way that, if $j > i$, all the lengths shown are positive. The notation "integer" is used to indicate that some integer is to be added to an expression to make it valid.

From Figure 2a:

$$\frac{1}{2} r_i + t_{ij} = \frac{1}{2} r_j + \theta_{ij} + \text{(integer)}$$

From Figure 2b:

$$\frac{1}{2} r_i - \bar{t}_{ij} = \frac{1}{2} r_j - \theta_{ij} + \text{(integer)}$$

Consequently:

$$\theta_{ij} = \frac{1}{2} (t_{ij} + \bar{t}_{ij}) + \frac{1}{2} \text{(integer)} \tag{2}$$

Corresponding arguments lead to the same equation for Group 2. By convention, $0 \leq \theta_{ij} < 1$. Therefore, it may be seen that (2) has two solutions for θ_{ij}, to be found by adding whatever half integers will bring $(\frac{1}{2})$ $(t_{ij} + \bar{t}_{ij})$ into the required range.

A more explicit expression for the two possible values of θ_{ij} can be developed. Let

$$\delta_{ij} = 0 \text{ or } \frac{1}{2}$$

man z = mantissa of z, as obtained by removing the integral part of z and, if the result is negative, adding unity

Thus, man (5.2) = 0.2, man (-0.2) = 0.8, and in general $0 \leq \text{man } z < 1$. Now (2) becomes

$$\theta_{ij} = \text{man} \left[\frac{1}{2} (t_{ij} + \bar{t}_{ij}) + \delta_{ij}\right] \tag{3}$$

The phasing represented by (3) will be called half-integer sychronization. The term can be consistently applied to a collection of signals. In other words, given a set $\delta_{i1}, \delta_{i2}, \ldots, \delta_{in}$, the resulting $\{\theta_{ij}\}$ have the property that $\theta_{ik} = \text{man}(\theta_{ij} + \theta_{jk})$. Furthermore, the same θ_{ik} is obtained by setting $\delta_{ik} = \text{man}(\delta_{ij} + \delta_{jk})$ in (3). The above summarizes into:

Lemma 2.—Under the conditions of lemma 1, each group of signals has half-integer synchronization.

The operational meaning of half-integer synchronization is easiest understood in the special case $t_{ij} = -\bar{t}_{ij}$, which occurs, for example, when speeds are the same in each direction. Then (3) gives $\theta_{ij} = 0$ or $\frac{1}{2}$ so that any two signals in the same group have the centers of their reds exactly in phase or exactly out of phase.

Theorem 1.—There is a half-integer synchronization which gives maximal equal bandwidths.

Proof (by construction). —Suppose we have a set of phases such that $(b + \bar{b})$ is maximal subject to $b > 0$ and $\bar{b} > 0$. (If none exists, the theorem is trivially true.) Divide the critical signals into Groups 1 and 2. Extend the reds of all other signals until they are critical, too, but not so far as to reduce bandwidth. The old reds lie wholly within the new. Move the center of the old red to the center of the new—this cannot extend the new red or change bandwidth. Classify the new critical signals into Groups 1 and 2. Change the pases of all Group 1 signals by an equal amount in the direction that will decrease the larger of $\bar{b}$ and b. The loss to the larger is just equaled by a gain to the smaller so that $(b + \bar{b})$ stays constant. Choose the amount of change so that $b = \bar{b}$.

Within each group there is half-integer synchronization. It remains to show that there is now half-integer synchronization between signals from different groups. Let S_i be from Group 1 and S_j from Group 2. Figure 3 shows that

$$\frac{1}{2} r_i + b + t_{ij} + \frac{1}{2} r_j = \theta_{ij} + (\text{integer}) \tag{4}$$

$$\frac{1}{2} r_i + \bar{b} + \bar{t}_{ij} + \frac{1}{2} r_j = -\left[\theta_{ij} - (\text{integer})\right] \tag{5}$$

But $b = \bar{b}$, whence

$$\theta_{ij} = \frac{1}{2} (t_{ij} + \bar{t}_{ij}) + \frac{1}{2} (\text{integer})$$

which is (2) again and so implies that S_i and S_j have half-integer synchronization.

In the foregoing we have also proved the following:

Corollary 1.—If the maximal equal bandwidths are greater than zero, max $(b + \bar{b})$ subject to $b > 0$ and $\bar{b} > 0$ equals max $(b + \bar{b})$ subject to $b = \bar{b}$.

Theorem 2. —Under any half-integer synchronization, $b = \bar{b}$.

Proof. —It suffices to consider critical signals. Let S_i be from Group 1 and S_j in Group 2. Figure 3 applies as do (4) and (5). Subtract (5) from (4) and substitute (2). It will be seen that $b = \bar{b}$.

Two special cases of Theorem 1 deserve separate mention. From (1) comes:

Corollary 2. —If speeds are the same in each direction at each point of the street, maximal equal bandwidths are achieved by a synchronization in which each θ_{ij} is either 0 or ½.

Explicit results are possible in the two signal case:

Corollary 3. —If there are only two signals, S_i and S_j, and speeds are the same in each direction, maximal equal bandwidths are achieved by

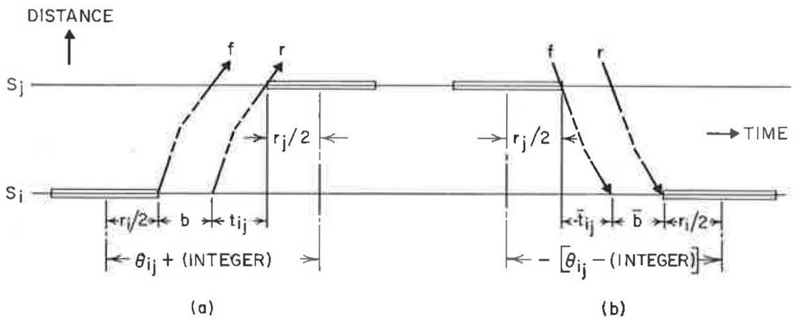


Figure 3. Geometry when a Group 1 and a Group 2 signal limit the green bands.

$$\theta_{ij} = \begin{cases} 0 & 0 \le \text{man}\left[\frac{x_j - x_i}{vC}\right] < \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} \le \text{man}\left[\frac{x_j - x_i}{vC}\right] < \frac{3}{4} \\ 0 & \frac{3}{4} \le \text{man}\left[\frac{x_j - x_i}{vC}\right] < 1 \end{cases}$$

Notice that θ_{ij} does not depend on the green splits of the two signals. This is not always true with more than two signals. Corollary 3 may be proved by constructing a space-time diagram with S_i at $x = 0$ and the start of S_i's green at $t = 0$. S_j's position may be varied along an x-axis where $x = (x_j - x_i)/vC$. At each x there are, by Corollary 2, only two possibilities for placing the center of S_j's red and the best one is fairly obvious.

Synchronization for Maximal Equal Bandwidths

The significance of the results of the previous section is that a synchronization for maximal equal bandwidths can be found by searching through a relatively few cases. By Theorem 1, it suffices to examine half-integer synchronizations. By Theorem 2, it suffices to examine only the outbound direction.

b_i = greatest outbound bandwidth under half-integer synchronization if S_i's red touches the front of the outbound band; and
B = the value of one of the maximal equal bandwidths.

It will be helpful in computations to permit b_i and B to be negative at times; the operational interpretation as a zero bandwidth is clear.

