Transpn Res. Vol. 2, pp. 199–234. Pergamon Press 1968. Printed in Great Britain

SCHEDULING OF TRAFFIC LIGHTS—A NEW APPROACH†

KARL E. STOFFERS

School of Engineering, Sacramento State College, Sacramento, California, U.S.A.

(*Received* 19 *March* 1968)

INTRODUCTION

THE APPROACH to traffic light scheduling in this paper has originated from a study of existing controllers and scheduling techniques. In the introduction a summary of this background will be given together with some discussion of the functional relation between traffic light schedules and intersection design.

(a) *Controllers*

Traffic lights at intersections are controlled by rotating switching devices. Usually the controllers for one intersection consist of one or two stages. Each stage contains a rotating stepper switch or an equivalent logical network.

(a.1) *Single-stage controllers.* In single-stage controllers (Fig. 2a) the lights are operated individually by the rotating time switch. The switch progresses at equally spaced instants of time (usually one step per sec) and initiates the changes of light display by contacting the connections wired to it. The timing of each light is set by wiring it to the appropriate contacts of the switch. The instants of time in which the display of one (or more) lights changes are called "starting points".

Since the lights can be timed independent of each other, a supervisory control logic should be provided, which interferes if "conflicting greens" would occur. The compatibility matrix (e.g. Fig. 1a) is an example of the specifications for such a logic.

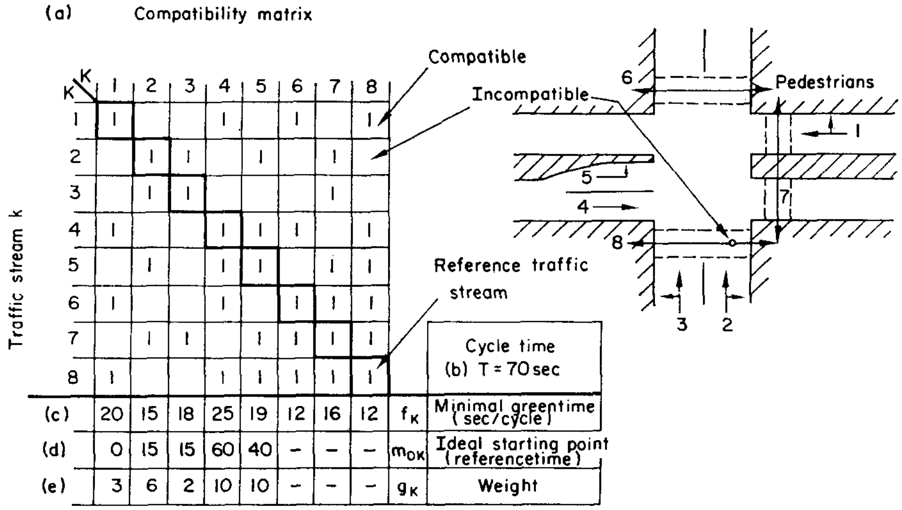
Single-stage controllers are used in many European cities. So far, they have not received much attention in the literature.

(a.2) *Two-stage controllers.* In two-stage controllers (Fig. 2b) the traffic lights are grouped into sets which are operated jointly. Each set is controlled by contacts that sit in a contact bench in an axial direction on the rotating cylinder, which forms the second stage of the controller. The time switch initiates changes of the signal display by sending an advance command to the promoter which then turns the cylinder into its next position. The signal displays are determined by the position of the cylinder. A light shows green whenever the corresponding contact at the cylinder is closed. With a given cylinder the sequence of the green intervals and the number of starting points that one can have are fixed.

Traffic-dependent controls frequently employ two-stage controllers. The starting points (times for the advance commands to the promoter) are then dependent on traffic measurements.

---

† This paper is based on the author's doctoral dissertation at the University Fredericiana in Karlsruhe (Germany).



K \ K	1	2	3	4	5	6	7	8		
1	1			1		1		1		
2		1	1		1		1			
3		1	1				1			
4	1			1	1	1		1		
5		1		1	1		1	1		
6	1			1		1	1	1		
7		1	1		1	1	1	1		
8	1			1	1	1	1	1		
(c)	20	15	18	25	19	12	16	12	f_K	Minimal greentime (sec/cycle)
(d)	0	15	15	60	40	–	–	–	m_{0K}	Ideal starting point (referencetime)
(e)	3	6	2	10	10	–	–	–	g_K	Weight

FIG. 1. Example 1.

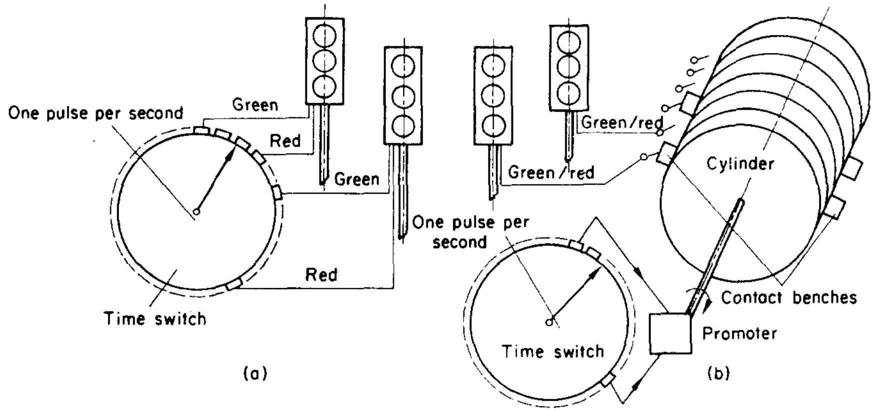


FIG. 2a. Single-stage control. b. Two-stage control.

(a.3) *Controller transitions.* For safety reasons it is necessary to have a minimal duration of any green interval for all lights. This minimal duration of the green must be assured, not only during stationary operation of the traffic controller, but also during the transition from one light schedule to another one and when the controller is being turned on or off. It is, therefore, not possible to have transitions at arbitrary times within a cycle. Customarily, the stationary use of a traffic light schedule is permitted to commence or terminate only at one particular instant during a cycle, called the "on–off point". The handling of controller transitions and the conditions which the on–off point must meet are different in single-stage and in two-stage controllers (see Fig. 3a).

For single-stage controllers one selects an instant of time within a phase as on–off point. The last starting point before this point and the first starting point subsequent to it must both be sufficiently far away to assure safety. The duration of the phase which contains the on–off point must be greater (or at least equal) to twice the minimal duration of a green interval. Furthermore, the phase which contains the on–off point must be the same in all schedules of an intersection. Frequently, it is referred to as the "zero phase"; the on–off point of schedules for single-stage controllers is also known as "good switching point" (G.S.P.). Single-stage controllers contain an auxiliary switch for each schedule, which interrupts all its connections to the time switch when the schedule is not used.

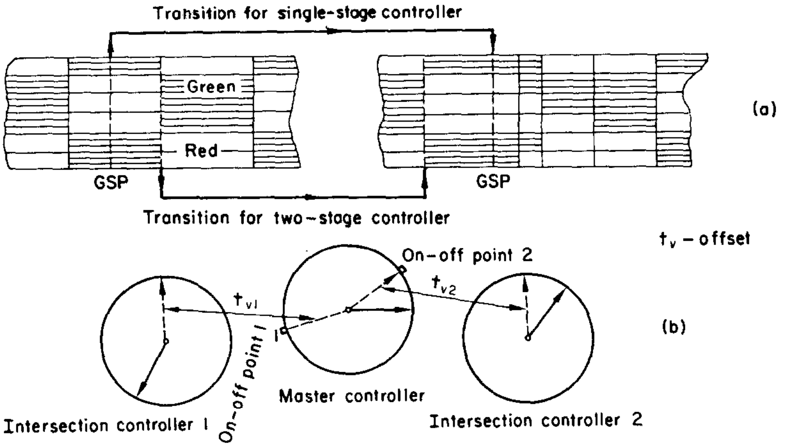


FIG. 3a. Controller transitions. b. Area control.

When changing between two schedules, the old schedule is disconnected from the time switch when the G.S.P. of that schedule is reached. The traffic lights will continue to show whatever they happen to be showing at the G.S.P. When the G.S.P. of the new schedule is reached by the time switch, the auxiliary switch of that schedule must close. When a single-stage controller is being turned on, the control must complete a "blind" cycle (control logic on, but power for the traffic lights off) to establish the zero phase before the traffic light displays commence.

