WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

# The Synchronization of Traffic Signals by Mixed-Integer Linear Programming

ulton [illegible]

John D. C. Little

129-65

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

# The Synchronization of Traffic Signals by Mixed-Integer Linear Programming

John D. C. Little

129-65

Sloan School of Management
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

August 15, 1965

Working Paper

Work reported was supported in part by U. S. Army Research Office (Durham) under contract DA-31-214-ARO-D-209 and in part by Project MAC, an MIT research program sponsored by the Advanced Research Projects Agency, Department of Defense, under the Office of Naval Research Contract No. Nonr-4102(01). Reproduction in whole or in part is permitted for any purpose of the United States Government.

# Abstract

Traffic signals can be synchronized so that a car, starting at one end of a main artery 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. For this case we formulate the arterial problem as a mixed-integer linear program: Given (1) an arbitrary number of signals, (2) the fraction of the cycle that is red at each signal, (3) upper and lower limits on each signal period, (4) upper and lower limits on speed each way between adjacent signals, (5) limits on change in speed, and (6) a constant of proportionality between the two bandwidths, find (1) a common signal period, (2) speeds between signals, and (3) the relative phasing of the signals so as to maximize the sum of the bandwidths. Several variants of the problem are also formulated.

A branch and bound algorithm is developed for solving the given mixed-integer linear program by solving a sequence of ordinary linear programs. A 10 signal example is worked out.

The problem of synchronizing a network of signals is also formulated. The resulting program consists of the arterial programs for the individual streets plus a set of further constraints that arise because the streets connect together to form closed loops. The objective function used for the network program is a weighted sum of the bandwidths in each direction on each artery. A 7 signal example is worked out.

## 1. Introduction

Traffic signals can be synchronized so that a car can go from one end of a street to the other without stopping, if the driver maintains the speeds used in setting the signals. The portion of the signal cycle for which this is possible is called the bandwidth for that direction. Synchronizations for large bandwidth are frequently called progressions. Traffic engineers have long set up progressions and they seem to be quite helpful when traffic is light, although possibly of little value when traffic is heavy. The work of Helly and Baker [1] tends to suggest that this is the case. The value of a progression can be enhanced by giving the drivers speed instructions, as has been done by von Stein. [2]

In an earlier paper [3] J. Morgan and the author have developed an algorithm for solving the following two problems of synchronizing traffic signals for a progression on an arterial street, in the case where bandwidths in each direction are unsplit.

1. Given an arbitrary number of signals, a common signal period, the green and red times for each signal, and specified travel times between adjacent signals, synchronize the signals to produce bandwidths that are equal in each direction and as large as possible.
2. Adjust the synchronization to increase one bandwidth to some specified, feasible value and maintain the other as large as is then possible.

Subsequently, R. Oliver pointed out to the author that a two-signal version of Problem 1 could be set up as a linear program. We here expand this idea into a rather general formulation of the maximal bandwidth problem. In the general case we still have a linear

program but unfortunately it is of the mixed-integer type. For Problems 1 and 2 the new formulation offers no advantages and many disadvantages. However, the linear programming format opens up the possibility of solving more general problems involving the introduction of new decision variables and new constraints.

For example, maximal bandwidth calculations as usually performed have a disconcerting feature. On a long street the critical signals that constrict bandwidth may turn out to be very far apart. Then a small change in one of the design speeds along the street is likely, upon re-solving the problem, to result in a different synchronization and a different bandwidth. Yet, drivers do not hold their speeds exactly constant and, indeed, as is well known, tend to adjust their speeds to the signals. Thus, it makes a good deal of sense to let the design speed between signals be a variable, at least, within certain limits. This can easily be done in the linear program.

Another variable that can be introduced explicitly is the signal period. In Problem 1, period is a constant and, although it is not too difficult to use our earlier methods to examine a considerable number of values in an organized way, the continuous variation of the linear program formulation seems preferable.

Perhaps the most interesting development is that the problem of synchronizing signals for a network of streets can be formulated as a mixed-integer linear program. The program for a network consists of arterial programs for individual streets plus additional constraints whenever the arteries connect together to form loops or cycles.

## 2. The Arterial Problem

2.1 Definitions. 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 $S_1, S_2, \ldots, S_n$ with the subscript increasing in the outbound direction.

Figure 1 shows a space-time diagram for travel on a street. Heavy horizontal lines indicate when the signals are red. The zig-zag lines represent trajectories of the cars passing unimpeded along the street in the directions shown. 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. Although drawn but once, the green bands occur once per cycle in parallel bands across the diagram.

Ordinarily the bandwidth in each cycle is single, i.e., is not split into two or more intervals within a cycle. A split bandwidth can, however, occur and special examples have been constructed in which the maximal total bandwidth is made up of two pieces in one or both directions.[4] This possibility will be ignored in the present formulation. The mathematical program to be constructed will maximize the weighted sum of two bandwidths, one taken from each direction without considering the possibility that other pieces may exist.

Certain signals form the ultimate limitation of bandwidth and will be called critical signals. 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 $S_i$ are critical, but no others are.

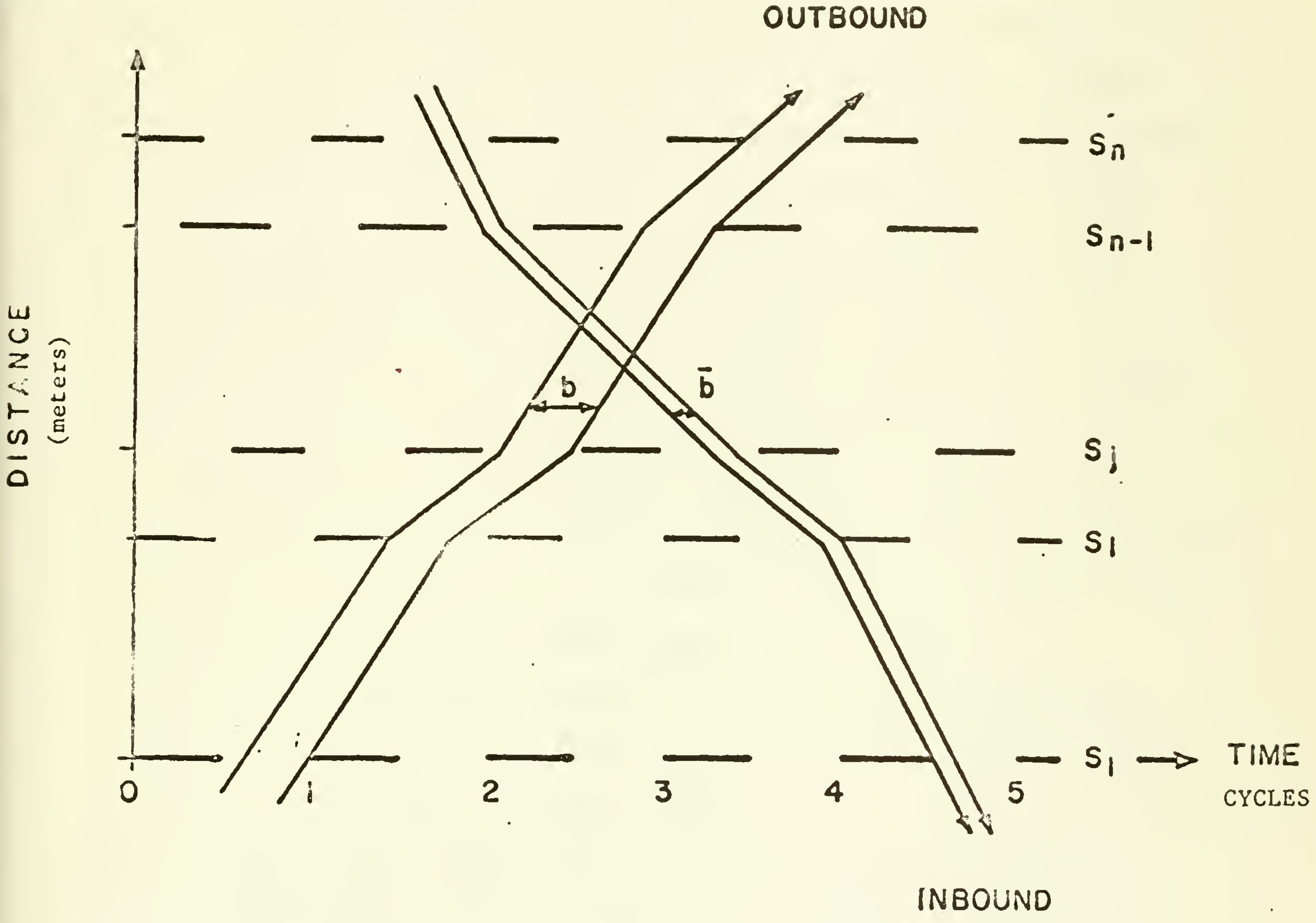

Figure 1. Space-time diagram showing outbound and inbound green bands.

2.3 Basic Maximal Bandwidth Formulation. First we set up Problem 1 above as a mixed-integer linear problem. Let

$r_i$ = red time of $S_i$ on street under study. (cycles)

$b$ ($\bar{b}$) = outbound (inbound) bandwidth. (cycles)

$t(h,i)$ ($\bar{t}(h,i)$) = travel time from $S_h$ to $S_i$ in outbound direction (travel time from $S_i$ to $S_h$ in inbound direction). These quantities are algebraically positive if $S_i$ follows $S_h$ in the outbound direction, otherwise negative. (cycles)

$\phi(h,i)$ ($\bar{\phi}(h,i)$) = distance from center of red at $S_h$ to the center of a particular red at $S_i$. See Figure 2. The two reds are chosen so that each is immediately to the left (right) of the same outbound (inbound) green band. $\phi(h,i)$ ($\bar{\phi}(h,i)$) is positive if $S_i$'s center of red lies to the right (left) of $S_h$'s. (cycles)

$w_i$ ($\bar{w}_i$) = distance from the right (left) side of $S_i$'s red to the green band. See Figure 2. (cycles)

$m(h,i)$ = $\phi(h,i) + \bar{\phi}(h,i)$ (2.1)

Note that a quantity having the dimensions of time can always be expressed in cycles by dividing by the period.