If S_i's red touches the front of the outbound green band, the situation is as shown in Figure 3a. Take as an origin for measurements the right side of S_i's red. The trajectory (not shown) that touches the right side of S_j's red passes S_i at a time which will be denoted u_{ij}. Figure 3a shows that

$$\text{man}\left[\theta_{ij} + \frac{r_j}{2} - \frac{r_i}{2} - t_{ij}\right]$$

except that, when this expression is zero, we shall want $u_{ij} = 1$. This may be accomplished by writing

$$u_{ij} = 1 - \text{man}\left[-\theta_{ij} - \frac{r_j}{2} + \frac{r_i}{2} + t_{ij}\right]$$

Substituting (3) and making the dependence of δ_{ij} explicit:

$$u_{ij}(\delta_{ij}) = 1 - \text{man}\left[\frac{1}{2}(r_i - r_j) + \frac{1}{2}(t_{ij} - \bar{t}_{ij}) - \delta_{ij}\right]$$

The trajectory that touches the left side of S_j's red passes S_i at $u_{ij} - r_j$. Therefore, since δ_{ij} is to take on either the value 0 or ½ and since S_i's red is to touch the front, the best δ_{ij} is identified by

$$\max\left[u_{ij}(0) - r_j,\ u_{ij}\left(\frac{1}{2}\right) - r_j\right]$$

Therefore

$$b_i = \min_{j} \max_{\delta = 0, \frac{1}{2}} \left[u_{ij}(\delta) - r_j\right]$$

and, finally,

$$B = \max_{i} b_i$$

Summarizing, we have the following:

Theorem 3. —The maximal equal bandwidth is max (0, B) where

$$B = \max_{i} \min_{j} \max_{\delta = 0, \frac{1}{2}} \left[u_{ij}(\delta) - r_j\right]$$

Let i = c be a maximizing i and $\delta_{c1}, \ldots, \delta_{cn}$ be the corresponding maximizing δ's. Then, a synchronization for maximal equal bandwidths is $\theta_{c1}, \ldots, \theta_{cn}$ obtained by substituting the δ_{cj} into (3).

Maximal Unequal Bandwidths

Average platoon lengths usually differ between the inbound and outbound directions. If the length exceeds bandwidth in one direction and not the other, it may be possible to shift bandwidth from one direction to the other and pass both platoons. We first show how to shift bandwidth and then suggest a method for dividing total bandwidth between directions on the basis of platoon size.

Let $\theta_{c1}, \ldots, \theta_{cn}$ be a maximal equal bandwidth synchronization with S_c a critical signal whose red touches the front of the outbound green band. The corresponding $u_{c1}, \ldots, u_{cn}$ are presumed known as is the maximal equal bandwidth, B. Let

α_j = a phase shift for S_j (cycles),

θ'_{cj} = man $(\theta_{cj} - \alpha_j)$ = adjusted phase for S_j (cycles), and

$g = \min_{i} (1 - r_i)$ = smallest green time (cycles)

The shifting procedure is described in Theorem 4.

Theorem 4. —The outbound bandwidth, b, can be assigned any value, max [0, B] ≤ b ≤ g, by making a phase shift.

$$\alpha_j = \max \left[u_{cj} - 1 + b - B,\ 0\right]$$

Then $\bar{b}$ = max [2B - b, 0], and $\bar{b}$ is as large as possible for the given b.

Alternatively, the inbound bandwidth, $\bar{b}$, can be assigned any value, max [0, B] ≤ $\bar{b}$ ≤ g, by making a phase shift:

$$\alpha_j = \max \left[\bar{b} + r_j - u_{cj},\ 0\right]$$

Then b = max [2B - $\bar{b}$, 0], and b is as large as possible for the given $\bar{b}$.

The shifting procedure may be developed as follows: suppose it is desired to increase outbound bandwidth to b > B, or, if B is negative, to b > 0. The trajectory at the front edge of the outbound band is moved to the left, pushing before it any reds that start to touch it. (See Fig. 4, or the change from Fig. 5 to 6.) During the movement, the critical signals will cut down $\bar{b}$ just as much as b is increased, except that, if $\bar{b}$ reaches zero, no further decrease can occur. Thus, by Corollary 1, $\bar{b}$ is as large as possible for the given b. There is a limit to the increase that can be made in b because eventually the pushing of a red to the left will bring the next red of that signal in from the right to cut into the rear of the outbound band. Then that signal limits both front and rear of the band. The signal must be one with the smallest value of green time; therefore b = g. From this argument we conclude that b can be increased from max [0, $\bar{B}$] to any value less than or equal to g and that $\bar{b}$ is then max [B - (b - B), 0]. Analogous remarks apply to increasing $\bar{b}$.

The algebra of the shift may be worked out from Figure 4. Define a u-axis which measures right and left from the front edge of the outbound green band under the given maximal equal bandwidth synchronization. The rear of the band is then at u = B and the right side of a red of S_j is at $u = u_{cj}$.

Consider first the shift to obtain b. The front of the outbound band is pushed left to the position u = B - b. (See the dashed line in the outbound portion of Fig. 4.) This will

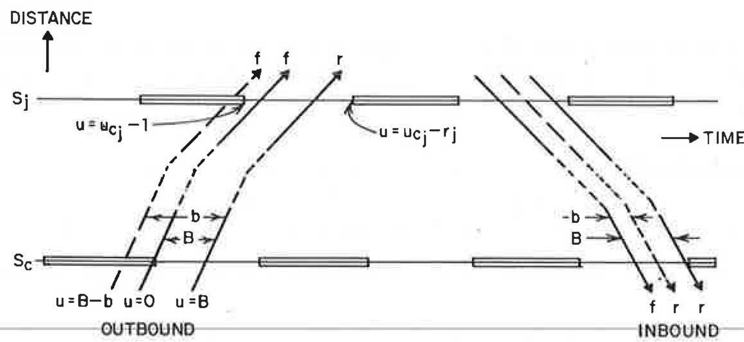


Figure 4. Widening the outbound green band from B to b.

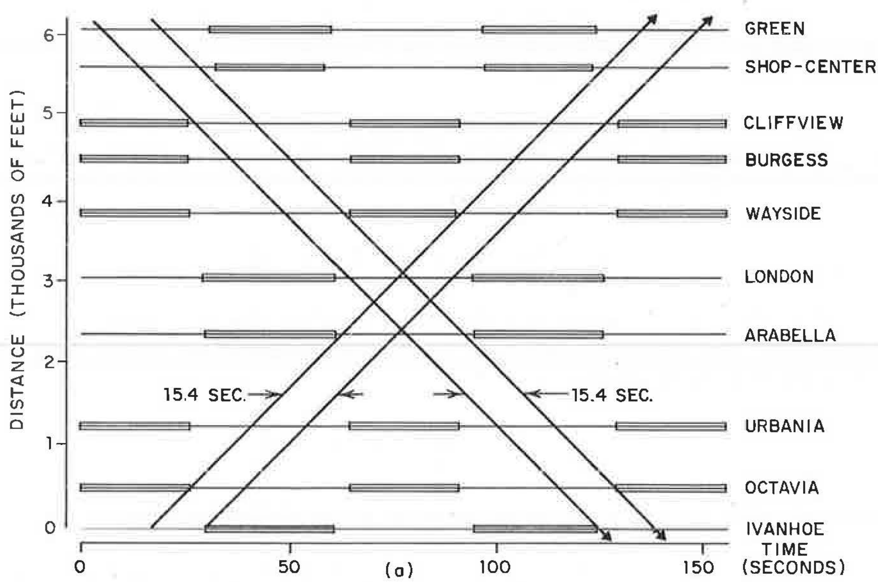


Figure 5. Space-time diagram for part of Euclid Avenue in Cleveland. C = 65 sec, $v = \bar{v} = 50$ ft/sec, $P = \bar{P}$.