In two-stage controllers a starting point is used as on–off point. It must be selected such that there is no traffic light in the schedule which displays green both before and after the on–off point. Therefore, no green interval is altered in duration if the schedule is turned on (off) at this point. Unless auxiliary equipment for a special phase is provided, all signals will display red during the transition between two schedules.

In coordinated systems of controllers the time of transition between two schedules (with either type of controllers) may be quite long (larger than a cycle time). This means a considerable disturbance of traffic flow, which practically forbids the change of traffic light schedules for the peak hours.

(a.4) *Master controller.* Coordinated controllers use the same cycle time at all intersections. The relative position of the schedules for different intersections with respect to time (offset) must be adjusted during the controller transition and should be monitored during stationary operation. Usually a master time switch is employed. It emits signals at the on–off points of the intersection schedules (see Fig. 3b).

(a.5) *Traffic light transitions.* The indications "amber" and "red and amber" are used to announce changes of signal display. The duration of these transitory indications is in many countries fixed by regulations. However, the "safety gaps" needed between the green intervals of incompatible traffic streams vary from case to case. Usually an "all-red" indication (both lights show "red") is needed between the green intervals for two incompatible traffic streams.

The amber, red and amber and all-red times should be generated by delays which are independent of the rotating controller stages. This will guarantee the proper length of these times even if the controller is operated in a traffic-dependent fashion or by hand. The supervisory logic for the single-stage controller mentioned under (a.2) cannot function properly if the transitory indications come directly from the time switch. If delays are used to generate the transitory times, there will only be two states for each signal which depend on the rotating controller stage. They are referred to as "green" and "red" even though they are slightly different (vs. time) from the signal displays.

For computational work it is convenient to make the reduction to two states per signal by including the subsequent safety gap into each green interval. The inclusion of the safety gap in the minimal green times in this paper is one reason for the stipulation that there must be exactly one green interval per cycle for each light.

The green times at the traffic light, at the rotating controller stage and in the computation of a traffic light schedule are all slightly different from each other (see Fig. 4b).

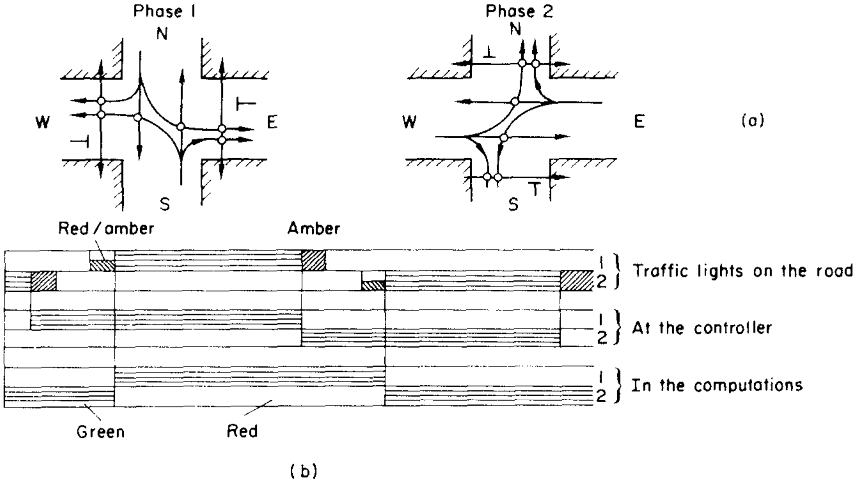


FIG 4a. Two-phase system. b. Display of lights for a pair of incompatible traffic streams.

(b) *Intersection layout and traffic light schedule*

Cars which arrive in the same approach lane to an intersection must be under the control of the same traffic light. The number of separate queuing areas provided by the geometry of the intersection specifies the maximum of the number of traffic lights that can be employed. If one green interval per cycle is required at all lights, this is also the maximum of the number of starting points that may be needed in the traffic light schedule.

(b.1) *Terms.* The following terms will be needed in the subsequent discussion:

1. A "traffic light" controls the right of way for a part of or all of the traffic arriving on one approach to an intersection.
2. A "traffic stream" is that part of the traffic which is under the control of the same traffic light.

Comment: in the discussion of traffic light schedules, the terms "traffic stream" and "traffic light" can be used interchangeably.

3. A "partial stream" is that part of the traffic which uses a specified approach and a specified exit of the intersection.

   Example: traffic stream 1 in Fig. 1 consists of the two partial streams "E–W" (straight ahead) and "E–N" (right turning).
4. A "conflict area" exists if the paths of two partial streams cross (or overlap).
5. A "conflict" exists if two partial streams are admitted simultaneously to their conflict area.

   Comment: in case of a conflict between two partial streams, they must "filter" through each other. The notions "incompatibility" and "conflict" are not equivalent. Example: traffic streams 1 and 6 in Fig. 1 are compatible even though there is a conflict between the partial stream "E–N" (right turning) and stream 6 (pedestrian).
6. A "storage pocket" is the queuing area for one traffic stream.

   Comment: a storage pocket may consist of more than one lane.
7. A "traffic light phase" (or for short a "phase") is a complete specification of signal indications which may occur simultaneously.

   Comment: in the model with two states/traffic stream ("green" = go, "red" = stop) a phase can be described by listing the traffic streams which go through. Examples: Figs. 4a, 5a and 5b.

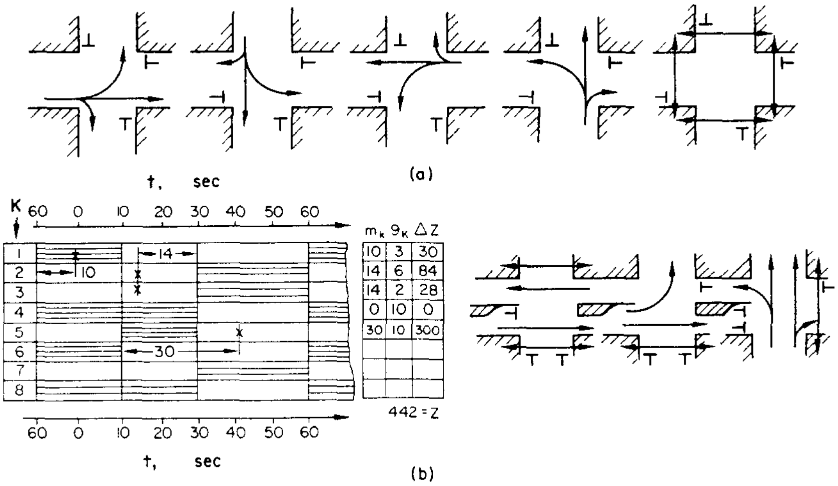


FIG. 5a. Multi-phase system. b. Split-phase system.

8. The "phase duration" is the length of time for which the light indications of the phase remain unchanged.

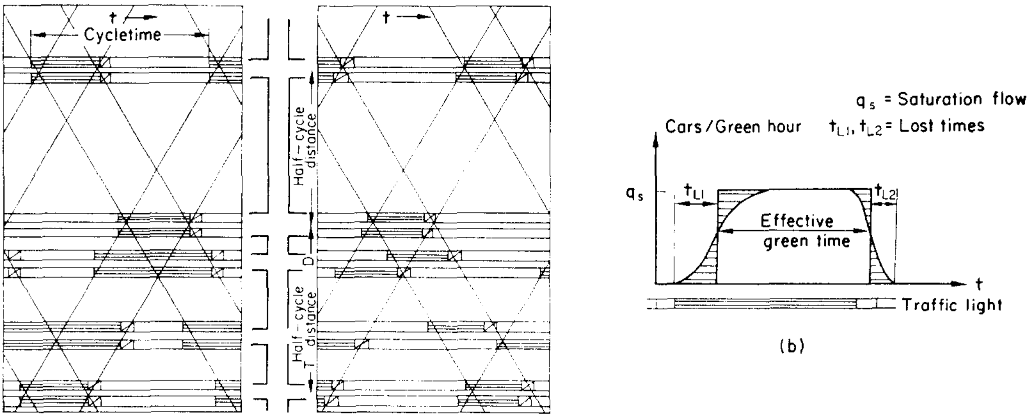
   Example: Fig. 5b shows a traffic light schedule with three phases. The phase durations are 20, 20 and 30 sec.