From Figure 2 can be read the identities:

$$(1/2)r_h + w_h + t(h,i) - w_i - (1/2)r_i = \phi(h,i) \qquad (2.2a)$$

$$(1/2)r_h + \bar{w}_h + \bar{t}(h,i) - \bar{w}_i - (1/2)r_i = \bar{\phi}(h,i)\ . \qquad (2.2b)$$

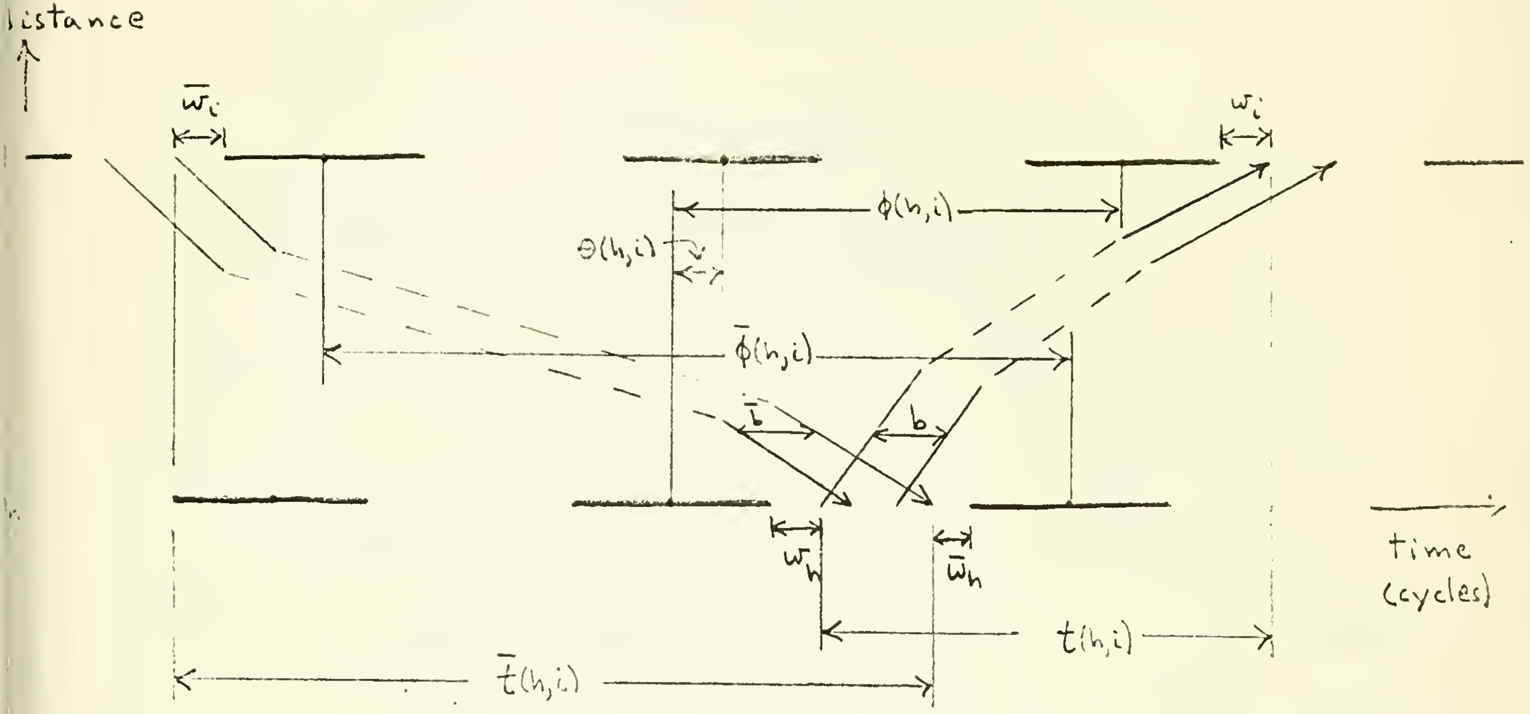

Figure 2. Geometry of green bands. Notice that $\text{Ø}(h,i) + \bar{\text{Ø}}(h,i)$ must equal an integer.

The values of $\emptyset(h,i)$ and $\bar{\emptyset}(h,i)$ have the very important constraint that their sum must be an integer, but otherwise are unrestricted. Therefore, the above two constraints can be replaced by

$$(w_h + \bar{w}_h) - (w_i + \bar{w}_i) + t(h,i) + \bar{t}(h,i) = m(h,i) - (r_h - r_i)$$

(2.3)

$$m(h,i) = \text{integer} .$$

From physical considerations, it follows that

$$t(h,j) = t(h,i) + t(i,j) \quad (2.4a)$$

$$t(h,i) = - t(i,h) \quad (2.5a)$$

Therefore,

$$\emptyset(h,j) = \emptyset(h,i) + \emptyset(i,j) \quad (2.4b)$$

$$\emptyset(h,k) = - \emptyset(i,h) \quad (2.5b)$$

The corresponding inbound expressions are obtained by adding bars to (2.4) and (2.5) above. Then,

$$m(h,j) = m(h,i) + m(i,j) \quad (2.4c)$$

$$m(h,i) = - m(i,h) \quad . \quad (2.5c)$$

From Figure 2 we also see that

$$w_i + b \leq 1 - r_i \quad (2.6a)$$

$$\bar{w}_i + \bar{b} \leq 1 - r_i \ . \quad (2.6b)$$

So far, the expressions apply to any arbitrarily indexed signals. No use has been made of the special numbering along the outbound direction. In later work on networks the present generality will be needed, but we can simplify notation here by defining

$$x_i = x(i, i \div 1) \qquad (2.7)$$

for $x = t, \bar{t}, \emptyset, \bar{\emptyset}, m$ .

Problem 1 will now be represented by a mixed-integer linear program. Constraints (2.6) are needed for each signal and (2.3) for each adjacent pair. (Although there are many other types of pairs, it suffices to consider adjacent pairs since the constraint (2.3) for any other pair can be obtained from linear combinations of (2.3) for adjacent pairs.)

LP1. Find $b, \bar{b}, w_i, \bar{w}_i, m_i$ to

$$\max\ (b + \bar{b}) ,$$

Subject to:

$$b = \bar{b} \qquad \text{(LP1.1)}$$

$$\left.\begin{aligned} w_i + b &\le 1 - r_i \qquad \text{(LP1.2a)} \\ \bar{w}_i + \bar{b} &\le 1 - r_i \qquad \text{(LP1.2b)} \end{aligned}\right\} \quad i = 1,\ldots,n$$

$$(w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (t_i + \bar{t}_i) = m_i - (r_i - r_{i+1}) \qquad \text{(LP1.3)}$$

$$i = 1,\ldots,n-1$$

$$m_i = \text{integer}$$

$$b, \bar{b}, w_i, \bar{w}_i \geq 0$$

LP1 has 3n equations and 3n+ 1 unknowns, not counting slack or artificial variables.

2.4 Adding Decision Variables. Next we let period and speed be variables. Each will be constrained by upper and lower limits. In addition, changes in speed from one street segment to the next will be limited. Finally, instead of requiring equal bandwidths in each direction, we let one bandwidth be a fixed proportion of the other. (However, this is just one of the possible constraining relations that might be used.)

Let

$k$ = constant of proportionality between $b$ and $\bar{b}$.

$T$ = signal period. (seconds)

$T_1, T_2$ = lower and upper limits on period, i.e., $T_1 \leq T \leq T_2$ . (seconds)

$z$ = signal frequency. (cycles/second)

$d(h,i)$ = distance between $S_h$ and $S_i$. (meters)

$d_i$ = $d(i, i+1)$

$v_i$ $(\bar{v}_i)$ = speed between $S_i$ and $S_{i+1}$ outbound ($S_{i+1}$ and $S_i$ inbound). (meters/second)

$e_i, f_i$ $(\bar{e}_i, \bar{f}_i)$ = lower and upper limits on outbound (inbound) speed. (meters/second)

$1/h_i, 1/g_i (1/\bar{h}_i, 1/\bar{g}_i)$ = lower and upper limits on change in outbound (inbound) reciprocal speed, i.e.

$1/h_i \leq (1/v_{i+1}) - (1/v_i) \leq 1/g_i$ .

$(\text{meters/second})^{-1}$

We are constraining change in speed by putting upper and lower limits on change in reciprocal speed. Although the two are not quite the same, constraining the change in reciprocal speed surely satisfies the basic intention, which is to have a means of preventing large abrupt speed changes. The reason for working with reciprocal speed is that it appears linearly in the constraints and can be transformed into $t_i$. Thus

$$t_i = (d_i/v_i)z \, , \quad \bar{t}_i = (d_i/\bar{v}_i)z \, . \tag{2.8}$$

In an expanded program $t_i, \bar{t}_i$, and z are decision variables, which, once known, determine the speeds.

After some algebraic manipulations the constrainsts on bandwidth, period, speed, and change in reciprocal speed can be added to LP1 to give a more versatile mixed-integer linear program.