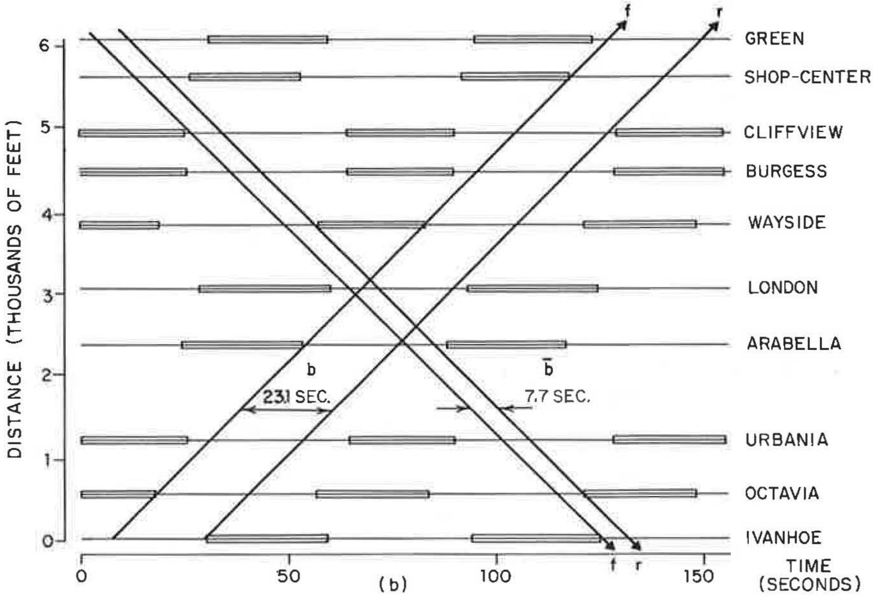


Figure 6. Space-time diagram for part of Euclid Avenue in Cleveland. $C = 65$ sec, $v = \bar{v} = 50$ ft/sec, $P = 0.3$ cycles, $\bar{P} = 0.1$ cycles.

require moving some reds but no more will be moved than necessary and those moved will be moved as little as possible. The next S_j red to the left of the old front edge is met at $u = u_{cj} - 1$. Therefore, the appropriate phase shift for S_j is to the left by an amount:

$$\alpha_j = \max \left[(u_{cj} - 1) - (B - b),\ 0\right]$$

For the case of shifting to obtain $\bar{b}$, it is first observed that under the given synchronization, as under any half-integer synchronization, the distance from the front of the inbound green band to the next S_j red on the left is the same as the distance from the rear of the outbound green band to the next S_j red on the right. (Otherwise we could contradict Theorem 2 by enlarging r_j until S_j started to reduce one green band and not the other.) Consequently we can calculate how much to shift S_j by seeing what is required to move the rear of the outbound band to the right and make it $\bar{b}$ wide. From Figure 4, we find that the magnitude of the shift should be

$$\alpha_j = \max \left[\bar{b} - (u_{cj} - r_j),\ 0\right]$$

To move the front of inbound to the left, these shifts are made to the left. This concludes the proof of Theorem 4. For completeness, if $g > 2B$, $\max (b + \bar{b}) = g$; otherwise $\max (b + \bar{b})$, $\max (b + \bar{b})$ subject to $b = \bar{b}$, and $\max (b + \bar{b})$ subject to $b > 0$ all equal $2B$.

Finally, we give a way to apportion total bandwidth between directions on the basis of the length (in time) of the platoons. Let

P ($\bar{P}$) = platoon length in the outbound (inbound) direction (cycles).

Whenever $P = \bar{P}$ maximal equal bandwidths are proposed. Otherwise, we proceed as follows: If $P + \bar{P} \leq 2B$, there may be enough bandwidth to accommodate both platoons. The bandwidth is made proportional to platoon length if possible. Thus, if $P > \bar{P}$:

$$\begin{aligned} b &= \min\,[2BP/(P + \bar{P}),\ g] \\ \bar{b} &= \max\,[2B - b,\ 0] \end{aligned}$$

If $P + \bar{P} > 2B$, the larger platoon is accommodated, if possible, and the remainder, if any, is given to the smaller. Thus, if $P > \bar{P}$,

$$\begin{aligned} b &= \min\,(P,\ g) \\ \bar{b} &= \max\,(2B - b,\ 0) \end{aligned}$$

except that if $\bar{b} = 0$, b is set to g. Appropriate interchanges apply if $\bar{P} > P$.

Summary of Method

To synchronize signals for maximal equal bandwidths, first number of signals in order of distance along the street, say, $i = 1, 2, \ldots, n$. The direction of increasing i will be called outbound. Next specify the following data: the signal period, C, in seconds; the red times, $r_1, \ldots, r_n$, in fractions of a cycle; the signal positions, $x_1, \ldots, x_n$, in ft; the outbound speeds between signals, $v_1, v_2, \ldots, v_{n-1}$, in ft/sec; and the inbound speeds between signals $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{n-1}$ in ft/sec.

The computation proceeds in the following steps:

1. Calculate $y_1, \ldots, y_n$ from

$$\begin{aligned} y_1 &= 0 \\ y_i &= y_{i-1} - \frac{1}{2}(r_i - r_{i-1}) + (x_i - x_{i-1})\frac{1}{2C}\left[\frac{1}{v_{i-1}} + \frac{1}{\bar{v}_{i-1}}\right] \end{aligned}$$

2. Calculate $z_1, \ldots, z_n$ from

$$\begin{aligned} z_1 &= 0 \\ z_i &= z_{i-1} + (x_i - x_{i-1})\frac{1}{2C}\left[\frac{1}{v_{i-1}} - \frac{1}{\bar{v}_{i-1}}\right] \end{aligned}$$

3. Calculate

$$B = \max_i \min_j \max_{\delta = 0, \frac{1}{2}} [u_{ij}(\delta) - r_j]$$

where

$$u_{ij}(\delta) = 1 - \operatorname{man}(y_j - y_i - \delta)$$

and the operation "man" is as defined earlier. Consider a specific i. As the max over δ is performed, the maximizing value (one for each j) may be recorded in a temporary table, $\delta_{i1}, \ldots, \delta_{in}$. As the max over i is performed, the maximizing i, say $i = c$, identifies the best set, $\delta_{c1}, \ldots, \delta_{cn}$, which is saved. For the following computations, it is necessary to save the set, $u_{c1}, \ldots, u_{cn}$, corresponding to the $\delta_{c1}, \ldots, \delta_{cn}$. This means saving the value of u_{ij} whenever a value of δ_{ij} is saved.

4. A synchronization, $\theta_{c1}, \ldots, \theta_{cn}$, for maximal equal bandwidths is calculated from

$$\theta_{cj} = \operatorname{man}\,[z_j - z_c + \delta_{cj}]$$

The bandwidth in each direction is max (0, B).

To adjust the synchronization for platoon lengths of P, outbound, and $\bar{P}$, inbound, specify P and $\bar{P}$, perform the foregoing calculations and continue as follows.