   Comments: the example in Fig. 5b is often called a "split-phase" system and considered as a variant of a two-phase system. However, with the above definition of the term "phase" it has three phases. It should be noted that this definition does not assume that each light will display green in only one phase. In this paper the light will be permitted to show green in several successive phases.

Some of the classical work on traffic light scheduling, e.g. Webster's (1958) investigation of the cycle time which gives minimum delay, implied that each light shows green in only one phase, so that there are no overlapping green intervals. His conclusion that an increase in the number of phases must mean an increase in "lost time", and a decrease in capacity does not hold if overlapping greens are permitted.

If the term "phase" is used as defined above, a new phase begins at each starting point of a schedule. The number of phases that a given two-stage controller can handle is then equal to the number of starting points provided on the cylinder. Later in the paper it will become clear that the above definition of "phase" is necessary for the mathematical treatment.

(b.2) *Two-phase system.* The two-phase system can work with the intersection layout that exists at intersections without traffic lights. Mainly for this reason, it was widely used when traffic lights were first introduced.



(a)

FIG. 6a. Time–distance diagram without (left) and with (right) different starting-points for opposing directions. b. Variation of discharge rate of queue in a fully saturated green interval.

The two-phase control of a regular four-approach intersection provides that all the traffic arriving in the same street (N–S or E–W, see Fig. 4a) should have the right to cross the intersection during the same time interval. It looks as if the control declared the intersection alternatingly to be part of one street and then of the other. The state of the controller which makes the intersection into a part of one of the streets is a phase.

Two-phase control has the following disadvantages:

1. Many conflicts.
2. Many partial streams use the same green interval.
3. Many partial streams use the same starting point.

Conflicts decrease traffic safety and lead to jammed intersections when traffic is heavy. The minimal duration for the green that is required will, in general, be different for all partial streams. Coordination requirements yield often different ideal starting points for the traffic streams of one phase. Especially the traffic that moves in opposing directions will usually have different ideal starting points. For an illustration see Fig. 6a.

(b.3) *Other scheduling systems.* The conflict between two streams disappears if the schedule is altered so that they pass the intersection successively. The number of conflicts can be decreased by increasing the number of phases. Figure 5a shows a five-phase system for the four-approach intersection which eliminates all conflicts. In this example the assumption that each light shows green in only one phase has been maintained. This schedule cannot be recommended because of the time lost for transitions.

An improvement over the two-phase system that has found wide application is the "split-phase system" which eliminates the conflict between the left-turning, partial streams and the opposing, straight-ahead traffic. Figure 5b shows a schedule with "split phase" for example 1. Here it is not true that each light shows green for only one phase. Lights 4 and 8 provide overlapping greens which extend through phases 1 and 2. Compared to the ordinary two-phase system the split-phase system requires additional storage pockets on the road and additional starting points at the intersection. Feuchtinger (1954) has suggested that one should, in continuation of the approach to scheduling taken in the split-phase system, use "staggered transitions" between the phases. His idea that as many overlapping greens as possible should be permitted in the schedules will be pursued in the study of phase sequences in this paper.

(b.4) *Channelization and progressive timing.* The attention of the drivers in an intersection is most heavily relied upon if the distance between the various conflict areas is small. Safety and the traffic volumes that can be handled in crossing lanes are largest if the crossing occurs under right angles and if the conflict area is well defined. Accurate channelization (e.g. by pavement markings) in the intersection and large intersection areas are therefore desirable. To smooth the flow of traffic through the intersection channelization of the partial streams in the approaches is advisable.

Because of channelization the partial streams appear more and more uncoupled, and the traffic control must aim at supervising the conflict areas independently. The distances between the conflict areas become so large that the travel times in the intersection can no longer be handled by "safety gaps". Progressive timing within the intersection may become necessary. In the limit the intersection becomes a network of one-way streets.

(c) *Measurements*

The minimal green times and the ideal starting points for coordinated intersections can be derived from measurements.

(c.1) *Minimal green times.* With queuing vehicles and fully used green intervals, the flow characteristic at a traffic light will look like the one depicted in Fig. 6b (see Webster, 1958). For the computation of schedules this characteristic is replaced by a rectangle with the same area. The height of the rectangle is the "saturation flow", its length is the "effective green time". At the beginning and the end of the green interval "lost times" occur. If there is an "all-red" between the end of the green interval and the commencement of "green" for the next incompatible traffic stream, this all-red shall be included in the lost time of the previous green. The minimal green time which then appears in a problem statement is the sum of the required effective green time per cycle and the lost times per green interval. A value for the minimal green time can, therefore, only be specified if the number of green intervals per cycle is known. Example: the 12 sec of minimal green time for traffic stream 6 in Fig. 1 contains 4 sec of effective green time (pedestrians enter the intersection) and 8 sec of lost time (pedestrians clear the road). If there were to be two green intervals instead of one (as assumed) the minimal green time of 12 sec would clearly be insufficient, since 16 sec would be needed for lost times alone. The minimal green in this case should be 20 sec.

The first experimental studies of the saturation flow have been made by Greenshields (1947). More recently Glück–Schelzke (1960) and Engel (1961) have taken measurements on European roads. A detailed discussion of saturation flow measurements is contained in

Webster (1958). All authors give rule-of-thumb values of slightly less than one car in 2 sec (1600–1800 cars/hr) for one lane.

It has been observed by Knoedel (1960) that travel time is the main criterion used by drivers in the selection of their route through a road network. A new traffic light schedule will affect the travel times and this will lead to an alteration of the traffic volumes at the intersection. Therefore, there is some uncertainty associated with the use of measured traffic volumes in the computations for a new traffic light schedule.

At well coordinated intersections, the cars will not start from rest when entering the intersection. Reliable values for the saturation flow in this case do not seem to exist. The data for discharging queues are frequently used even if queuing is not expected to occur.

(c.2) *Ideal starting points.* In a street network the distribution of the arrival times at most intersections is strongly influenced by the green times at adjacent intersections. Therefore, the green times at the intersections should be coordinated, i.e. the choice of the starting points for the green intervals should allow for the travel times between intersections.

For a single intersection—with or without coordination—minimization of average delay with bounds on the maximal duration of red is widely accepted as an objective. Unfortunately, this criterion is too complicated to apply to be useful for the simultaneous optimization at several intersections. Figures of merit which are formalizations of the intuitive approach used in the graphical time–space chart are being used instead. Morgan and Little (1964) have introduced the "maximization of bandwidth" as criterion for the determination of "green waves".

If one assumes the traffic light schedule for one intersection on a street to be known and the speed along that street to be predetermined, one obtains immediately the ideal starting points at the other intersections along the streets that lead to the intersection which already has a schedule. The only difference between a straight forward time = distance/speed calculation and the specifying of ideal starting points that arises is that the ideal starting points have to be specified modulo cycle time T. Usually the ideal starting points for the traffic that moves in opposing directions on a street will not be the same because the travel time between the two adjacent intersections is not exactly half a cycle time (see Fig. 6a for an example). For a "green wave" along a street, one needs acceptable values of speed and a cycle time for which the ideal starting points for the opposing directions come close to each other at all intersections.

The basic tool used in the study of coordination problems at present is the time–space diagram, e.g. Fig. 6a. Its user has to struggle with two difficulties: (a) He is using a trial-and-error procedure. Unless an "ideal solution" is found, there is no way of telling whether the best possible solution has been found. (b) It is practically impossible to visualize the coordination along more than one route simultaneously. Therefore, the coordination of a network by means of the time–space diagram means alternating trials on various routes.

The predetermined speed that is used to find the ideal starting points ought to be the "natural speed" which would result from the fundamental diagram or car-following theory, e.g. Leutzbach (1966) and Haight (1963, chapter 3). However, for streets in cities the data needed for this theoretical approach can usually not be obtained. Wardrop and Charlesworth (1952) have developed a method for the measurement of the average speed from a moving vehicle which can be used.

The distances between intersections, the necessity to keep the cycle time at values of approximately 1–2 min, and the minimal green times impose severe restrictions on the choice of the design speed. Usually the "natural speed" can only be accounted for in a very crude fashion. Fortunately, the drivers tend to adapt their speed to the design speed of a coordination in order to avoid stopping. Von Stein (1961) has found that even changes in design speed between successive road blocks are well adhered to if the design speeds are made known by speed signals. Von Stein believes that changes of up to 10 m.p.h. between successive road blocks are acceptable.