LP2. Find $b, \bar{b}, z, w_i, \bar{w}_i, t_i, \bar{t}_i, m_i$ to

$$\max\ (b + \bar{b})$$

subject to:

$$\bar{b} = k\ b \tag{LP2.1}$$

$$1/T_2 \le z \le 1/T_1 \tag{LP2.2}$$

$$\left.\begin{aligned} w_i + b &\le 1 - r_i & \text{(LP2.3a)} \\ \bar{w}_i + \bar{b} &\le 1 - r_i & \text{(LP2.3b)} \end{aligned}\right\} \quad i = 1, \ldots, n$$

$$\left.\begin{aligned} (w_i+\bar{w}_i)-(w_{i+1}+\bar{w}_{i+1})+(t_i+\bar{t}_i) &= m_i-(r_i-r_{i+1}) & \text{(LP2.4)} \\ m_i &= \text{integer} & \\ (d_i/f_i)z \le t_i &\le (d_i/e_i)z & \text{(LP2.5a)} \\ (d_i/\bar{f}_i)z \le \bar{t}_i &\le (d_i/\bar{e}_i)z & \text{(LP2.5b)} \end{aligned}\right\} \quad i=1,\ldots,n-1$$

$$\left.\begin{aligned}(d_i/h_i)z \le (d_i/d_{i+1})t_{i+1} - t_i \le (d_i/g_i)z & \qquad \text{(LP2.6a)}\\ (d_i/\bar{h}_i)z \le (d_i/d_{i+1})\bar{t}_{i+1} - \bar{t}_i \le (d_i/\bar{g}_i)z & \qquad \text{(LP2.6b)}\end{aligned}\right\} \; i=1,\ldots,n-2$$

$$b,\ \bar{b},\ w_i,\ \bar{w}_i \ \ge\ 0$$

LP2 involves (11n-10) constraints and 5n variables, not counting slack and artificial variables.

2.5 *Symmetric Problems*. Certain symmetries, if they exist, can be used to reduce the size of LP2. In particular, if the constraints on speed are the same in each direction, and if bandwidths in each direction are required to be equal, the size of the linear program is cut about 40%.

A time-saving technique may be to solve for maximal equal bandwidths and then adjust the synchronization for other cases. Reference 3 shows that, with some qualifications to cover special conditions, max $(b + \bar{b})$ is a constant that can be divided between the directions. Rules are given there for setting one bandwidth to any feasible value and the other to the largest value that is then possible for the given speeds and period.

The linear program for the symmetric case is based on the following two theorems.

Theorem 1. *If the limits on speed and change in speed are the same in each direction, and if LP2 has any optimal solutions, then LP2 has an optimal solution in which* $t_i = \bar{t}_i$ *for each i.*

*Proof*. To have the same limits on speed and change in speed in each direction is to have $e_i = \bar{e}_i$, $f_i = \bar{f}_i$, $g_i = \bar{g}_i$, and $h_i = \bar{h}_i$.

Let $\{t_i, \bar{t}_i\}$ be the set of of t's in an optimal solution for LP2. Define

$$t_i' = \bar{t}_i' = (t_i + \bar{t}_i)/2 , \quad i = 1,\ldots,n-1 .$$

We claim that the $t_i \leftarrow t_i'$ and $\bar{t}_i \leftarrow \bar{t}_i'$ will yield a new optimal solution for LP2. Since the change of t's does not affect the objective function, the only question is feasibility. Clearly, LP2.4 is still satisfied. Adding LP2.5a to LP2.5b and dividing by 2 gives

$$(d_i/f_i)z \leq t_i' \leq (d_i/e_i)z ,$$

so that LP2.5a and b are satisfied. A similar argument shows that LP2.6a and b are satisfied. This completes the proof.

Theorem 2. <u>If the bandwidths in each direction are required to be equal, and if LP2 has any optimal solutions, then LP2 has an optimal solution in which $w_i = \bar{w}_i$ for each i.</u>

<u>Proof</u>. Given any optimal solution to LP2 with $b = \bar{b}$ , replace $w_i$ and $\bar{w}_i$ by $w_i'$ and $\bar{w}_i'$ as defined by

$$w_i' = \bar{w}_i' = (w_i + \bar{w}_i)/2 .$$

The same type of arguments used to Theorem 1 demonstrate the feasibility of the new solution in LP2.3a and b and LP2.4. Optimality is then immediate since the objective function is unchanged.

If the conditions of both theorems are satisfied we can formulate the following simplified mixed-integer linear program;

LP3. Find b, z, $w_i$, $t_i$, $m_i$ to

$$\max b$$

Subject to:

$$(1/T_2) \leq z \leq (1/T_1) \qquad \text{(LP3.1)}$$

$$\left. w_i + b \leq 1 - r_i \right\} \quad i=1,\ldots,n \qquad \text{(LP3.2)}$$

$$\left.\begin{array}{ll} w_i - w_{i+1} + t_i = (1/2)m_i - (1/2)(r_i - r_{i+1}) & \text{(LP3.3)} \\ \qquad m_i = \text{integer} & \\ (d_i/f_i)z \leq t_i \leq (d_i/e_i)z & \text{(LP3.4)} \end{array}\right\} \quad i=1,\ldots,n-1$$

$$\left. (d_i/h_i)z \leq (d_i/d_{i+1})t_{i+1} - t_i \leq (d_i/g_i)z \right\} \quad i=1,\ldots n-2 \qquad \text{(LP3.5)}$$

$$b,\ w_i \geq 0$$

LP3 has (6n-5) constraints and 3n variables not counting slack and artificial variables.

2.5 Determining the Synchronization. The linear programming variables determine the synchronization of the signals. Let

$\theta(h,i)$ = relative phase (offset) of $S_h$ and $S_i$, measured as the time from the center of a red of $S_h$ to the next center of red of $S_i$. (cycles)

An illustration of $\theta(h,i)$ is given in Figure 2. We adopt the convention $0 \leq \theta(h,i) < 1$. A set of $\theta(1,i)$, $i=2,\ldots,n$ will be called a synchronization of the signals.

In order to give simple, explicit expressions for $\theta$ and certain other quantities, we define, for arbitrary real x:

$$\mathrm{int}[x] = \text{largest integer} \leq x \tag{2.9}$$

$$\mathrm{man}[x] = x - \mathrm{int}[x] \tag{2.10}$$

Thus, for example, int [5.2] = 5, man [5.2] = .2, int [-1.3] = -2, man [-1.3] = .7 .

From Figure 2 and (2.2) we see that

$$\theta(h,i) = \mathrm{man}[\emptyset(h,i)] \tag{2.11}$$

$$\theta(1,i) = \mathrm{man}[w_1 - w_i + \sum_{k=1}^{i-1} t_k + (1/2)(r_1 - r_i)] \tag{2.12}$$

In the symmetric case of LP3, (2.2) shows that $\emptyset(h,i) = \overline{\emptyset}(h,i)$, whence

$$\emptyset(h,i) = (1/2)m(h,i) \tag{2.13}$$

$$\theta(1,i) = \mathrm{man}[(1/2) \sum_{k=1}^{i-1} m_k] . \tag{2.14}$$

2.6 Limits on the Integer Variables. The integer variable $m(h,i)$ is rather limited in the values it can take on without causing an infeasible program. If most of the infeasible values can be excluded in advance, we should be able to save computation time in the algorithm of the next section.

We seek integers $m'(h,i)$ and $m''(h,i)$ such that

$$m'(h,i) \leq m(h,i) \leq m''(h,i) .$$

From (2.3)

$$m(h,i) = (w_h + \overline{w}_h) - (w_i + \overline{w}_i) + t(h,i) + \overline{t}(h,i) + (r_h - r_i) .$$

Since $0 \leq w_k \leq 1 - r_k$ for any k and since a similar relation holds for $\overline{w}_k$ ,

$$t(h,i)+\bar{t}(h,i)-(1-r_h)-(1-r_i) \leq m(h,i) \leq t(h,i)+\bar{t}(h,i)+(1-r_h)+(1-r_i) .$$

Therefore,

$$m''(h,i) = \text{int}[(1-r_h)+(1-r_i)+t(h,i)+\bar{t}(h,i)] \tag{2.15a}$$

$$m'(h,i) = -\text{int}[(1-r_h)+(1-r_i)-t(h,i)-\bar{t}(h,i)] \tag{2.15b}$$

However, t and $\bar{t}$ are usually variables and can be bounded in terms of the constants of the problem. Thus, in LP2,

$$d_i/f_iT_2 \leq t_i \leq d_i/e_iT_1 .$$

Following the notation of LP2, we obtain:

$$m''_i = \text{int}[(1-r_i)+(1-r_{i+1})+(d_i/e_iT_1)+(d_i/\bar{e}_iT_1)] \tag{2.16a}$$

$$m'_i = -\text{int}[(1-r_i)+(1-r_{i+1})-(d_i/f_iT_2)-(d_i/\bar{f}_iT_2)]. \tag{2.16b}$$

## 3. A Branch and Bound Algorithm

The algorithms presently available for solving the general mixed-integer linear programming problem are usually considered only partially satisfactory. Relatively small problems sometimes take an inordinate length of time to solve. Consequently, a specialized algorithm has been developed for the maximal bandwidth problem by using branch and bound methods. These methods have proven rather successful in some other combinatorial problems. [5,6]

3.1 General Description. The basic idea of the method is to break up the set of all feasible solutions into smaller and smaller subsets and to calculate for each an upper (lower) bound on the objective function of the best solution therein. The bounds guide

the partitioning of the subsets and eventually identify a maximizing (minimizing) solution. When a subset is found that contains a single solution whose objective function is greater (less) than or equal to the upper (lower) bounds for all other subsets, that solution is optimal. The subsets of solutions are conveniently represented as the nodes on a tree and the process of partitioning as a branching of the tree; hence the name, "branch and bound".

The convergence of the process can be assured by devising a partitioning procedure that, at worst, leads to eventual enumeration of all solutions. The computational efficiency of the process, however, is very dependent on the methods used to perform the partitions and calculate the bounds. The successful applications to date have exploited special features of the particular class of problems at hand to develop good bounds.

A branch and bound algorithm for LP2 (or LP3)will now be described. Consider an r-signal problem ($2 \leq r \leq n$) consisting of the first r signals of the n-signal problem. Suppose that we pick specific integer values for the variables $m_1, \ldots, m_{r-1}$ in LP2. LP2 is now an ordinary linear program and can be solved in a straightforward manner to maximize the objective function. Now the set of n-signal solutions that have the specific values $m_1, \ldots m_{r-1}$ is a subset of the set of all n-signal solutions. Furthermore, the maximal objective function for the r-signal problem having these m-values forms an upper bound on the objective function for any n-signal solution with these m-values, since adding more signals can only restrict the possibilities for maximization. Thus we have a method for defining subsets of solutions (by specifying m-values)

and for placing bounds on their objective functions (by solving ordinary linear programs). The subsets can be further partitioned by specifying values for more of the m's.