5. Calculate $g = \min\,(1 - r_i)$.

6. If $P = \bar{P}$ accept equal bandwidth solution.

7. If $\bar{P} > P$, go to Step 11, otherwise continue.

8. If $P + \bar{P} \leq 2B$, set $b = \min[g, 2BP/(P + \bar{P})]$. Otherwise, set $b = \min(P, g)$; unless $P \geq 2B$, in which case, set $b = g$.
9. Calculate $\alpha_1, \ldots, \alpha_n$ from $\alpha_j = \max(u_{cj} - 1 + b - B, 0)$.
10. Calculate $\bar{b} = \max(2B - b, 0)$. Go to Step 14.
11. If $P + \bar{P} \leq 2B$, set $\bar{b} = \min[g, 2B\bar{P}/(P + \bar{P})]$. Otherwise, set $\bar{b} = \min(\bar{P}, g)$, unless $\bar{P} \geq 2B$, in which case, set $\bar{b} = g$.
12. Calculate $\alpha_1, \ldots, \alpha_n$ from $\alpha_j = \max(\bar{b} + r_j - u_{cj}, 0)$.
13. Calculate $b = \max(2B - \bar{b}, 0)$
14. The adjusted synchronization, $\theta'_{c1}, \ldots, \theta'_{cn}$, is calculated from

$$\theta'_{cj} = \text{man}\,(\theta_{cj} - \alpha_j)$$

and the bandwidths are b, outbound and $\bar{b}$, inbound as previously determined.

For plotting space-time diagrams it is helpful to know where the edges of the green bands are. Take as a reference point the center of a red of S_c. The left side of an outbound band is at $r_c/2$, the right side at $(r_c/2) + b$. An inbound band has its left side at $1 - (r_c/2) - \bar{b}$ and its right side at $1 - (r_c/2)$. The edges of the same outbound band at S_j are found by adding t_{cj} to the edges at S_c; for inbound, add $\bar{t}_{cj}$.

Examples

The method has been used to synchronize the signals on a stretch of Euclid Avenue in Cleveland under off rush-hour conditions. (During rush hours a complete one-way synchronization is used.) Signals are at 0, 550, 1250, 2350, 3050, 3850, 4500, 4900, 5600, 6050 ft. Corresponding red times are 0.47, 0.40, 0.40, 0.47, 0.48, 0.42, 0.40, 0.40, 0.40, 0.42 cycles. C = 65 sec, v = 50 ft/sec in both directions. Figure 5 shows the space-time diagram for maximal equal bandwidths. B = 0.237 or 15.4 sec. Figure 6 shows the case: P = 0.30 cycles, $\bar{P}$ = 0.10 cycles.

Discussion

The ability to handle different platoon speeds between different signals make it possible to adjust the synchronization for the presence of a queue waiting at a signal. Such a queue might arise because turning traffic is entering the main street at the previous intersection or might be the tail of a platoon that does not fit through the green band. Let τ be the time length of the queue waiting when the next platoon tries to come through. Unless the queue is released early, the arriving platoon will have to stop or slow down. Let v be the normal platoon speed and x the distance from the preceding signal. If the speed, $v' = v/[1 - (\tau/x)\,v]$, is used in the computation, the synchronization will permit the platoon to travel at v and not stop (but note that the departing platoon is longer than the arriving platoon). Similarly, an allowance can be made for cars leaving the platoon, although, unless the cars always leave from the head, it may be more reasonable to expect (or guide) the lead car to maintain a planned speed and encourage subsequent cars to adjust speed to close the gaps. A negative value of v' is permitted by our calculation; this would imply a backward movement of the green wave.

Although we have ruled out problems for which the maximal bandwidth is split in one or both directions, our results extend to one aspect of these cases. Denote the largest unsplit segment of a bandwidth as the primary bandwidth and the corresponding green band as the primary green band. The substitution of these terms for bandwidth and green band throughout the analysis will make the results hold for any problem. In most problems our method is likely to maximize total bandwidth even if split, but it is possible to construct examples where this does not happen.

The method may be useful in connection with traffic control by on-line computer. The α_j adjustment could be made to follow actual flow. Another possibility is to vary C. Suppose that the green split restrictions are expressed in cycles and remain valid for a range of C. From (1) it may be seen that, if each speed is multiplied by a con-

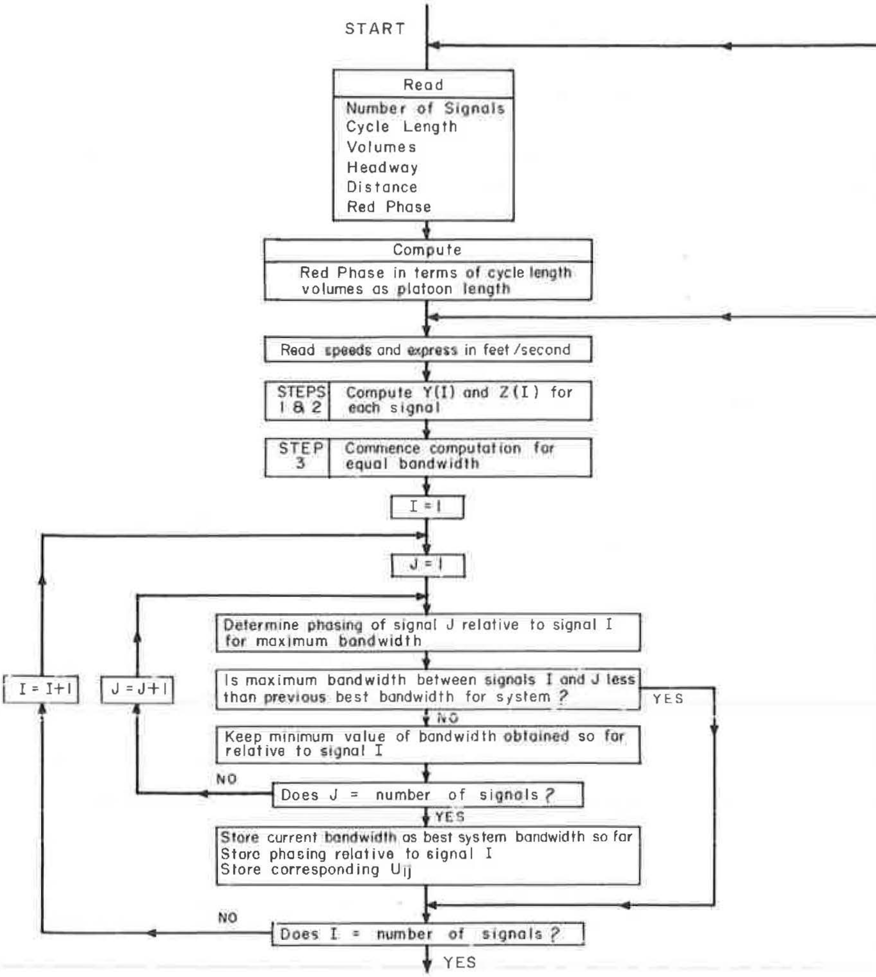


Figure 7. Flow chart of Program TSS3.