(d) *Problem statement*

Consider a road intersection with traffic lights. Assume a fixed-time traffic light schedule to be specified by:

(i) A(0, 1)—matrix which states for each pair of lights whether they may show green simultaneously.
(ii) Cycle time T.
(iii) Minimal green time $f_{k\,\mathrm{min}}$ per cycle for each light k.
(iv) "Ideal starting points" m_{0k} (desired green beginnings) for some or all lights.
(v) A weight g_k for the deviation m_k between the actual and the desired green begin of light k.

Find a schedule which minimizes the sum

$$z = \sum_k |m_k| g_k$$

subject to the constraints *a* to *c*.

The following additional stipulations must be met:

(vi) The deviations m_k have to be evaluated with respect to the closest ideal starting point ($|m_k| \leqslant T/2$).
(vii) The schedule shall provide exactly one green interval per cycle for each light.
(viii) Restrictions of the phase sequence which depend on the controller used must be included during the solution process.

Figure 1 shows a sample problem.

(e) *Justification of the model*

(e.1) *Compatibility matrix.* The provision of many storage pockets at intersections means that many partial streams are uncoupled. The traffic light control then comes close to supervising the conflict areas separately. The compatibility matrix which specifies each conflict separately does not include any restrictions which are not necessary for safety reasons. It leaves the maximal possible flexibility for the positioning of the green intervals to meet coordination requirements. The *a priori* assumption of a phase sequence is certainly much more restrictive.

The determination of the entries in the compatibility matrix is at this time left to human judgement and experience—as was so far the case with the choice of phases and phase sequences. Although this is still subjective, the reduction to more elementary isolated decisions should be an improvement over the existing practice.

(e.2) *Cycle time.* In coordinated systems there is one common cycle time for many intersections. The necessity to find a schedule with a given cycle time does frequently arise. To find the cycle time for a network is a complex problem in itself; it will not be discussed here.

(e.3) *Green intervals.* The determination of minimal green times from the traffic volumes must allow for many local factors (like slopes, road surface, geometric dimensions of the intersection, etc.). It should, therefore, be left to the engineer familiar with the intersection and not included in a formal treatment of traffic light scheduling.

The minimal green times contain the effective green time per cycle and the lost time per green interval (see Section c). A value of minimal green time per cycle can, therefore, only be specified if the number of green intervals per cycle is known. Existing traffic light schedules, provide as a rule, one green interval per cycle at all lights. Morgan and Little (1964) have found some justification for this ("bandwidths are unsplit") in their studies of coordination.

In this paper it is stipulated that there must be one green interval per cycle at all lights. Since each light is permitted to be green in an arbitrary number of (successive) phases, this stipulation will provide the basis for the selection of phase sequences.

(e.4) *Weighting of starting-point deviations.* The optimization criterion "weighted sum of starting-point deviations" represents a new attempt to formalize and generalize the intuitive optimization criterion used in the time–space diagram. The criterion differs from the "maximal band width criterion" in that it separates the specification of necessary green times from the study of the coordination requirements. Independent green time specifications for the intersections in a green wave are desirable, e.g. because the volumes vary from intersection to intersection. The separate formulation of the coordination objectives eliminates the restriction to the study of coordination along one path in the network, which is part of the space–time diagram and the bandwidth approach.

Progressive timing requires specification of "ideal differences" between starting points and not ideal starting points specified relative to a fixed reference time. Such requirements have not been included expressedly in the problem statement. However, they can be included as will be shown by an example at the end of the paper.

Differences between ideal and actual starting points mean that the speed for driving without stopping must differ from the original design speed (see Fig. 15). Since drivers adhere to such variations in the required speed as long as they are not too large, schedules which contain small starting-point deviations should be satisfactory. They are, of course, inferior to an "ideal solution" ($\sum_k |m_k| g_k = 0$), which contains no starting-point deviations at all.

The choice of the weights is left to human judgement. This seems reasonable, since the priorities that should enter into a coordination problem (e.g. distinguish between high-priority arterial routes and a local business street) cannot be obtained from measurements alone.

(f) *Mathematical approach*

The problem will be attacked in three separate steps:

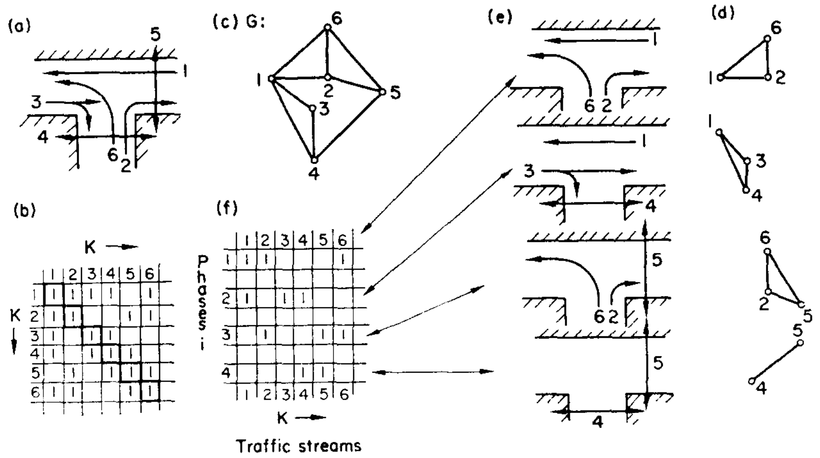
1. Find all the phases that agree with the compatibility matrix.
2. Construct all maximal phase sequences which yield one green interval per cycle for each light out of these phases.
3. Find the optimal schedule (phase durations, offset) for each possible phase sequence (separately) and then select the best solution.

The breakdown into these steps provides the possibility to eliminate answers which do not fit the control equipment after each step. The restrictions which are imposed by the controller transitions can be incorporated in step 2.

1. PHASES

(1.a) *Compatibility matrix, compatibility graph and traffic light phases*

A pair of traffic streams is "compatible" if the two traffic streams may cross the intersection simultaneously, it is "incompatible" otherwise. The compatibility matrix contains a 1 in row *i* and column *k* if traffic streams *i* and *k* are compatible; if they are incompatible, the entry is 0 or blank. The compatibility matrix is symmetric with respect to its main diagonal. The elements on the main diagonal are all set equal to 1. Example 2 (Fig. 7a, b) provides an illustration. The traffic streams 1 and 2 are compatible; traffic streams 1 and 5 are incompatible.



(b)

	1	2	3	4	5	6
1	1	1	1	1		1
2	1	1			1	1
3	1		1	1		
4	1		1	1	1	
5		1		1	1	1
6	1	1			1	1

(f)

	1	2	3	4	5	6
1	1	1				1
2	1		1	1		
3		1			1	1
4				1	1	

FIG. 7. Example 2.

The information contained in the compatibility matrix can be represented as a symmetric graph *G* called "compatibility graph". The compatibility graph has a node for each traffic stream. Two nodes are connected by an undirected edge if and only if the corresponding pair of traffic streams is compatible. The compatibility graph has no self-loops (there are no edges connecting with both ends to the same node). Figure 7b, c shows the compatibility matrix and the compatibility graph for the example.

Every traffic stream that goes through in a phase must be compatible with all other traffic streams in that phase. All edges that can exist between the nodes of these traffic streams must, therefore, be present in the compatibility graph. The nodes and edges which correspond to a traffic light phase form a complete subgraph or "clique" of the compatibility graph. For the sake of efficiency each traffic light phase should let through as many traffic streams as possible. There must not be a traffic stream compatible with all the traffic streams in a phase which does not go through in that phase. The phases, therefore, correspond to the maximal complete subgraphs or "dominant cliques" of the compatibility graph. Figure 7d shows the dominant cliques for the compatibility graph in Fig. 7c. Figure 7e gives sketches of the corresponding traffic light phases in the intersection.

The description of a phase by listing all traffic streams that go through can also be given in the form of an indicator composed of (0, 1) elements. The indictator contains a position for each traffic stream. The positions of those traffic streams which go through contain a 1, the others contain a 0 or blank. Examples: Fig. 7f.