3.2 Flow Diagram. The ideas of the previous section will now be expressed more formally. Let

$X$ = a subset of solutions of the n-signal problem; namely, the set of all solutions that have the *m*-values of a particular *r*-signal problem.

$r(X)$ = the number of signals in the problem defining X.

$m_1(X),\dots,m_{r-1}(X)$ = the integer values of $m_i$ for the *r*-signal problem defining X.

$X_1, X_2 \dots$ = subsets of solutions obtained by partitioning X further.

The set X is partitioned into subsets $X_1, X_2, \dots$ each defined by a different *(r+1)*-signal problem. Any of these (*r+1*)-signal problems, say the one for $X_j$, will have the same $m_i$'s as X up to $i = r-1$:

$$m_i(X_j) = m_i(X) \qquad i=1,\dots r-1 ,$$

and, in addition, a value for $m_r$. Let

$m_r', m_r''$ = smallest and largest value of $m_r$ to be used in partitioning X.

$\alpha(X)$ = an upper bound on the objective function for the solutions in X, as obtained by solving an ordinary linear program for the *r*-signal problem with specified *m*-values.

$\beta$ = best n-signal objective function found so far in the calculation.

A flow diagram of the algorithm is shown in Figure 3. We shall trace through the boxes. As this is done, it may be helpful to refer to Figure 4, which shows the tree of a worked-out 10 signal problem. In the tree representation, subsets of solutions are represented by nodes and so we shall use X to refer to both a subset and its node.

Box 1 initializes three quantities: X to the set of all solutions, r to 1, and $\beta$ to zero. Box 2 initializes two quantities to be used in the construction of the nodes branching out of X: The index j for counting the nodes is set to 1 and $m_r$, whose values will define the nodes, is set to its lower limit $m_r'$.

Box 3 sets up a node $X_j$ on a branch out of X. The node may be identified by $m_r(X_j)$, which is set to the current value of $m_r$. (In Figure 4 the value of $m_r(X_j)$ is written inside the circle of each node, $X_j$. To define the node completely, we also need $m_1(X_j),\ldots,m_{r-1}(X_j)$, but these can be found by tracing back through the tree to "all solutions" and reading off the $m_i$ in the nodes encountered. Next LP2 is solved for the (r+1)-signal problem defined by the current m-values, $m_1,\ldots,m_r$. The maximal value of the objective function becomes the upper bound $\alpha(X_j)$ and is used to label the node.

Box 4 tests whether an n-signal problem is currently being solved. If so, box 5 tests to see whether the resulting objective function is better than the previous best. If this is the case, box 6 replaces the previous best with the new solution.

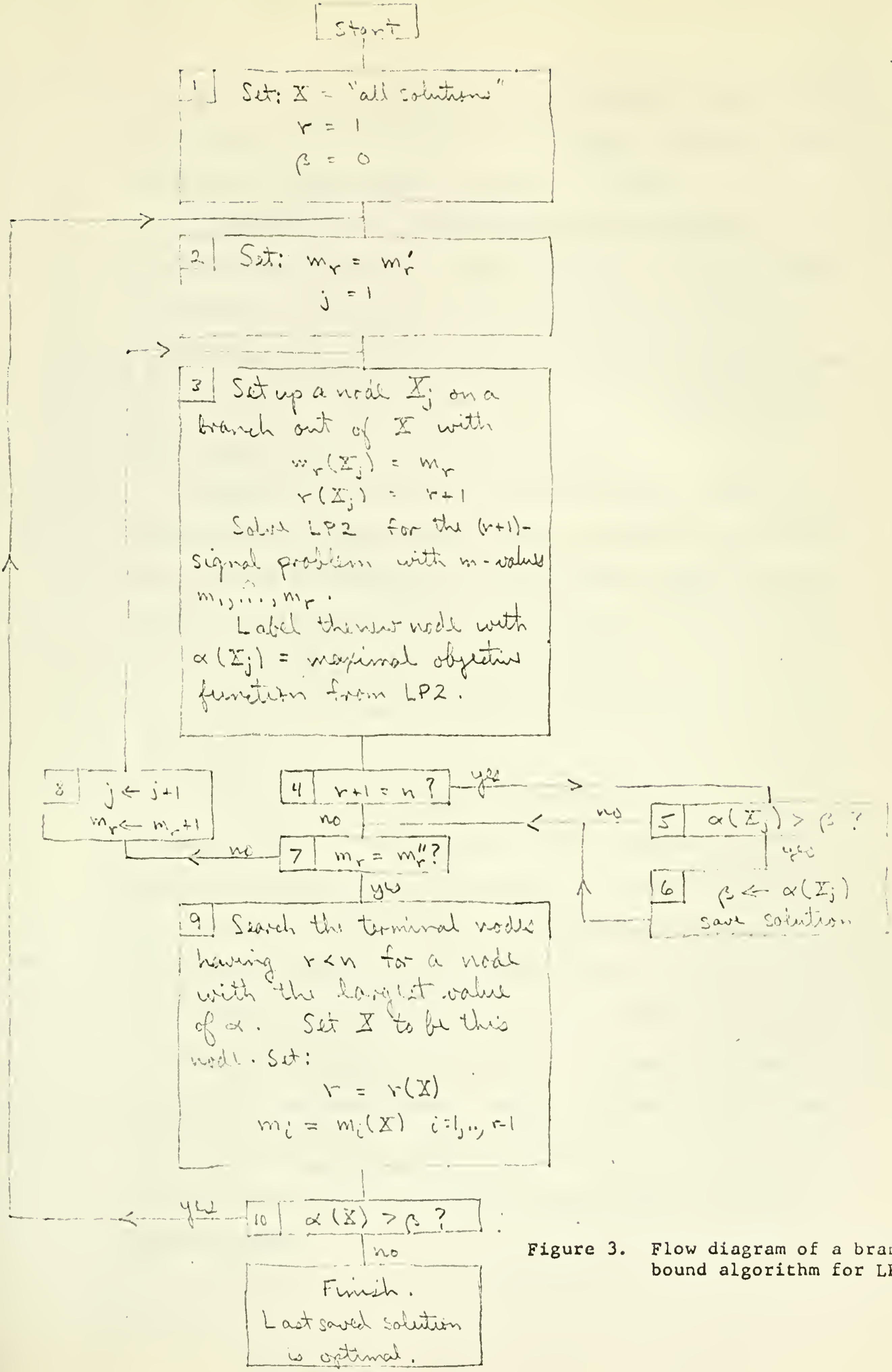
Start
1 Set: $X$ = "all solutions"
$r = 1$
$\beta = 0$
2 Set: $m_r = m_r'$
$j = 1$
3 Set up a node $X_j$ on a branch out of $X$ with
$m_r(X_j) = m_r$
$r(X_j) = r+1$
Solve LP2 for the $(r+1)$-signal problem with m-values $m_1, \ldots, m_r$.
Label the new node with $\alpha(X_j)$ = maximal objective function from LP2.
4 $r+1 = n$ ?
yes
no
5 $\alpha(X_j) > \beta$ ?
yes
no
6 $\beta \leftarrow \alpha(X_j)$
save solution
7 $m_r = m_r''$?
no
yes
8 $j \leftarrow j+1$
$m_r \leftarrow m_r+1$
9 Search the terminal nodes having $r < n$ for a node with the largest value of $\alpha$. Set $X$ to be this node. Set:
$r = r(X)$
$m_i = m_i(X)$ $i=1,..,r-1$
10 $\alpha(X) > \beta$ ?
yes
no
Finish.
Last saved solution is optimal.

Figure 3. Flow diagram of a branch and bound algorithm for LP2.

Box 7 tests to see whether all the planned $m_r$ values have been used to create new nodes. If not, box 8 increases j and $m_r$ by one and returns to box 3 to set up a new node.

Box 9 searches for a new node from which to branch. The candidates are the terminal nodes of the tree, i.e., those without branches. There is no need to consider nodes with $r = n$. Of the eligible nodes, the one with the largest $\alpha$ is selected and becomes the new X. The current value of r is reset to r(X) and the current values of $m_i$ to $m_i(X)$ for $i=1,\ldots,r-1$.

Box 10 tests to see whether the largest upper bound, $\alpha$, is greater than the best n-signal objective function so far. If so, there is some possibility of finding a larger n-signal objective function and the calculation returns to box 2 to branch from the new X. Otherwise, the calculation is finished, and the current best solution is optimal for the original mixed-integer n-signal problem.

It may be helpful to trace through the first few nodes in Figure 4. First the node "all solutions" is laid out. Then, since $m_1' = 0$ and $m_1'' = 1$, the nodes $m_1 = 0$ with $\alpha = .440$ and $m_1 = 1$ with $\alpha = .292$ are laid out. The largest $\alpha$ of these two is .440 and so the $m_1 = 0$ node is for branching. Out of it comes the nodes: $m_2 = 0$, $\alpha = .282$ and $m_2 = 1$, $\alpha = .381$. The next step would be to branch from the $m_2 = 1$, $\alpha = .381$ node. Eventually the tree is developed as shown. Notice that the 10-signal node labeled "optimum" has an objective function of .282, which is greater than or equal to the upper bound for any of the other terminal nodes.