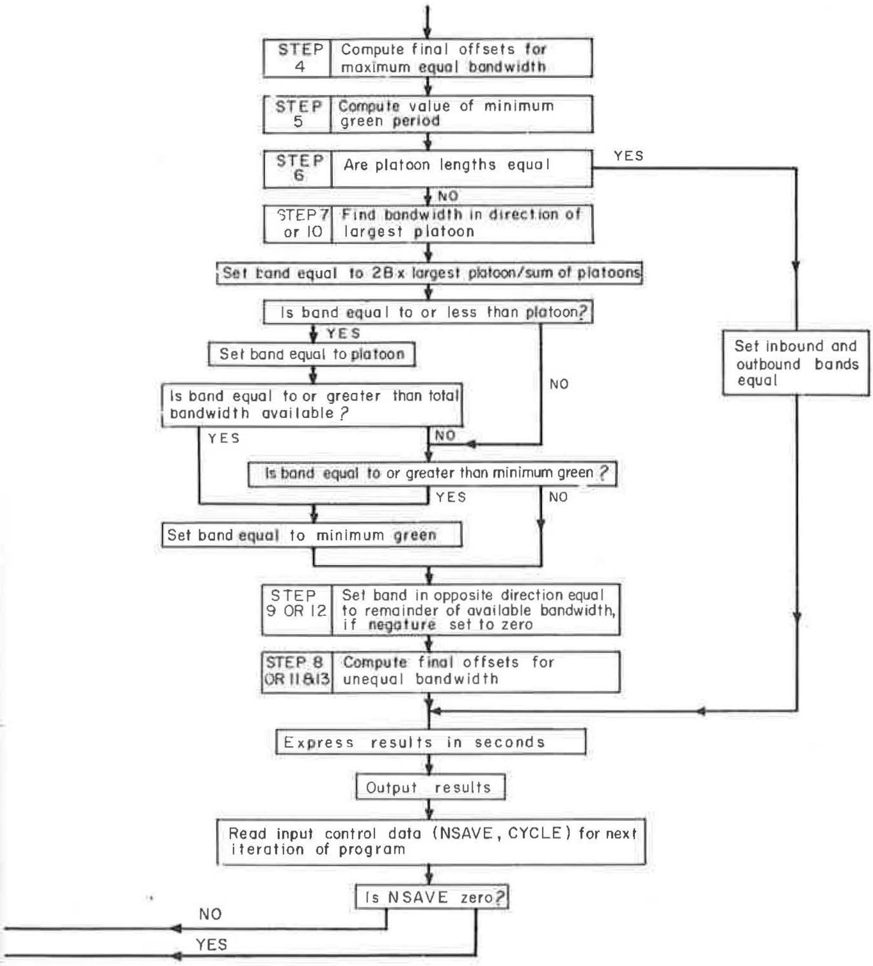
stant and C is divided by that constant, the travel times (expressed in cycles) between signals are unchanged. Therefore, the synchronization for maximum bandwidth is unchanged. Suppose, then, that in real operations the traffic speed temporarily decreases from the planned speed because of weather, increased vehicle density, or the like. Normally, the synchronization becomes invalid, but this will not happen if the signal period is correspondingly lengthened, as could easily be done in real time control. Such lengthening would probably increase capacity slightly.

COMPUTER PROGRAMS

Description of Programs

Program TSS3. —Program TSS3* performs all the necessary computation to determine the offsets of the signals and the green bandwidths. The program is written in LOAD

*A listing of the source program and a storage map are available (at cost of xerox reproduction and handling) for both Program TSS3 and Program TSS4—Supplement XS-7 (Highway Research Record 118), 12 pages.



Note: Steps refer to summary in text

and GO FORTRAN, a system developed by the Civil Engineering Systems Laboratory for the IBM 1620 computer at the Massachusetts Institute of Technology. The program has been limited to 20K storage so that the smallest IBM 1620 installation can use it. The machine language object deck, which is available from MIT, will run on any basic 1620 installation. A flow chart of the program is shown in Figure 7.

Program TSS4. —Program TSS4* takes the output of program TSS3 and plots a space-time diagram, using the Cal-Comp Digital plotter on line. The program is also written in LOAD and GO FORTRAN; it does, however, use two subroutines, PLT, XNM, that are not generally available. These routines are in the object deck which can be used on any 20K installation having a digital plotter on line.

The flow chart (Fig. 8) shows the structure of the program.

*See footnote on preceding page.

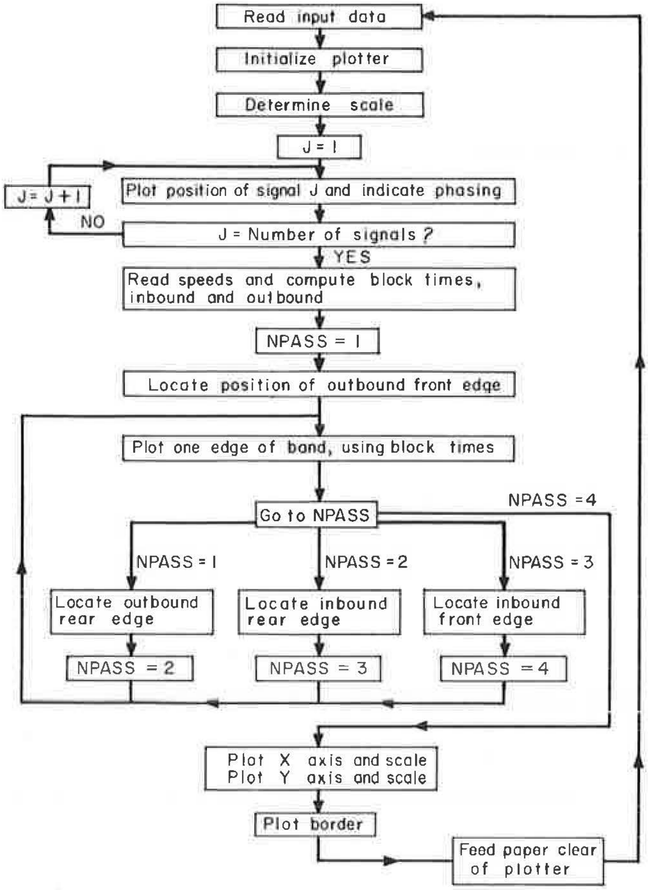


Figure 8. Flow chart of Program TSS4.

Operating Details

Program TSS3

INPUT. All of the data are input from punched cards. There are four types of cards used as follows:

Card type 1, system parameters;
Card type 2, signal characteristics;
Card type 3, speeds; and
Card type 4, control card.

The data included on each card are summarized in Table 1. In the LOAD and GO system there is no format; data must, however, be separated by blank columns. Blank columns are not interpreted as zero in the LOAD and GO system.

Card Type 1. Card contains the data indicating the number of signals in the system, the cycle length, the inbound and outbound volumes and the vehicle headway. The vehicle headway is used to convert the hourly volumes into platoon lengths (sec), as follows:

$$\text{Platoon length} = \frac{\text{Hourly volume} \times \text{vehicle headway}}{3{,}600}$$

TABLE 1

INPUT DATA TO PROGRAM TSS3

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

For streets with several lanes the engineer may use either hourly lane volumes or adjust headways accordingly.

Card Type 2. Each card contains two entries, the first being the x value; that is, the distance of the signal in feet from the origin, taken as the first signal. The second entry indicates the length of the red phase for the signal in seconds. Increasing x is defined as the outbound direction: type 2 cards must be in order of increasing x.

Card Type 3. Each card contains two entries, the first giving the block speed inbound and the second the block speed outbound. The cards must be in order of increasing x, i.e., outbound.