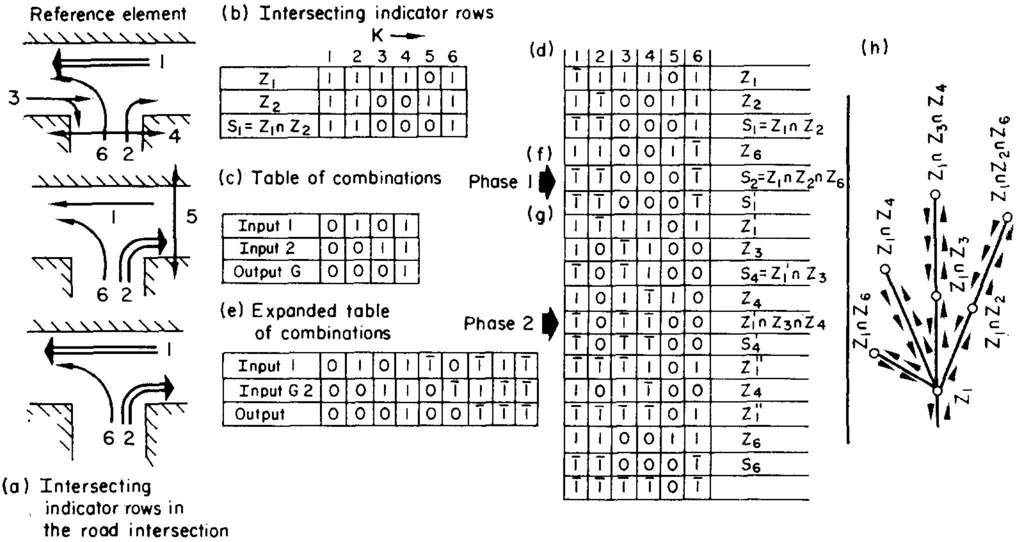
(1.b) *An algorithm for the determination of all phases*

In the sequel an algorithm will be described which obtains all the phases by means of boolean operation performed on the rows of the compatibility matrix. More generally speaking, this is an algorithm which determines the family of dominant cliques (maximal complete subgraphs) for a symmetric graph from a nodal incidence matrix.

The rows of the compatibility matrix can be considered as indicators for subsets of the universal set that contains all the traffic streams at the road intersection. Each subset consists of a "reference traffic stream" specified by the 1-element on the main diagonal and those traffic streams which are compatible with this reference traffic stream.

The set of traffic streams which are compatible with two reference streams is obtained if the intersection [here "intersection" (symbol $\cap$) is the term from set theory not the "road intersection"] of the corresponding sets is formed.

The intersection contains those traffic streams which occur in both sets. Figure 8a shows an example. The upper two sketches show traffic streams 1 and 2 of Fig. 7 together with all streams with which they are compatible. (The reference stream in each illustration is drawn heavy.) The bottom figure shows the intersection with the two reference streams. It should be noted that the intersection of two subsets (rows) is not meaningful here if it does not contain both reference elements.



(b) Intersecting indicator rows

	1	2	3	4	5	6
Z_1	1	1	1	1	0	1
Z_2	1	1	0	0	1	1
$S_1 = Z_1 \cap Z_2$	1	1	0	0	0	1

(c) Table of combinations

Input 1	0	1	0	1
Input 2	0	0	1	1
Output G	0	0	0	1

(e) Expanded table of combinations

Input 1	0	1	0	1	$\bar{1}$	0	$\bar{1}$	1	$\bar{1}$
Input G 2	0	0	1	1	0	$\bar{1}$	1	$\bar{1}$	$\bar{1}$
Output	0	0	0	1	0	0	$\bar{1}$	$\bar{1}$	$\bar{1}$

(d)

1	2	3	4	5	6	
$\bar{1}$	1	1	1	0	1	Z_1
1	$\bar{1}$	0	0	1	1	Z_2
$\bar{1}$	$\bar{1}$	0	0	0	1	$S_1 = Z_1 \cap Z_2$
1	1	0	0	1	$\bar{1}$	Z_6
$\bar{1}$	$\bar{1}$	0	0	0	$\bar{1}$	$S_2 = Z_1 \cap Z_2 \cap Z_6$
$\bar{1}$	$\bar{1}$	0	0	0	$\bar{1}$	S_1'
1	$\bar{1}$	1	1	0	1	Z_1'
1	0	$\bar{1}$	1	0	0	Z_3
$\bar{1}$	0	$\bar{1}$	1	0	0	$S_4 = Z_1' \cap Z_3$
1	0	1	$\bar{1}$	1	0	Z_4
$\bar{1}$	0	$\bar{1}$	$\bar{1}$	0	0	$Z_1' \cap Z_3 \cap Z_4$
$\bar{1}$	0	$\bar{1}$	$\bar{1}$	0	0	S_4'
$\bar{1}$	$\bar{1}$	$\bar{1}$	1	0	1	Z_1''
1	0	1	$\bar{1}$	0	0	Z_4
$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	0	1	Z_1''
1	1	0	0	1	1	Z_6
$\bar{1}$	$\bar{1}$	0	0	0	$\bar{1}$	S_6
$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	0	$\bar{1}$	

FIG. 8. Determination of phases (computational examples).

The indicator for the intersection is obtained if the operation "and" is performed with corresponding elements of the two indicator rows. The combination rules for the "and" are shown in Fig. 8c. In Fig. 8b they are applied to rows 1 and 2 of Fig. 7b. The computation shown in Fig. 8b is unsatisfactory since it does not keep track of the reference elements. To remedy this a boolean algebra with three-valued variables will be employed instead of the two-valued variables of Fig. 8b. The symbols and their meanings are:

0—traffic stream not present;

1—traffic stream present;

$\bar{1}$—reference traffic stream.

For the three-valued variables there are nine possible combinations of two elements. The "and operation" for these shall be defined by Fig. 8e.

The first three rows (from the top) of Fig. 8d show the indicators, which now correspond to Fig. 8a. They also serve to illustrate some of the combination rules laid down in Fig. 8e.

Intersections of rows, which do not contain all the reference elements, will not be formed if the 1-elements without flags in a row are used to determine the row with which an intersection shall be formed (this exploits the symmetry of the compatibility matrix). Example: Row 1 contains as its first 1-element without flag (from the left) the 1 in position 2. Therefore, rows 1 and 2 shall be intersected (see Fig. 8d), which yields a new row S_1. In S_1 the 1-element without flag in position 6 indicates that S_1 and row 6 shall be intersected next. The result is row S_2.

If we use the 1-elements without flag to guide the progress of the computation, the 1-elements with flags become a memory for the previously executed steps of the computational process. Each row that is contained in the result is indicated by a flagged 1-element. Later on this memory capacity associated with the flags will be exploited further. Then the rules $\bar{1}$ and $\bar{1} = \bar{1}$, 0 and $\bar{1} = 0$ and $\bar{1}$ and $0 = 0$ will be needed. The interpretation of the $\bar{1}$ as a "reference" element will then not always be meaningful.

The process of intersecting the previous result of the computation with another row which corresponds to an unflagged 1-element is called "forward step". Clearly after a finite number of forward steps, the computation must terminate because there is no unflagged 1-element left in the result. An indicator which has flags on all 1-elements corresponds to a maximal set of mutually compatible traffic streams—which is a phase. Example: row S_2 of Fig. 8d represents phase 1 of Fig. 7.

Next, it shall be shown how one can proceed to find all phases that contain one particular reference stream. After the indicator for a first phase has been arrived at, a new computational rule, the "backstep", must be applied. In the backstep, the second to last intermediate result is taken up again, and the 1-element which has guided the last forward step is converted into a flagged 1-element. Example: in Fig. 8d one returns from row S_2 to row S_1 and attaches a flag to the 1 in position 6. Thus, row S_1' is obtained from row S_1. The flag on the 1-element has now the function of the "Ariadne-string" in the searching of a labyrinth.

After the backstep was executed, one must try to perform new forward steps. In the example, row S_1 does not contain a 1-element without flag and another backstep must follow. Row S_1 looks, in fact, identical to row S_2. It is apparent that more than one indicator representing a particular phase will arise in the computational process. In order to recognize each phase only once as a new result, the following rule is used: an indicator represents a new phase if (a) it contains only flagged 1-elements and (b) the number of flagged 1-elements is equal to the number of forward steps minus the number of backsteps that have been performed. Under (b) the selection of an initial row in the compatibility matrix must be counted as a forward step.

The sequence of indicator rows which are generated by the forward steps and backsteps can be represented as a tree. The end points of this tree are phases, repetition of known phases and the row initially taken from the compatibility matrix. For the example, the tree is sketched in Fig. 8b.

Finally, the phases which do not contain the first reference element have to be found also. One way to include these would be to enlarge the compatibility matrix by a fictitious traffic stream which is compatible with all traffic streams at the road intersection. (A row and a column containing only 1-elements would have to be added to the compatibility matrix.) The computation would then be started with the fictitious reference element. After it finished, the fictitious element would have to be deleted from the results.