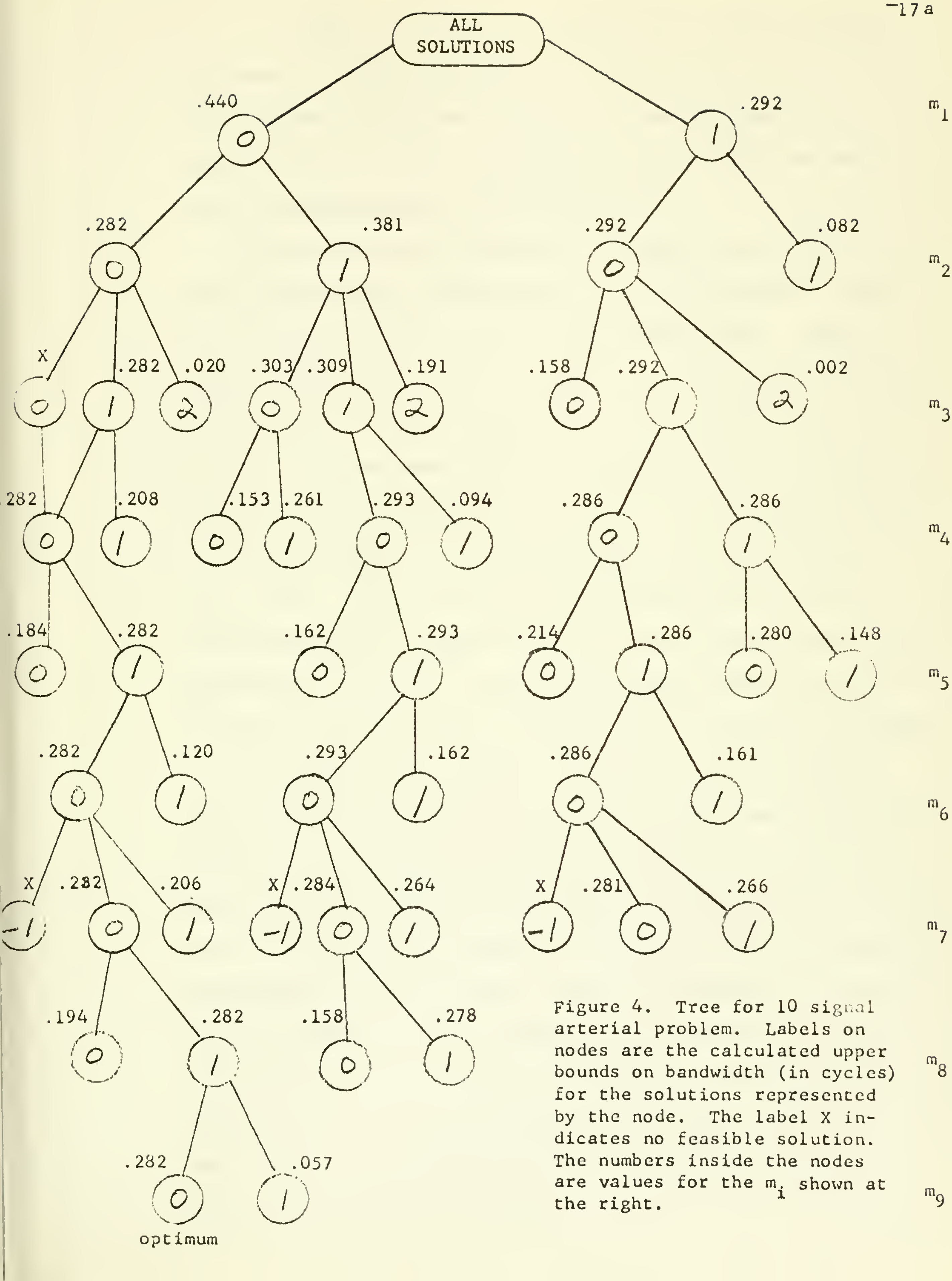

Figure 4. Tree for 10 signal arterial problem. Labels on nodes are the calculated upper bounds on bandwidth (in cycles) for the solutions represented by the node. The label X indicates no feasible solution. The numbers inside the nodes are values for the $m_i$ shown at the right.

3.3 Example. The 10 signal problem being used as a numerical example is taken from Reference 3 and represents a stretch of Euclid Avenue in Cleveland. The distances and green splits are taken from actual usage. The limits on period, speeds, and speed changes have been made up fairly arbitrarily.

The constants of the problem are briefly as follows. The signals, starting with $S_1$ and working outbound, are located at 0, 168, 381, 716, 929, 1173, 1371, 1493, 1706, and 1843 meters. The corresponding red times are .47, .40, .40, .47, .48, .42, .40, .40, .40, and .42 cycles. Limits on period are $T_1 = 55$ seconds and $T_2 = 75$ seconds. Lower and upper limits on speed are $e_i = \bar{e}_i = 13.4$ meters/second (30 mph, 48.3 km/hr) and $f_i = \bar{f}_i = 17.9$ meters/second (40 mph, 64.4 km/hr) for each i. Limits on change in reciprocal speed are $1/g_i = 1/\bar{g}_i = - 1/h_i = - 1/\bar{h}_i = .0121$ (meters/second)$^{-1}$ for each i. This corresponds to a maximum possible change in speed of about ±2.2 meters/second (4.9 mph, 7.9 km/hr) at the lower limit of speed and ±3.9 meters/second (8.7 mph, 14.0 km/hr) at the upper limit of speed.

The problem can be solved by LP2 or, since we have chosen to solve for equal bandwidths, by LP3. The tree developed by the branch and bound algorithm appears in Figure 4. A total of 52 linear programs are required. The maximal bandwidth for each direction is .282 cycles. The space-time diagram for the street is shown in Figure 5. The optimal period is 75 seconds, making the bandwidth 21.2 seconds. Starting between $S_1$ and $S_2$ and working up, the speeds are: 17.9, 17.9, 17.1, 14.2, 13.4, 14.9, 13.4, 15.6, 17.9 meters/second. The phases, $\theta(1,i)$, in order of increasing index starting with $i=1$, are: 0, 0, 0, 1/2, 1/2, 0, 0, 0, 1/2, 1/2 cycles. The critical signals are $S_1$, $S_3$, $S_5$, $S_6$, and $S_9$ .

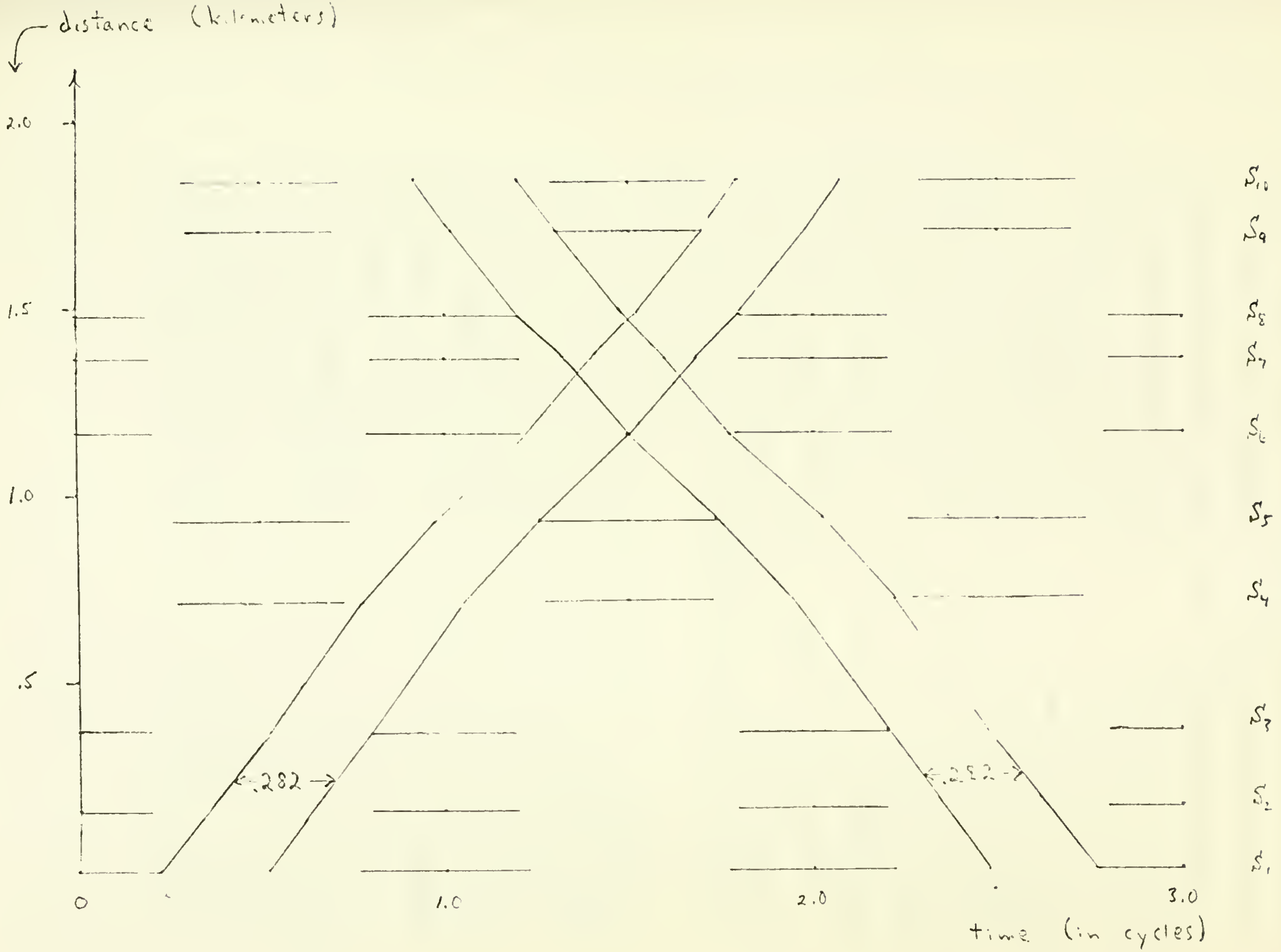

Figure 5. Space-time diagram for symmetric 10 signal problem. Maximal bandwidth is .282 cycles or 21.2 seconds. The signal period is 75 seconds.

The results can be compared to those obtained by exploring various constant speeds and periods with the methods of Reference 3. The best result found was 15.2 meters/second at 65 seconds for a bandwidth of .235 cycles. This is essentially the synchronization displayed in Reference 3. The additional flexibility of variable speed has permitted at 20% increase in bandwidth.

3.4 Discussion. For the numerical example reported here, the steps of the branch and bound algorithmwere carried out manually except for the linear programs. In some linear programming codes, it is possible to suppress constraints. Then the entire *n*-signal problem can be loaded and only those constraints corresponding to the current *r*-signal problem used. Changes in the $m_i$ are effected by changes in the constants vector. Frequently, time can be saved by starting a new problem from the basis of a similar problem just completed.