Card Type 4. Card is used to direct the program operations for the following problem, if any. If a completely new problem is to be run, or if the volumes on the previous problem are to be changed, set: NSAVE = positive integer, and CYCLE = any value. The program will complete the current problem and return to start a new problem by reading card type 1. Note that a value for CYCLE must be given; if it is left blank the program will take the first data item on the following card. The program can be terminated by trying to read a card type 1 or a card type 4.

If it is desired to rerun the same physical problem, i.e., distances, red and volumes, but changes must be made in cycle length and/or speeds, set: MSAVE = 0, CYCLE = value wanted. The program will complete the current problem and return to read card type 3 and another card type 4.

PROGRAM OPERATION. The program does not require any special sense switch settings. Running times depend on the particular problem. However, Table 2 indicates running times obtained with the program on problems with 10, 20, 30, 40 and 50 signals, and these may be considered as generally representative.

Card type 4 can be used to make several runs with the same set of signals and volumes but with different cycle lengths or speeds. If the volumes are to be changed or a different problem is to be run the NSAVE entry on card type 4 should be positive and CYCLE may be set to any value, but some value must be given.

OUTPUT. All output is by punched cards, there being five types of cards. Table 3 summarizes the arrangement of the output cards.

Card Type 1. The entries correspond to the input entries for number of signals, cycle length and vehicle headway.

Card Type 2. The first two entries correspond to the input values of inbound and outbound volumes. The last two entries indicate the inbound and outbound volumes possible through the computed green bandwidths for the input vehicle headway.

Card Type 3. The first entry indicates the inbound bandwidth in seconds; the second entry is the outbound bandwidth in seconds. The final entry gives the number of a critical signal, where the signals are numbered in the outbound direction.

Card Type 4. Entries 1 and 2 correspond to the entries on input card type 2. Entry 3 indicates the time in seconds from the origin to the right hand side of red for each signal, where the origin is taken as the center of red for the critical light (final entry on card type 3). This information is useful for plotting the space-time diagram by hand. The final entry indicates the offset of the signal with respect to the critical signal, given in terms of cycle length.

Card Type 5. Card is principally of use to program TSS4. The first entry is the position of the front edge of the outbound band and the second entry is the position of the rear edge of the inbound band, at the first signal. The third and fourth entries are the total system travel time, inbound and outbound, respectively.

Program TSS4

INPUT. All of the data are input from punched cards. There are two different sources for the input data, the first being the output from program TSS3 and the second the speed data as input to program TSS3 on cards type 3. Table 4 summarizes the input data to program TSS4.

TABLE 2

RUNNING TIMES WITH PROGRAM TSS3
(All entries in seconds)

Number of Signals	Read Object Deck	Read Data	Computations	Total
10	30	10	20	60
20	30	20	25	75
30	30	30	75	135
40	30	40	110	180
50	30	50	200	280

TABLE 3

OUTPUT DATA FROM PROGRAM TSS3

Entry	Description of Data Item	Units
Card type 1		
1	Number of Signals	Integer
2	Cycle Length	Seconds
3	Vehicle Headway	Seconds
	ONE CARD PER PROBLEM	
Card type 2		
1	Inbound Volume (input)	Veh/Hour
2	Outbound Volume (input)	Veh/Hour
3	Possible Inbound Volume through band	Veh/Hour
4	Possible Outbound Volume through band	Veh/Hour
	ONE CARD PER PROBLEM	
Card type 3		
1	Inbound bandwidth	Seconds
2	Outbound bandwidth	Seconds
3	Number of Restricting Signal	Integer
	ONE CARD PER PROBLEM	
Card type 4		
1	Distance of Signal from Origin	Feet
2	Red Phase of Signals	Seconds
3	RHS of red phase	Seconds
4	Offset relative to restricting signal	Fraction of Cycle Length
	NUMBER OF CARDS = NUMBER OF SIGNALS	
Card type 5		
1	Front edge of outbound band	Fraction of
2	Rear edge of inbound band	Cycle Length
3	Total system travel time inbound	Fraction of
4	Total system travel time outbound	Cycle Length
	ONE CARD PER PROBLEM	

PROGRAM OPERATION. The program does not require any special sense switch settings. The width of the plot has been scaled so that it will always be 8 in. wide, the convenient plotting width of the 12-in. Cal-Comp plotter. The horizontal scale of the plot will always be 20 sec/in. Running times for the program depend on the problem and exact hardware configuration. However, plots will normally take between 5 and 10 min.

OUTPUT. The only output is the plot, examples of which are shown.

TABLE 4

INPUT DATA TO PROGRAM TSS4

Entry	Description of Data Item	Units
1	Data cards exactly as output from program TSS3	
2	Block speed inbound	MPH
	Block speed outbound	MPH
	(Exactly as input card type 3)	
	NUMBER OF CARDS = NUMBER OF SIGNALS-1	

TABLE 5

SAMPLE PROBLEM

Signal Number	X Value Feet	Red Phase Seconds	Block Speed MPH	
			Inbound	Outbound
1	000	30.5	30.0	30.0
2	550	26.0	30.0	30.0
3	1250	26.0	30.0	30.0
4	2350	30.5	50.0	50.0
5	3050	31.0	50.0	50.0
6	3850	27.0	50.0	50.0
7	4500	26.0	40.0	40.0
8	4900	26.0	40.0	40.0
9	5600	26.0	40.0	40.0
10	6050	27.0		

Restrictions

Program TSS3 is limited to problems with 50 signals or less, by the 20K storage restriction imposed on the program. Block speeds may be either positive or negative but not zero. All other variables should be positive.

Program TSS4 is limited to problems with 20 signals or less, if the smaller Cal-Comp plotter is being used. For the larger Cal-Comp plotter 50 signals could be accommodated in the 20K storage, although plots are likely to be very long and consequently slower. Program TSS4 will accept any output from Program TSS3. Zero bandwidth is plotted as one straight line.

TABLE 6 SAMPLE PROBLEM INPUT TO PROGRAM TSS3

10	65.0	400.	400.	2.
0.	30.5			
550.	26.0			
1250.	26.0			
2350.	30.5			
3050.	31.0			
3850.	27.0			
4500.	26.0			
4900.	26.0			
5600.	26.0			
6050.	27.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
50.0	50.0			
50.0	50.0			
50.0	50.0			
40.0	40.0			
40.0	40.0			
40.0	40.0			
1	1.			
10	65.0	850.	0.	2.
0.	30.5			
550.	26.0			
1250.	26.0			
2350.	30.5			
3050.	31.0			
3850.	27.0			
4500.	26.0			
4900.	26.0			
5600.	26.0			
6050.	27.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
50.0	50.0			
50.0	50.0			
50.0	50.0			
40.0	40.0			
40.0	40.0			
40.0	40.0			
1	1.			
10	65.0	600.	200.	2.
0.	30.5			
550.	26.0			
1250.	26.0			
2350.	30.5			
3050.	31.0			
3850.	27.0			
4500.	26.0			
4900.	26.0			
5600.	26.0			
6050.	27.0			
30.0	30.0			
30.0	30.0			
30.0	30.0			
50.0	50.0			
50.0	50.0			
50.0	50.0			
40.0	40.0			
40.0	40.0			
40.0	40.0			
1	1.			