Practically equivalent to this is a change of the compatibility matrix. After the possibilities for forward steps and backsteps based on the initial reference element have been exhausted, one replaces all the 1-elements in the column of the initial reference element by flagged 1-elements. Thereafter, a new initial reference is selected. If this is continued until all traffic streams have been used as initial elements, the complete "family" F of phases will result. Because of the alterations in the compatibility matrix, the rules $\bar{1}$ and $\bar{1} = \bar{1}$ and 0 and $\bar{1} = 0$ are necessary.

Figure 9 shows a block diagram for the algorithm. Figure 10 shows the phases for example 1 (Fig. 1).

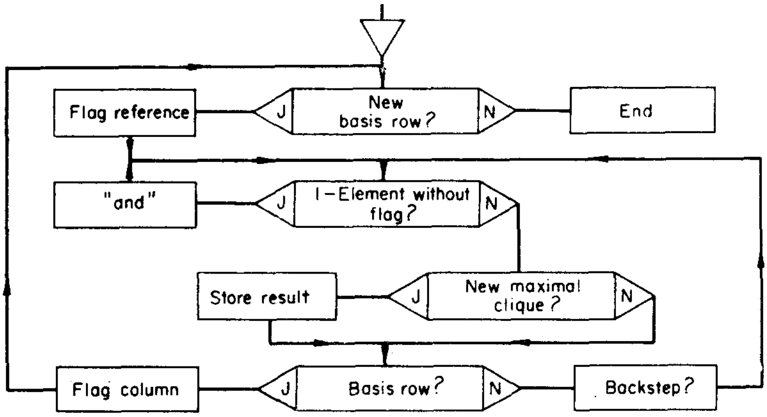
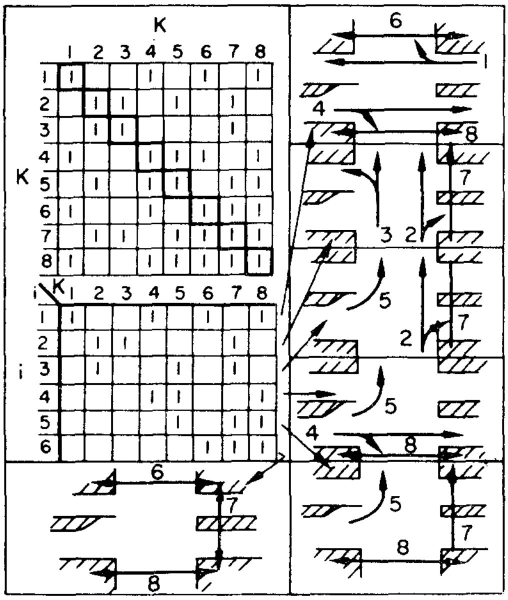


FIG. 9. Algorithm for the determination of the phases (dominant cliques).



K	1	2	3	4	5	6	7	8
1	1			1		1		1
2		1	1		1		1	
3		1	1				1	
4	1			1	1	1		1
5		1		1	1		1	1
6	1			1		1	1	1
7		1	1		1	1	1	1
8	1			1	1	1	1	1

i \ K	1	2	3	4	5	6	7	8
1	1			1		1		1
2		1	1				1	
3		1			1		1	
4				1	1			1
5					1		1	1
6						1	1	1

FIG. 10. Phases for example 1.

Other algorithms for the determination of dominant cliques have been described by Fulkerson and Gross (1964), Kirchgässner (1964) and Maghout (1959). They seem to be less efficient when used in a digital computer.

(1.c) *Incompatibility graph and blocking groups*

Instead of the compatibility graph G which contains an edge for every pair of compatible traffic streams, one can draw an "incompatibility graph" which has an edge for each pair of incompatible traffic streams. The incompatibility graph is the complementary graph to the compatibility graph and shall be denoted by $\bar{G}$. $\bar{G}$ has no self-loops. For example 2 (from Fig. 7) G and $\bar{G}$ are shown side by side in Fig. 11.

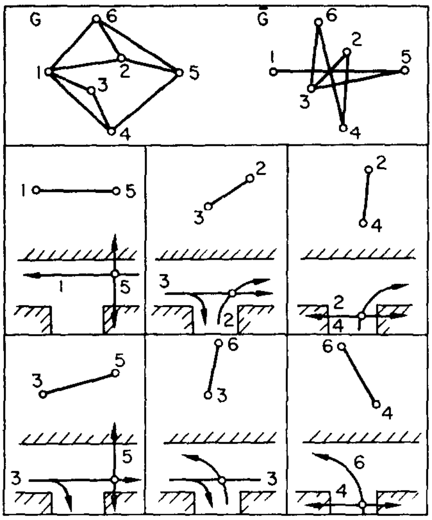


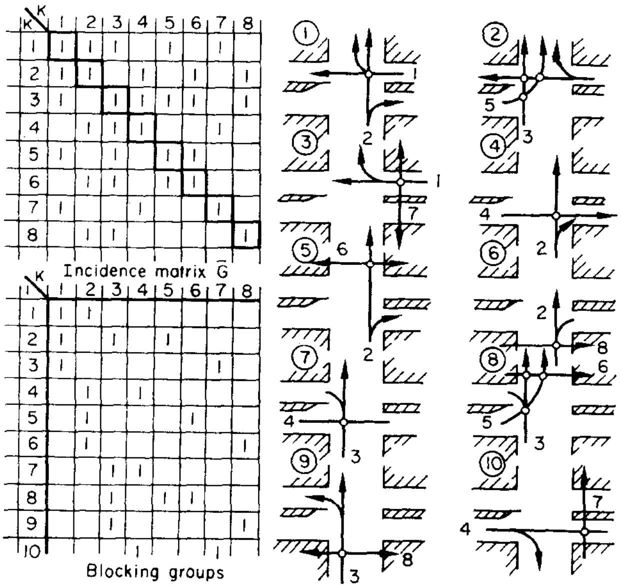
FIG. 11. Blocking groups for example 2.

If two traffic streams are incompatible, the green intervals of their traffic lights may not overlap. The green intervals of traffic streams, which form a clique of $\bar{G}$, must all occur at different times in the traffic light schedule. The sum of the minimal green times for traffic streams which form a dominant clique in $\bar{G}$, therefore, yields a lower bound on the cycle time of the schedule. Example: in Fig. 1 the traffic streams 1, 2 and 5 are mutually incompatible and form a clique of $\bar{G}$. The green times required for these streams are 20, 18 and 19 sec respectively. Clearly, the problem would not possess a solution if it had been required to provide these green times within less than $20+18+19 = 57$ sec. The dominant cliques of $\bar{G}$ will be called "blocking groups". The sum of the minimal green times for the elements of a blocking group is the "length of the blocking group".

No schedule meeting conditions (a)–(c) can exist if there is a blocking group longer than the cycle time required under (b). The converse of this statement is not true in general. There need not be a feasible schedule for cases where no blocking group is longer than the desired cycle time. However, for phase sequences which meet the criteria set forth in the discussion of phase sequences in the following paragraph, a solution will always exist as will be shown later.

Figures 11 and 12 show the dominant cliques of $\bar{G}$ for examples 1 and 2.

The algorithm given in Section 1.b can be used to determine the dominant cliques of $\bar{G}$. Before starting the computation, the compatibility matrix must be complemented. In the positions outside the main diagonal, the 1 is replaced by 0, and the 0 is replaced by 1. The elements on the main diagonal are again 1. Figure 12 shows the nodal incidence matrix for the complementary graph $\bar{G}$ of example 1 (Fig. 1) as an illustration.



K \ K	1	2	3	4	5	6	7	8
1	1	1	1		1		1	
2	1	1		1		1		1
3	1		1	1	1	1		1
4		1	1	1			1	
5	1		1		1	1		
6		1	1		1	1		
7	1			1			1	
8		1	1					1

Incidence matrix $\bar{G}$

I \ K	1	2	3	4	5	6	7	8
1	1	1						
2	1		1		1			
3	1						1	
4		1		1				
5		1				1		
6		1						1
7			1	1				
8			1		1	1		
9			1					1
10				1			1	

Blocking groups

FIG. 12. Blocking groups for example 1.