The computational limits of the algorithm are relatively unexplored. If experience from a rather different application [5] is a guide, the number of linear programs required is likely to increase exponentially with n. If so, there may well be a fairly sharp upper limit to the size of the problem that is computationally feasible. We feel, however, that the example here demonstrates that problems of a size of practical interest can be solved.

A solution that is likely to be good but may not be optimal can be obtained by solving roughly 2n linear programs. The procedure is simply to branch next from the node with the largest $\alpha$

among those just created from the current X. Frequently there are only two such nodes. The scanning of the other terminal nodes is omitted. The value of r increases monotonically to n. In the example of Figure 4, the solution obtained in this manner has a bandwidth of ..278.

## 4. Network Problems

The mixed-integer formulation can be extended to networks. The network program consists of arterial programs for the individual streets plus "cycle constraints" that connect the streets together. An objective function can be formed from the bandwidths of the arteries. A new decision variable, the red-green split, can usefully be introduced at certain signals.

Some of the decision variables of LP2 will be left out of the network formulation to be developed in this section. The reason is that we can expect network problems to be larger and computationally more difficult than individual arterial problems. Therefore it seems desirable to show ways in which the program might be simplified. We shall keep speed fixed over the length of an artery, but variable both for the artery as a whole and from artery to artery.

4.1 Arterial Constraints. In a network problem the term artery will be used to refer to any street in which a progression is desired, i.e., on which we define a bandwidth and bring it into the mathematical program. As earlier, the signals of the network are presumed to be designated $S_1, S_2, \ldots, S_n$ but we can no longer suppose that signals with adjacent index values are adjacent. Consider then an artery and suppose that the signals

at its extreme ends are $S_i$ and $S_j$. We designate one direction, say $S_i$ to $S_j$, and outbound and thereby define artery ij.

Quantities relating to artery ij will be identified by a superscript ij. Thus, in a straightforward way, the appropriate definitions of Section 2 generalize to $b^{ij}$, $r_k^{ij}$, and $w_k^{ij}$. In addition, let

$v^{ij}$ = speed in outbound direction on artery ij. (meters/second)

$e^{ij}$, $f^{ij}$ = lower and upper limits on outbound speed on artery ij. (meters/second)

$u^{ij}$ = $1/v^{ij}T$ = reciprocal speed outbound on artery ij. $(\text{meters/cycle})^{-1}$

Analogous notation is used as appropriate for inbound variables. When two signals are required to define a quantity, the arterial designation is redundant and will be omitted. For example, the quantities $\phi(h,i)$, $\bar{\phi}(h,i)$, m(h,i) and d(h,i) will be used as defined in Section 2.

The constraints for artery ij are as follows. For the artery as a whole:

$$(1/f^{ij})z \leq u^{ij} \leq (1/e^{ij})z \quad , \qquad \text{(LP4.1a)}$$

$$(1/\bar{f}^{ij})z \leq \bar{u}^{ij} \leq (1/\bar{e}^{ij})z \quad ; \qquad \text{(LP4.1b)}$$

for each signal $S_k$ on the artery:

$$w_k^{ij} + b^{ij} \leq 1 - r_k^{ij} \quad , \qquad \text{(LP4.2a)}$$

$$\bar{w}_k^{ij} + \bar{b}^{ij} \leq 1 - r_k^{ij} \quad ; \qquad \text{(LP4.2b)}$$

for each pair of adjacent siganls on the artery, say, $S_k$ followed by $S_\ell$ outbound:

$$(w_k^{ij} + \bar{w}_k^{ij}) - (w_\ell^{ij} + \bar{w}_\ell^{ij}) + d(k,\ell)\ (u^{ij} + \bar{u}^{ij}) = m(k,\ell) - (r_k - r_\ell) \qquad \text{(LP4.3)}$$

$$m(k,\ell) = \text{integer} \qquad \text{(LP4.4)}$$

4.2 Objective Function. In a progression the bandwidth determines the maximum length (in time) of a platoon that can pass unimpeded down the street when the prescribed speeds are maintained. Therefore, when a progression system is appropriate, the streets (or directions on streets) should presumably be allocated bandwidth in some manner according to the flows on them. Two devices are available for allocating bandwidth among streets (and directions): the objective function and the constraints.

Ideally we might constrain each bandwidth to be greater than some specified size needed to pass a known flow. Unfortunately, there is no guarantee that the result will be feasible. As an alternative, we can choose some important bandwidth, call it $b^o$, and maximize this, but require that each other bandwidth be greater than or equal to a specified fraction of $b^o$. The fractions would be chosen according to relative flows.

Such constraints are not sufficient to get the most out of the system. Sometimes the bandwidth on an artery will not be in conflict with bandwidth on other arteries. In order to guarantee that such a bandwidth is maximized, every bandwidth must be included in the objective function. The less important bandwidths can be weighted by small coefficients and $b^o$ by a large coefficient.

To formalize these ideas, let $b^o$, $b^1$,...,$b^q$ be the bandwidths of the problem (reindexed here for convenience) and let

$a^i$ = weight assigned to the $i^{th}$ bandwidth; $i=0,\dots,q$.

$k^i$ = fraction of $b^o$ to be guaranteed to the $i^{th}$ bandwidth; $i=1,\dots, q$.

We take as the objective of the network program

$$\max a^o b^o + \dots + a^q b^q, \qquad \text{(LP4.5)}$$

as constrained by

$$b^i \geq k^i b^o, \qquad i=1,\dots,q \qquad \text{(LP4.6)}$$

Some care must be exercised in the choice of the $k^i$ so as not to let an unimportant artery limit $b^o$.

4.3 Cycle Constraints. A new constraint with a new integer variable must be introduced whenever several arteries intersect to form a closed loop or, as we shall call it, a cycle. An as example of a cycle, $S_1$, $S_3$, $S_4$ and $S_7$ appear at the arterial intersections of a cycle in Figure 6. The basic reason for the new constraints lies in a physical requirement of simultaneity. Suppose we set a master clock to zero at a center of red of $S_1$. If we proceed down artery 13 to $S_3$, the time of the center of red there is fixed by the progression on 13. As a result, the time of the center of red for $S_3$ along the direction of artery 35 is also fixed on the master clock. The progression on 35 then fixes the time of center of red at $S_5$. Continuing around the cycle, we eventually return to determine the time of a center of

red at $S_1$. This time must be an integral number of periods after time zero. The algebraic statement of this requirement constitues the new constraint.

The terminology of graph theory is helpful in describing networks. Signals appear at <u>nodes</u>. Lengths of street between adjacent signals are called <u>arcs</u>. An arc has a direction, which will be the direction chosen as outbound on the street segment. A sequence of arcs, irrespective of arc directions, that closes on itself will be a <u>cycle</u>. The node for $S_k$ will be denoted simply by k. The arc from i to j will be denoted (i,j). An arc's direction is indicated in diagrams by an arrow.

The formulation of the constraint for a single cycle proceeds as follows: Suppose we trace around a cycle and find pieces of arteries $i_1j_1$, $i_2j_2$,...$i_pi_p$ in that order. Suppose that the signals at which arteries intersect are represented by the nodes, $k_1$, $k_2$,...,$k_p$ with $k_1$ at the intersection of arteries $i_pj_p$ and $i_1j_1$, $k_2$ at the intersection of $i_1j_1$ and $i_2j_2$, etc. The cycle will then be denoted $C(k_1,\dots,k_p)$.

Let t = 0 be the time (measured in cycles) of a center of red at $k_1$ along artery $i_1j_1$. Then $t = \emptyset(k_1, k_2)$ is the time of a center of red at $k_2$ along artery $i_1j_1$. The same instant is also the center of green at $k_2$ along artery $i_2j_2$. Therefore a center of red at $k_2$ along $i_2j_2$ occurs at

$$t \;=\; \emptyset(k_1,\, k_2) + 1/2.^{*}$$

* The use of 1/2 assumes that red time on one street of the intersection coincides with green time on the cross street and vice versa. More complicated arrangements are possible and might require replacing the 1/2 by some other fraction.

Starting from this time we can add $\emptyset(k_2,k_3)$ to find a center of red at $k_3$ along artery $i_2j_2$ . Adding 1/2 we turn the corner and get on artery $i_3j_3$ . Proceeding around the cycle, we eventually determine the time at which a center of red occurs at $k_1$ along artery $i_1j_1$ . Let

$$c(k_1,\ldots,k_p) = \text{integer variable for the cycle } C(k_1,\ldots,k_p).$$

Then the cycle constraint is:

$$c(k_1,\ldots,k_p) = \emptyset(k_1,k_2)+\emptyset(k_2,k_3)+\ldots+\emptyset(k_{p-1},k_p)+\emptyset(k_p,k_1)+(p/2). \quad \text{(LP4.7)}$$

Appropriate expressions for $\emptyset(i,j)$ in terms of other variables of the program are given by (2.2) or (2.13), possibly with some help from (2.4) and (2.5).

Multiple cycles introduce multiple constraints. The number of possible cycles is greater than the number of required constraints. The minimum number of constraints can be developed by tracing over the network as follows: Pick a starting node. Trace over the arcs (irrespective of their direction) that make up each artery passing through the starting node. Then, for each node reached by the tracing so far, trace over the arcs on the arteries that pass through the node. Now, whenever the tracing procedure closes a cycle, form a constraint from the cycle. Continue the process until no further arcs can be traced. If the whole network has been traced, the job is finished. Otherwise, the network breaks down into two (or more) disconnected sub-networks, one of which has just been isolated. Pick another starting node and continue.

The tracing process works because it identifies which signals bear rigid time relationships to the starting signal by virtue of

being connected to it through a sequence of arterial progressions. Whenever a cycle is closed the rigid relationships must be made compatible by a constraint..