TABLE 7 SAMPLE PROBLEM OUTPUT FROM PROGRAM TSS3

10	65.000000	2.0000000		001
400.00000	400.00000	324.75528	324.75528	002
11.727274	11.727274	7		003
.00000000	30.499999	47.750000	.50000000	004
550.00000	26.000000	45.500000	.50000000	005
1250.0000	26.000000	13.000000	.00000000	006
2350.0000	30.499999	47.750000	.50000000	007
3050.0000	31.000000	48.000000	.50000000	008
3850.0000	27.000000	46.000000	.50000000	009
4500.0000	26.000000	13.000000	.00000000	010
4900.0000	26.000000	13.000000	.00000000	011
5600.0000	26.000000	13.000000	.00000000	012
6050.0000	27.000000	13.500000	.00000000	013
.92727280	2.0727272	1.6791956	1.6791956	014
10	65.000000	2.0000000		015
600.00000	200.00000	600.00000	49.510566	016
21.666666	1.7878816	7		017
.00000000	30.499999	47.750000	.50000000	018
550.00000	26.000000	35.560608	.34708627	019
1250.0000	26.000000	13.000000	.00000000	020
2350.0000	30.499999	47.750000	.50000000	021
3050.0000	31.000000	48.000000	.50000000	022
3850.0000	27.000000	39.196968	.39533796	023
4500.0000	26.000000	13.000000	.00000000	024
4900.0000	26.000000	13.000000	.00000000	025
5600.0000	26.000000	11.583338	.97820520	026
6050.0000	27.000000	3.9128900	.85250600	027
.92727280	2.0727272	1.6791956	1.6791956	028
10	65.000000	2.0000000		029
850.00000	.00000000	941.53860	.00000000	030
34.000005	.00000006	7		031
.00000000	30.499999	35.727264	.31503484	032
550.00000	26.000000	23.227269	.15734260	033
1250.0000	26.000000	7.3181750	.91258730	034
2350.0000	30.499999	47.318175	.49335654	035
3050.0000	31.000000	37.772725	.34265730	036
3850.0000	27.000000	26.863629	.20559429	037
4500.0000	26.000000	13.000000	.00000000	038
4900.0000	26.000000	11.181814	.97202790	039
5600.0000	26.000000	64.249998	.78846150	040
6050.0000	27.000000	56.579550	.66276230	041
.92727280	2.0727272	1.6791956	1.6791956	042

Program Availability

Machine language object decks for both programs TSS3 and TSS4 can be obtained from the Department of Civil Engineering at MIT. Program TSS3 can be punched from the source listing given in this report and compiled with FORTRAN compilers without format. However, on 20K machines memory overflows may occur if trace instructions are also included at compilation time. Program TSS4 contains plotting routines not generally available. The source listing is included for users who may wish to use their own routines. In program TSS4 the plotting routines are as follows: PLT for drawing lines and moving pen; and XNM for plotting numbers.

Sample Problem

The sample problem consists of a system of 10 signals with the characteristics given in Table 5. The input cards to program TSS3 are given in Table 6; the output from this program is given in Table 7. The input to program TSS4 is listed in Table 8. The plotted output from program TSS4 is shown in Figures 9, 10 and 11. The system parameters are as follows: cycle length, 65 sec; headway, 2 sec.

TABLE 8 SAMPLE PROBLEM INPUT TO PROGRAM TSS4

10	65.000000	2.0000000		001
400.00000	400.00000	324.75528	324.75528	002
11.727274	11.727274	7		003
.00000000	30.499999	47.750000	.50000000	004
550.00000	26.000000	45.500000	.50000000	005
1250.0000	26.000000	13.000000	.00000000	006
2350.0000	30.499999	47.750000	.50000000	007
3050.0000	31.000000	48.000000	.50000000	008
3850.0000	27.000000	46.000000	.50000000	009
4500.0000	26.000000	13.000000	.00000000	010
4900.0000	26.000000	13.000000	.00000000	011
5600.0000	26.000000	13.000000	.00000000	012
6050.0000	27.000000	13.500000	.00000000	013
.92727280	2.0727272	1.6791956	1.6791956	014
30.0	30.0			
30.0	30.0			
30.0	30.0			
50.0	50.0			
50.0	50.0			
50.0	50.0			
40.0	40.0			
40.0	40.0			
40.0	40.0			
10	65.000000	2.0000000		015
600.00000	200.00000	600.00000	49.510566	016
21.666666	1.7878816	7		017
.00000000	30.499999	47.750000	.50000000	018
550.00000	26.000000	35.560608	.34708627	019
1250.0000	26.000000	13.000000	.00000000	020
2350.0000	30.499999	47.750000	.50000000	021
3050.0000	31.000000	48.000000	.50000000	022
3850.0000	27.000000	39.196968	.39533796	023
4500.0000	26.000000	13.000000	.00000000	024
4900.0000	26.000000	13.000000	.00000000	025
5600.0000	26.000000	11.583338	.97820520	026
6050.0000	27.000000	3.9128900	.85250600	027
.92727280	2.0727272	1.6791956	1.6791956	028
30.0	30.0			
30.0	30.0			
30.0	30.0			
50.0	50.0			
50.0	50.0			
50.0	50.0			
40.0	40.0			
40.0	40.0			
40.0	40.0			
10	65.000000	2.0000000		029
850.00000	.00000000	941.53860	.00000000	030
34.000005	.00000000	7		031
.00000000	30.499999	35.727264	.31503484	032
550.00000	26.000000	23.227269	.15734260	033
1250.0000	26.000000	7.3181750	.91258730	034
2350.0000	30.499999	47.318175	.49335654	035
3050.0000	31.000000	37.772725	.34265730	036
3850.0000	27.000000	26.863629	.20559429	037
4500.0000	26.000000	13.000000	.00000000	038
4900.0000	26.000000	11.181814	.97202790	039
5600.0000	26.000000	64.249998	.78846150	040
6050.0000	27.000000	56.579550	.66276230	041
.92727280	2.0727272	1.6791956	1.6791956	042
30.0	30.0			
30.0	30.0			
30.0	30.0			
50.0	50.0			
50.0	50.0			
50.0	50.0			
40.0	40.0			
40.0	40.0			
40.0	40.0			

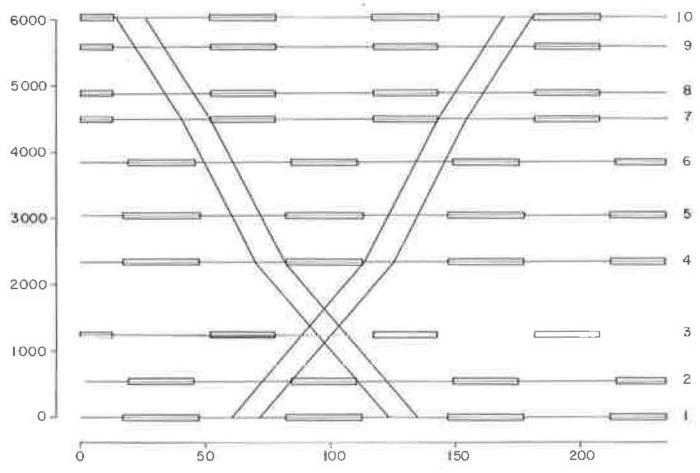


Figure 9. Input volumes: inbound, 40 veh/hr; outbound, 400 veh/hr.