2. PHASE SEQUENCES

(2.a) *Necessary properties*

Sets of phases and phase sequences can be represented as unordered and ordered sets of indicator rows. They appear then as a rectangular matrix **A** which has a row for each phase C_i and a column for each traffic light K. Figure 14a shows an example of an unordered set of phases; Fig. 13 gives some examples of ordered phase sequences. Phase sequences must meet two conditions:

1. Each light must show green in at least one phase.
2. All the phases in which a light displays green must occur consecutively.

Condition 1 requires that there be at least one 1-element in each column of **A**. This does not depend on the ordering of the phases and can easily be checked. From now on it will be tacitly assumed that this requirement is met.

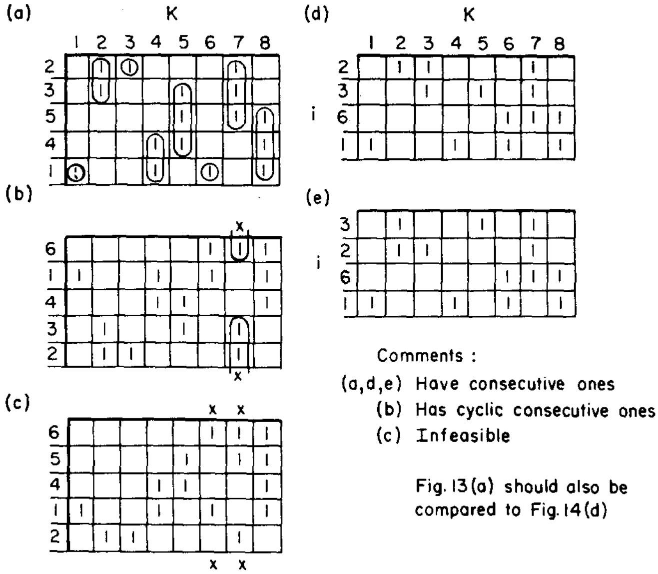
Condition 2 is concerned with the positions of the 1-elements in those columns which contain more than one of them. Two phases shall be said to "overlap" if there are one or more columns of **A** in which they both contain a 1. Two cases have to be distinguished:

(i) *The phase sequence is an "open sequence"*. In this case there exists a "first" phase C_a and a "last" phase C_n which do not overlap:

$$C_a \cap C_n = \phi \tag{1}$$

The matrix **A** for the phase sequence can be written with C_a and C_n forming the top and bottom rows respectively. Condition 2 requires then that all the 1-elements in the columns occur consecutively. In the example in Fig. 13a this property is emphasized by encircling the "consecutive ones". Following Fulkerson and Gross (1964) a matrix **A** will be said to possess the "consecutive-ones property" if there exists a permutation of its rows with consecutive ones in the columns.

If a set of phases has the consecutive-ones property, they can be used for schedules in two-stage controllers. The on–off point is then located between the "last" phase C_n and the first phase C_a. If the phase which includes the on–off point (G.S.P.) of a single-stage controller for the intersection is present among the phases, such a set can also be employed on a single-stage controller.



Comments :
(a,d,e) Have consecutive ones
(b) Has cyclic consecutive ones
(c) Infeasible

Fig. 13(a) should also be compared to Fig. 14(d)

FIG. 13. Phase sequences.

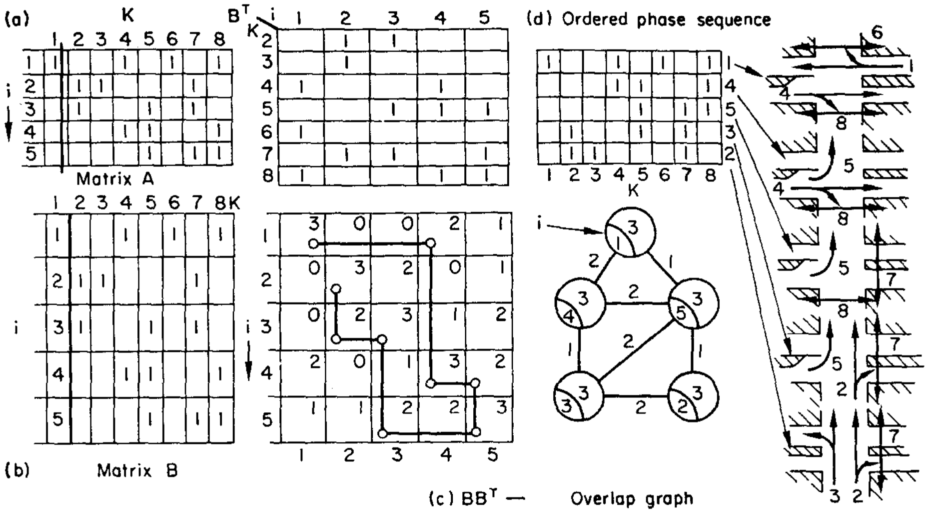


FIG. 14. Ordering a set of phases.

(ii) *The phase sequence is a "closed sequence"*. Phases which cannot be ordered such that there is no overlap between a first and last phase of the sequence may still provide consecutive ones if the first and last rows of **A** are considered as adjacent to each other. Figure 13b shows an example. Matrices of this kind will be said to have "cyclic consecutive ones".

Phase sequences with cyclic consecutive ones cannot be used in two-stage controllers. They can be used in single-stage controllers provided the zero phase, which contains the on–off point (G.S.P.) of the intersection, is contained in the sequence.

(2.b) *Sequencing of a set of phases*

Let a set of phases be represented by a matrix **A**, which has an arbitrary order of its rows. A construction shall be given which yields all the sequences with consecutive ones, that can be formed from these phases, provided there are any. Its basic idea is to eliminate most permutations from actual inspection by consideration of the overlap between the phases. To begin with, some necessary conditions for phase sequences with consecutive ones will be stated:

1. The first phase C_a and the last phase C_n of a phase sequence are called its end points. The end points must satisfy the condition

$$C_a \cap C_n = \phi \tag{1}$$

Phases which overlap with all the other given phases cannot be end points.

2. A second condition for the end points can be obtained after a slight simplification of the matrix **A**. Since duplications of columns (e.g. columns 1 and 6 in Fig. 14a) have no effect upon the consecutive ones, they should be deleted. A matrix **A**, the columns of which are all different, will be denoted as matrix **B**. For an end point of a matrix **B**, the number of elements in the phase must be by one smaller than the overlap with the neighboring phase.

$$|C_a| = |C_a \cap C_{a+1}| + 1$$

and

$$|C_n| = |C_{n-1} \cap C_n| + 1 \tag{2}$$

This is a second condition for the end points.

3. Now, assume an end point C_a to be known. For any two phases C_i and C_k in a sequence with consecutive ones C_i will have to be closer to C_a than C_k if

$$|C_a \cap C_i| > |C_a \cap C_k| \tag{3}$$

and vice versa.

Therefore, if one end point C_a and part of a phase sequence with consecutive ones, say $C_a, \ldots, C_e, \ldots, C_i$ is known, the next phase C_k that follows must be the one (out of those left) which has the strongest overlap with C_i. Furthermore, because of (3), one must have $|C_e \cap C_k| < |C_i \cap C_k|$ for all the other phases C_e in the known part of the sequence. If the latter is not the case, a "contradiction" is present, and the sequence cannot have consecutive ones.

Repeated application of these rules based on (3) must lead to an ordering for the given set of phases if they have the consecutive-ones property. Since conditions (1)–(3) are necessary but possibly not sufficient, the sequences found have to be tested by inspection.

The search for sequences which meet (1)–(3) can be carried out in the "overlap graph" of the phases. This graph has a node for each phase and an edge between two nodes if the phases overlap. The number of 1-elements in a row (traffic streams in a phase) is entered into the node; the amount of overlap between two phases is attached as a label to the corresponding edge. Figure 14a, b, c shows the matrices **A** and **B** and the overlap graph for an example.

Phase sequences are indicated in the overlap graph by a path which goes through each node exactly once and proceeds along edges of the graph. Such a path is usually called a "Hamiltonian path". Since the first and the last node of the path are not joined by an edge, it is an "open" Hamiltonian path. If the overlap graph is not connected, this reasoning applies for each component and any combination of paths from the various components is a solution.

To find all paths which satisfy (1)–(3), one can first locate all nodes which are candidates for end points and then trace all satisfactory paths starting at these nodes.