If the network has a signal at every intersection of arteries (i.e. there are no overpasses), the graph of the system is planar. Then the number of cycle constraints equals the number of distinct areas enclosed by arteries. Figure 6 shows a planar graph requiring two cycle constraints. Cycle constraints are not necessarily unique. If Figure 6, for example, we could use the pair from C(1,3,4,7) and C(7,4,5,6) or equally well the pair from C(1,3,4,7) and C(1,3,5,6). Algebraically, either of these pairs can be obtained from the other.

4.4 Red Time Constraints. A signal at the intersection of two arteries may be a critical signal for one but not the other. Then a shift of green time from a street to the cross street may increase bandwidth on one artery without reducing it on the other. Thus, the red time at arterial intersections may sometimes be a useful decision variable. The variable will presumably be constrained. The fraction of time red on each street may be limited because of flow requirements. The absolute time red may be limited on the low side to permit pedestrian crossing, and on the high side to placate drivers.

For signals at which red time is a decision variable, let

$E_k^{ij}$, $F_k^{ij}$ = lower and upper limits on $r_k^{ij}$. (cycles)

$G_k^{ij}$, $H_k^{ij}$ = lower and upper limits on $r_k^{ij}T$. (seconds)

Then the variable $r_k^{ij}$ is constrained by

$$E_k^{ij} \leq r_k^{ij} \leq F_k^{ij} \qquad \text{(LP4.8a)}$$

$$G_k^{ij}z \leq r_k^{ij} \leq H_k^{ij}z \qquad \text{(LP4.8b)}$$

4.5 Steps in Formulating a Network Problem. The general mixed-integer linear program for the network problem is notationally somewhat cumbersome and rather than write the program out we list the steps that generate it.

1. Determine which streets of the network are to be arteries.

2. Set up the objective function and related constraints, (LP4.5) and (LP4.6)

3. Set up the period constraints:

$$(1/T_2) \leq z \leq (1/T_1) \qquad \text{(LP4.9)}$$

4. For each artery, set up the constraints, (LP4.1) - (LP4.4), or, if preferred, the constraints of LP2 or LP3.

5. Set up cycle constraints using (LP4.7) and the procedure of Section 4.3.

6. Add red time constraints (LP4.8) for any red times being made decision variables.

Network problems can be solved by the branch and bound methods of Section 3 with a few modifications. As there, the integer

variables are introduced one at a time. At any stage, specific values are given to some of the integer variables and serve to define a subset of solutions to the network problem. An upper bound on the objective function for solutions in the subset is found by solving an ordinary linear program using the specific values of the integer variables.

However, in the network problem, the full _n_-signal objective function must be used even for an _r_-signal problem. Otherwise the upper bound obtained is inappropriate. As a result, however, every bandwidth that appears in the objective function must have a constraining equation (LP4.2) in each linear program solved, for otherwise the objective function will have an unbounded maximum. Therefore, the "_r_-signal" problem will usually include constraints from more than r signals. It would in fact be permissible to use all the constraints of the original problem (except those involving integer variables not yet assigned specific values) in every ordinary linear program. With the exception of the unboundedness problem just noted, however, the constraints associated with a signal would not be expected to affect the objective function very much until the constraint (LP4.3) containing the integer variable is added. Thus, adding the constraints signal by signal holds down the size of the linear program and may be expected to save computation time.

The order of adding signals, however, will affect the size of the tree. By and large, troublesome constraints should be introduced early. Examples would be constraints for signals close together and constraints for cycles that have small perimeters.

4.6 Example. Figure 6 displays a 7 signal, 5 artery problem which has been solved in the symmetric case. The arteries are 13, 35, 47, 56, and 16. The objective function used is

$$b^{13} + .01(b^{35} + b^{16} + b^{56} + b^{47}),$$

along with the constraints

$$b^{35} \geq .5b^{13}, \quad b^{16} \geq .5b^{13}, \quad b^{56} \geq .5b^{13}, \quad b^{47} \geq .5b^{13}$$

Limits on period are 50 and 100 seconds. Speed on each artery is limited by 14 and 16 meters/second (31.3 and 35.8 mph). All red times are taken to be .5 cycles except $r_7^{13}$, which is made a decision variable subject to: $.4 \leq r_7^{13} \leq .6$ cycles and $25 \leq r_7^{13}T \leq 50$ seconds. Distances between signals are shown in Figure 6.

The complete program is given in the appendix along with the solution tree. There are 44 constraints and 10 integer variables. However, in the branch and bound solution we deal with 42 constraints and 8 integer variables. The reason is that each of the two cycle constraints is introduced by limiting the values that some regular integer variable is permitted to range over.

The network example is not hard to solve. 15 linear programs were required. In the optimal solution the bandwidths are $b^{13} = .35$, $b^{35} = .286$, $b^{47} = .5$, $b^{56} = .5$, and $b^{16} = .286$ cycles. The period is 62.5 seconds. The speeds are $v^{13} = 16$, $v^{35} = 14$, $v^{47} = 16$, $v^{56} = 16$, and $v^{16} = 14$ meters/second. The value of $r_7^{16} = .5$ cycles.

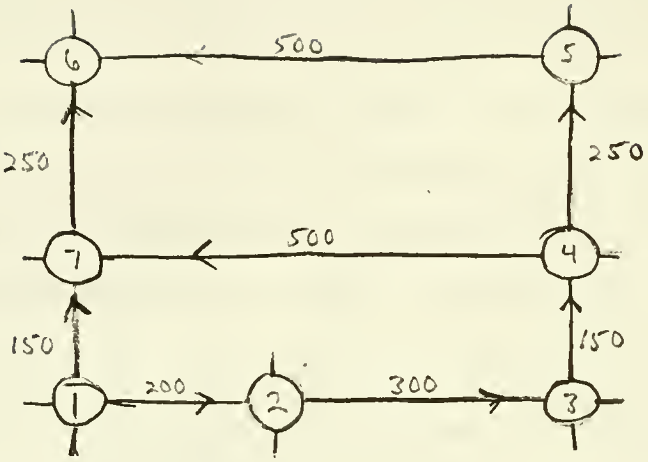

Figure 6. A network of 7 signals. In this network, five arteries: 13, 35, 56, 16, and 47, have been selected for consideration. Two cycle constraints are required. Arrows indicate direction chosen as outbound. Numbers on street segments are distances in meters. The red-green split has been made a decision variable at 7.

## 5. Summary

A variety of maximal bandwidth problems have been formulated as mixed-integer programs. These include single arterial problems, more complicated arterial problems using speed and period as decision variables, and problems involving street networks. Some of these cases, including that of the street network, do not appear to have been put in any kind of optimizing format before.

Branch and bound methods are given for solving the programs generated and examples are worked out. Although it is not known at this point how large a program can reasonably be solved by these methods, it seems likely that problems large enough to be of practical interest can be solved.

* * * * *

## Acknowledgement

The author wishes to thank William H. Allen, III for his constructive suggestions and for solving the 10 signal arterial example in the form LP2.

## Appendix 1

The mixed-integer linear program for the network of Figure 6 is given below for the symmetric case.

Find $b^{ij}$, $z$, $u^{ij}$, $w_k^{ij}$, $m(k,\ell)$, $c\ (i,j,k,\ell)$, and $r_k^{ij}$ to

$$\max b^{13} + .01(b^{35} + b^{16} + b^{56} + b^{47})$$

Subject to:

bandwidth constraints:

$$b^{35} \geq .5b^{13} \qquad (A.1)$$

$$b^{16} \geq .5b^{13} \qquad (A.2)$$

$$b^{56} \geq .5b^{13} \qquad (A.3)$$

$$b^{47} \geq .5b^{13} \qquad (A.4)$$

period constraints:

$$.01 \leq z \leq .02 \qquad (A.5,6)$$

artery 13:

$$.0625z \leq u^{13} \leq .0714z \qquad (A.7,8)$$

$$w_1^{13} + b^{13} \leq .5 \qquad (A.9)$$

$$w_2^{13} + b^{13} \leq .6 \qquad (A.10)$$

$$w_1^{13} - w_2^{13} + 200u^{13} = .5m(1,2) - .05 \qquad (A.11)$$

$$w_3^{13} + b^{13} \leq .5 \qquad (A.12)$$

$$w_2^{13} - w_3^{13} + 300u^{13} = .5m(2,3) +.\ 05 \qquad (A.13)$$

artery 35: $.0625z \leq u^{35} \leq .0714z$ (A.14,15)

$w_3^{35} + b^{35} \leq .5$ (A.16)

$w_4^{35} + b^{35} \leq .5$ (A.17)

$w_3^{35} - w_4^{35} + 150u^{35} = .5m(3,4)$ (A.18)

$w_5^{35} + b^{35} \leq .5$ (A.19)

$w_4^{35} - w_5^{35} + 250u^{35} = .5m(4,5)$ (A.20)

artery 56: $.0625z \leq u^{56} \leq .0714z$ (A.21,22)

$w_5^{56} + b^{56} \leq .5$ (A.23)

$w_6^{56} + b^{56} \leq .5$ (A.24)

$w_5^{56} - w_6^{56} + 500u^{56} = .5m(5,6)$ (A.25)

artery 47: $.0625z \leq u^{47} \leq .0714z$ (A.26,27)

$w_4^{47} + b^{47} \leq .5$ (A.28)

$w_7^{47} + b^{47} \leq 1 - r_7^{47}$ (A.29)

Red time constraints $r_7^{47} + r_7^{16} = 1$ (A.30)

$.4 \leq r_7^{16} \leq .6$ (A.31,32)

$25z \leq r_7^{16} \leq 50z$ (A.33,34)

$w_4^{47} - w_7^{47} + 500u^{47} = .5m(4,7) - .5(.5 - r_7^{47})$ (A.35)

artery 16:

$$.0625z \le u^{16} \le .0714z \tag{A.36,37}$$

$$w_1^{16} + b^{16} \le .5 \tag{A.38}$$

$$w_7^{16} + b^{16} \le 1 - r_7^{16} \tag{A.39}$$

$$w_1^{16} - w_7^{16} + 150u^{16} = .5m(1,7) - .5(.5 - r_7^{16}) \tag{A.40}$$

$$w_6^{16} + b^{16} \le .5 \tag{A.41}$$

$$w_7^{16} - w_6^{16} + 250u^{16} = .5m(7,6) - .5(r_7^{16} - .5) \tag{A.42}$$

cycle C(1,3,4,7):

$$m(1,2) + m(2,3) + m(3,4) + m(4,7) - m(1,7) = 2c(1,3,4,7) \tag{A.43}$$

cycle C(4,5,6,7):

$$m(4,5) + m(5,6) - m(7,6) - m(4,7) = 2c(4,5,6,7) \tag{A.44}$$

$m(i,j)$, $c(i,j,k,\ell)$ integers

$w_k^{ij}, b^{ij} \ge 0$

In using the branch and bound algorithm, the constraints were introduced in the following blocks, each terminating with an integer variable constraint: (1) A.1-A.10, A.16, A.23, A.28, A.38, A.11 . (2) A.12, A.13 . (3) A.14, A.15, A.17, A.18 . (4) A.26, A.27, A.29-A.35 . (5) A.36, A.37, A.39, A.40, A.43 . (6) A.19, A.20 . (7) A. 41, A.42 . (8) A.21, A.22, A.24, A.25, A.44 .