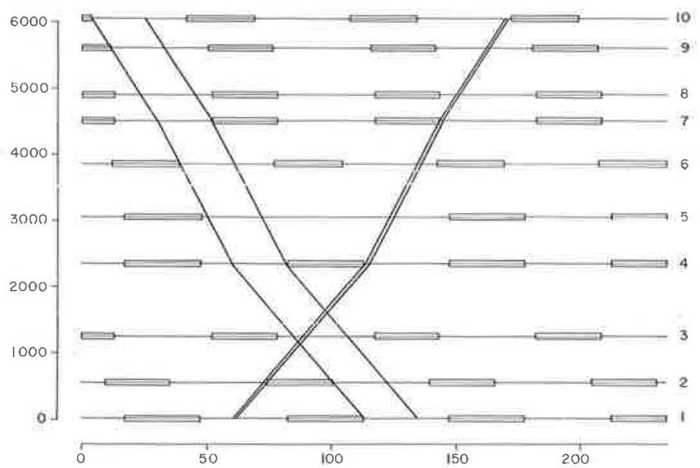


Figure 10. Input volumes: inbound, 600 veh/hr; outbound, 200 veh/hr.

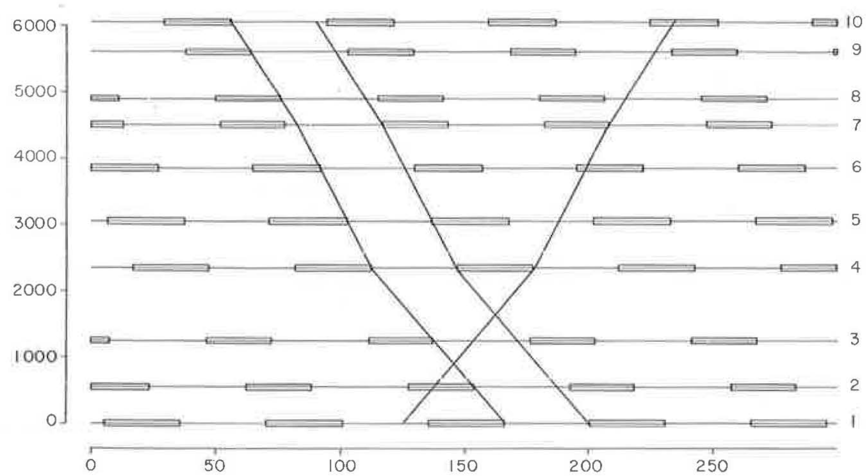


Figure 11. Input volumes: inbound, 850 veh/hr; outbound, 0 veh/hr.

Computations are to be made in volume combinations as follows:

Inbound Volume	Outbound Volume
400 (veh/hr)	400 (veh/hr)
600 (veh/hr)	200 (veh/hr)
850 (veh/hr)	0 (veh/hr)

REFERENCES

1. Morrison, H. M., Underwood, A. F., and Bierley, R. L. Traffic Pacer. Highway Research Board Bull. No. 338, pp. 40-67, 1962.
2. Von Stein, W. Traffic Flow with Pre-Signals and the Signal Funnel. In Theory of Traffic Flow, Herman, R., ed., Amsterdam, Elsevier, 1961.
3. Von Stein, W. From the Laws of Traffic Flow to Traffic Cybernetics. Presented to 2nd Internat. Symposium on the Theory of Road Traffic Flow, June 1963.
4. Matson, T. M., Smith, W. S., and Hurd, F. W. Traffic Engineering. New York, McGraw-Hill, 1955.
5. Bruening, M. E. How Are Progressive Signals Timed. 1940 Proc., ITE, pp. 79-91.
6. Petterman, J. L. Timing Progressive Signal Systems. Traffic Engineering, Vol. 17, Pt. I, pp. 194-199, Feb. 1947; Pt. II, pp. 242-249, Mar. 1947.
7. Raus, J. A Method of Computing Offset Patterns for Multi-Cycle Signal Systems. Traffic Engineering, Vol. 33, pp. 25-28, May 1963.
8. Bowers, D. A. Progressive Timing for Traffic Signals. 1947 Proc., ITE, pp. 93-100.
9. Evans, H. K., ed. Traffic Engineering Handbook. ITE, 2nd Ed., New Haven, 1950.
10. Davidson, B. M. Design of Signal Systems by Graphical Solutions. Traffic Engineering, Vol. 31, pp. 22-28, Nov. 1960.
11. Little, J. D. C., Martin, B. V., and Morgan, J. T. Synchronizing Traffic Signals for Maximal Bandwidth. Rept R64-08, Civ. Eng. Dept., M. I. T., Mar. 1964.
12. Newell, G. F. The Flow of Highway Traffic Through a Sequence of Synchronized Traffic Signals. Operations Research, Vol. 8, pp. 390-405, May 1960.
13. Newell, G. F. Synchronization of Traffic Lights for High Flow. Quarterly of Applied Mathematics, Vol. 21, pp. 315-324, Jan. 1964.
14. Grace, M. J., and Potts, R. B. The Diffusion of Traffic Platoons. Operations Research, Vol. 12, pp. 255-275, Mar. 1964.
15. Hesketh, F. A. Remarks on 'Synchronizing Traffic Signals for Maximal Bandwidth.' Rhode Island Dept. of Public Works, Div. of Roads and Bridges, Dec. 1964.

Appendix

DICTIONARY OF VARIABLE NAMES USED IN COMPUTER PROGRAMS

PROGRAM TSS3

Program Variable	Equivalent Variable Used in Formulation of Technique	Description
Variables as first defined		
ALHS	None	See definition in program listing
BAND	B	Bandwidth in one direction
BIN	$\bar{b}$	Inbound bandwidth
BOUT	b	Outbound bandwidth
BRHS	None	See definition in program listing
CYCLE	C	Cycle length of system
HEDWY	None	Vehicle Headway
LTBST	C	Critical light
NSIG	n	Number of Signals
PHASE (I)	δ_{cj}	Relative Offset of Light
PLATI	$\bar{P}$	Inbound platoon length
PLATO	P	Outbound platoon length
RED (I)	r_i	Red phase of Signal
SPEDI (I)	$\bar{v}_i$	Block speed inbound
SPEDO (I)	v_i	Block speed outbound
TIME (I)	None	Used to find time to signal from origin
VOLIN	None	Inbound Volume
VOLOT	None	Outbound Volume
W (I)	u_{ij}	See definition in summary step 3
X (I)	x_i	Distance of signal from first signal
Y (I)	y_i	See definition in summary step 1
Z (I)	z_i	See definition in summary step 2

NOTES

Variables A, B, I, J, K, M, NSAVE, SAVE and WMIN used as indices and temporary storage.
Variables SPEDI (I), SPEDO (I), PHASE (I) are redefined during the computation as follows:

SPEDI (I) used to save values of PHASE (I)
SPEDO (I) used to save values of W (I)
PHASE (I) used to save values of final offsets for equal and unequal bandwidth cases.

PROGRAM TSS4

Variables given where different to meaning in TSS3

Program Variable	Description
IXY (1)	X co-ordinate at start of line
IXY (2)	Y co-ordinate at start of line
IXY (3)	X co-ordinate at end of line
IXY (4)	Y co-ordinate at end of line
RHS	Length of plot in X(time) direction
SCHR	Horizontal scale factor
SCVT	Vertical scale factor
X1, X2, X3	as IXY (1 or 3)
Y1, Y2, Y3	as IXY (2 or 4)
Y (I)	Distance of a signal from first signal

NOTES

Variables I, B, DUMB, YLS, ALS, NSAVE, M, SAVE and NPASS used as indices and temporary storage.