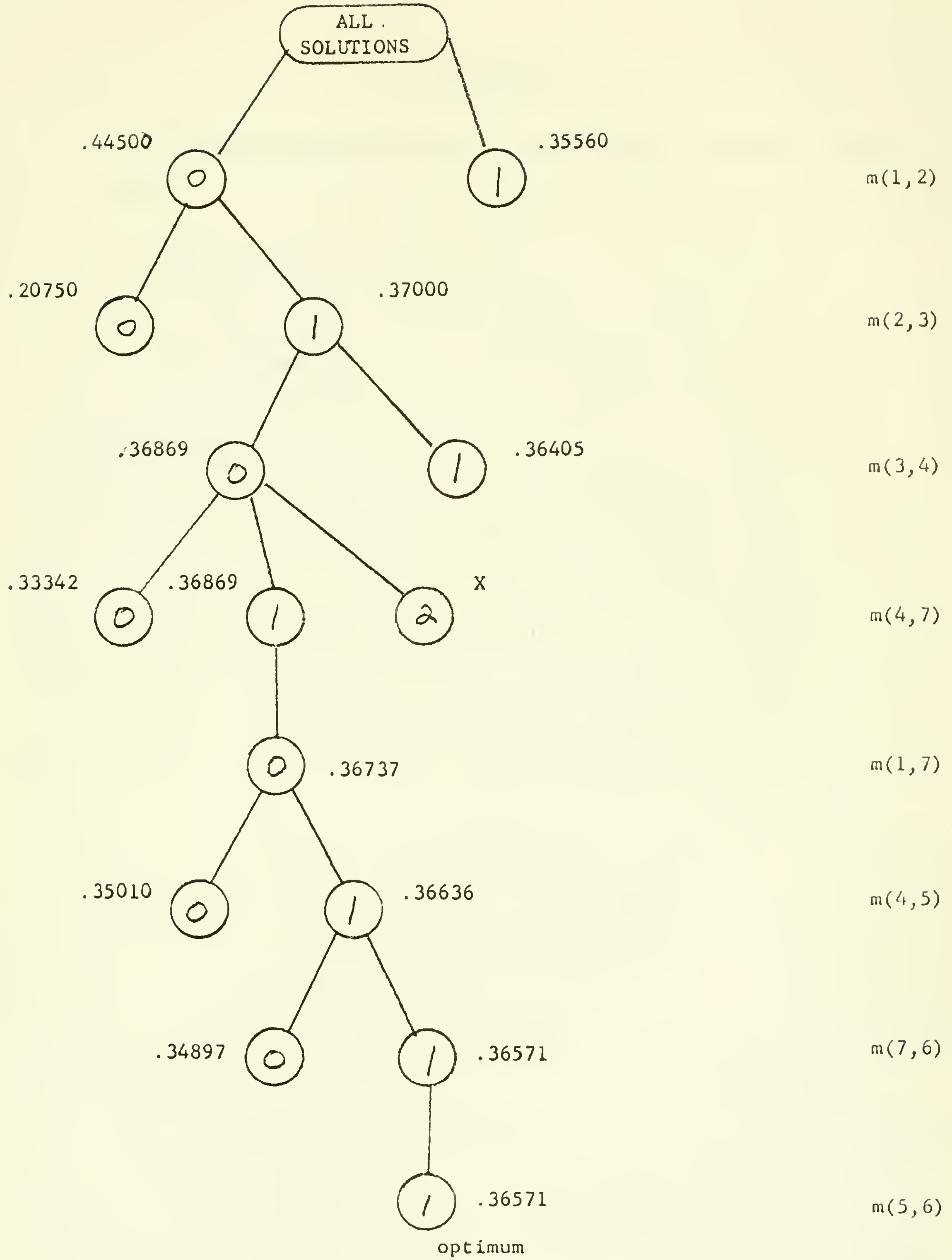

Figure 7. Tree for network example. Labels on the nodes are the calculated upper bounds on the objective function. The objective function is a weighted sum of arterial bandwidths. Numbers inside the nodes are values for the $m_i$ shown at right. Cycle constraints have been used to limit the possible choices of m(1,7) and m(5,6).

## Appendix 2

The mixed-integer program for the 10 signal problem is given below.

Find b, z, $w_i$, $t_i$, $m_i$ to

$$\max b$$

Subject to:

$$.0182 \leq z \leq .0133 \qquad \text{(A2.1,2)}$$

$$w_1 + b \leq .53 \qquad \text{(A2.3)}$$

$$9.4z \leq t_1 \leq 12.5z \qquad \text{(A2.4,5)}$$

$$w_2 + b \leq .60 \qquad \text{(A2.6)}$$

$$w_1 - w_2 + t_1 = (1/2)m_1 - .035 \qquad \text{(A2.7)}$$

$$11.9z \leq t_2 \leq 15.9z \qquad \text{(A2.8,9)}$$

$$-2.03z \leq .787t_2 - t_1 \leq 2.03z \qquad \text{(A2.10,11)}$$

$$w_3 + b \leq .60 \qquad \text{(A2.12)}$$

$$w_2 - w_3 + t_2 = (1/2)m_2 \qquad \text{(A2.13)}$$

$$18.7z \leq t_3 \leq 25.0z \qquad \text{(A2.14,15)}$$

$$-2.58z \leq .636t_3 - t_2 \leq 2.58z \qquad \text{(A2.16,17)}$$

$$w_4 + b \leq .53 \qquad \text{(A2.18)}$$

$$w_3 - w_4 + t_3 = (1/2)m_3 + .035 \qquad \text{(A2.19)}$$

$$11.9z \le t_4 \le 15.9z \tag{A2.20,21}$$

$$-4.05z \le 1.571t_4 - t_3 \le 4.05z \tag{A2.22,23}$$

$$w_5 + b \le .52 \tag{A2.24}$$

$$w_4 - w_5 + t_4 = (1/2)m_4 + .005 \tag{A2.25}$$

$$13.6z \le t_5 \le 18.2z \tag{A2.26,27}$$

$$-2.58z \le .873t_5 - t_4 \le 2.58z \tag{A2.28,29}$$

$$w_6 + b \le .58 \tag{A2.30}$$

$$w_5 - w_6 + t_5 = (1/2)m_5 - .03 \tag{A2.31}$$

$$11.1z \le t_6 \le 14.8z \tag{A2.32,33}$$

$$-2.95z \le 1.231t_6 - t_5 \le 2.95z \tag{A2.34,35}$$

$$w_7 + b \le .60 \tag{A2.36}$$

$$w_6 - w_7 + t_6 = (1/2)m_6 - .01 \tag{A2.37}$$

$$6.8z \le t_7 \le 9.1z \tag{A2.38,39}$$

$$-2.40z \le 1.624t_7 - t_6 \le 2.40z \tag{A2.40,41}$$

$$w_8 + b \le .60 \tag{A2.42}$$

$$w_7 - w_8 + t_7 = (1/2)m_7 \tag{A2.43}$$

$$11.9z \leq t_8 \leq 15.9z \qquad \text{(A2.44,45)}$$

$$-1.48z \leq .573t_8 - t_7 \leq 1.48z \qquad \text{(A2.46,47)}$$

$$w_9 + b \leq .60 \qquad \text{(A2.48)}$$

$$w_8 - w_9 + t_8 = (1/2)m_8 \qquad \text{(A2.49)}$$

$$7.7z \leq t_9 \leq 10.2z \qquad \text{(A2.50,51)}$$

$$-2.58z \leq 1.555t_9 - t_8 \leq 2.58z \qquad \text{(A2.52,53)}$$

$$w_{10} + b \leq .58 \qquad \text{(A2.54)}$$

$$w_9 - w_{10} + t_9 = (1/2)m_9 + .01 \qquad \text{(A2.55)}$$

$$w_i, b \geq 0$$

The above data is arranged in blocks. The first (r-1) blocks represent an r-signal problem for signals $S_1, \ldots, S_r$.

## References

1. W. Helly and P. G. Baker, "Accelerated Noise in a Congested Signalized Environment," Report R & D 65-2, Engineering Department of the Port of New York Authority, June 1965.

2. W. von Stein, "Traffic Flow with Pre-Signals and the Signal Funnel," in R. Herman, (ed.), *Theory of Traffic Flow*, Elsevier, Amsterdam, 1961.

3. J. T. Morgan and J. D. C. Little, "Synchronizing Traffic Signals for Maximal Bandwidth," *Operations Research*, **12**, 896-912, (Nov.-Dec. 1964).

4. J. Johnson, "Optimum Control of an Unsaturated Artery," Data Processing Division, IBM, August 1964.

5. J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem," *Operations Research*, **11**, 972-989, (Nov.-Dec. 1963).

6. Z. A. Lomnicki, "A Branch-and-Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem, *Operational Research Quarterly*, **16**, 89-100, (March 1965).