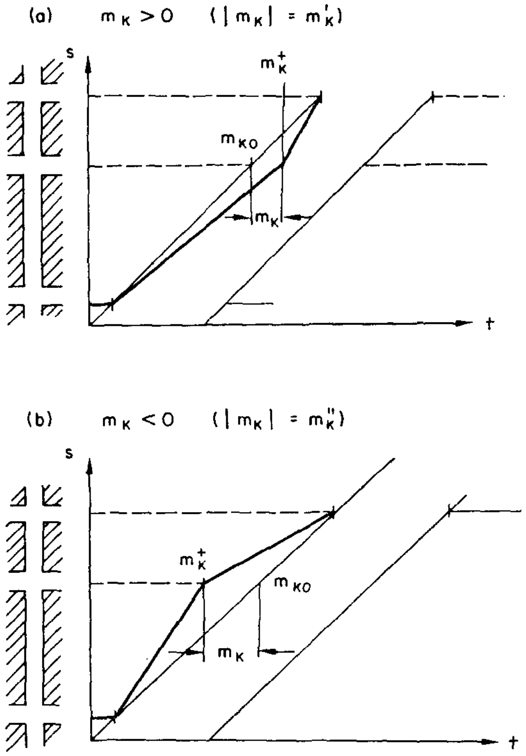


FIG. 15. Starting-point deviations in time–space chart.

In Fig. 14c the nodes 1, 2, 3 and 4 satisfy conditions (1) and (2) and have to be considered as possible end points. Now assume node 1 as end point. According to rule (3) the sequence must always proceed along the strongest edge available for departure. This yields the sequence 1–4–5–3–2 which is sketched in Fig. 14c. Inspection shows that the sequence has consecutive ones. Figure 14d shows this sequence in the intersection.

Starting with node 2 the same sequence results in reverse order. If node 3 is taken as end point, there are two choices for progression. One can start with 3–2 and node 5 would have to follow node 2. However, the overlap between 3 and 5 is stronger than between 2 and 5, the edge 2–5 is being bypassed by the stronger edge 3–5, which is a contradiction. If one starts from node 3 and goes 3–5, one must continue 3–5–4–1 and cannot reach node 2, which is again a contradiction. Starting out from node 4, it is not possible either to arrive at a Hamiltonian path without contradiction. Thus, the sequence 1–4–5–3–2 is the only possible one for the example.

The consideration of all sets of phases that can be formed from the six phases for intersection 1 (see Figs. 1 and 10) yields four maximal phase sequences:

1–4–5–3–2

1–6–5–3–2

1–6–2–3

2–6–1–4

Since these sequences can also be used in reverse order, there are eight sequences for consideration when a schedule is desired.

Remarks

1. Since the sufficiency of (1)–(3) has not been investigated, it remains necessary to set up the sequences that obey these conditions and to test for consecutive ones by inspection.
2. There can be more than one sequence for a given group of phases. The phases 1, 2, 3 and 6 of example 1 are a case in point (see Fig. 13d, e).
3. If there is a set of n phases with the consecutive-ones property, then any subset of these must have the property.
4. If a set of n phases does not have the consecutive-ones property, then any larger set which contains these as a subset cannot have the property.
5. For the determination of traffic light schedules later on, only "maximal phase sequences" will be needed. A phase sequence which can be obtained from a larger one by dropping a phase is not of interest. Example: the sequence 1–6–3–2 is not of interest since it is contained in the maximal sequence 1–6–5–3–2. However, 1–6–2–3 is a new sequence.

A previous investigation of the consecutive-ones property was made by Fulkerson and Gross (1964). They give a construction for a sequence with consecutive ones which employs the overlap between the columns. They also state the following property:

A(0, 1) matrix, which has the consecutive-ones property is unimodular. (All determinants that can be formed from such a matrix are -1, 0 or $+1$) (4)

Proof of this theorem can be obtained from induction on the rows or from the conditions for unimodularity stated by Hoffman and Kruskal (1956).

The overlap of two phases is obtained if the inner product of the two indicator rows is formed. The number of 1-elements in an indicator is given by the inner product of the indicator with itself. The matrix $\mathbf{B}.\mathbf{B}^T$ ($\mathbf{B}$ multiplied with the transposition of $\mathbf{B}$) contains all the information needed for the overlap graph. It is, in fact, not necessary to draw the overlap graph. The paths satisfying (1)–(3) can be searched directly in $\mathbf{B}.\mathbf{B}^T$. Example: Fig. 14b, c.

(2.c) *Reconstruction of compatibility graph and compatibility matrix*

The family F of phases completely characterizes the compatibility graph.

$$G = G(F) \tag{5}$$

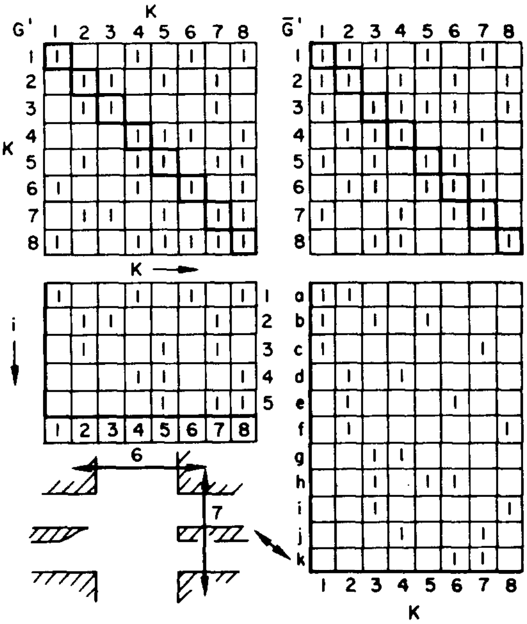
This must be true since each edge of G occurs in at least one maximal complete subgraph; and, on the other hand, the subgraphs have no edges which do not occur in G. The graph G is the union of its maximal complete subgraphs.

The construction of the compatibility graph which corresponds to a given set of phases is, therefore, very simple: to find the row of the compatibility matrix which corresponds to a particular reference element, one must take the union of all the phase indicators which contain the reference element. Example: row 8 of the compatibility matrix in Fig. 1 is the union of the phases C_1, C_4, C_5 and C_6 (Fig. 10). To obtain a row of the compatibility matrix, one writes all the phase indicators which contain the reference element "on top of each other".

If it is desired to draw the compatibility graph by inspection of the phases, it is more convenient to compare the columns since they correspond to the nodes. Two nodes are joined by an edge if, and only if, there is overlap between the columns in the phase sequence.

(2.d) *Phase sequence and compatibility graph*

Usually the family F of phases does not have the consecutive-ones property and a traffic light schedule can only consist of a subset T of F. This corresponds to an altered compatibility graph G'. If we reconstruct the compatibility graph from the subset T instead of the family F, only those compatibilities which occur in T will show up in G'. Example: if a compatibility graph G' is reconstructed from the first five phases of example 1 (Fig. 14), the traffic streams 6 and 7 are incompatible. In G these streams are compatible; however, this is utilized only in phase 6 which is not in the sequence (see Fig. 10) used for the reconstruction (see Fig. 16).



G'	1	2	3	4	5	6	7	8
1	1			1		1		1
2		1	1		1		1	
3		1	1				1	
4				1	1	1		1
5		1		1	1		1	1
6	1			1		1		1
7		1	1		1		1	1
8	1			1	1	1	1	1

$\bar{G}'$	1	2	3	4	5	6	7	8
1	1	1	1		1		1	
2	1	1		1		1		1
3	1		1	1	1	1		1
4		1	1	1			1	
5	1		1		1	1		
6		1	1		1	1	1	
7	1			1		1	1	
8			1	1				1

1			1		1		1	1
	1	1				1		2
	1			1		1		3
			1	1			1	4
				1		1	1	5
1	2	3	4	5	6	7	8	

a	1	1						
b	1		1		1			
c	1						1	
d		1		1				
e		1				1		
f		1						1
g			1	1				
h			1		1	1		
i			1					1
j				1			1	
k						1	1	
	1	2	3	4	5	6	7	8

FIG. 16. Phases and blocking groups for G'.

The compatibility graph G of an intersection contains all compatibilities which are permitted, whereas the compatibility graph G' which corresponds to a particular phase sequence contains only those compatibilities which are utilized in that phase sequence.

Let T be a subset of the family of phase F and assume that T has the consecutive-ones property. The following can be shown:

1. $G' \subset \subset G$ — G' is strictly contained in G. It can be obtained from G by deleting edges (6)

2. $F' = T$ — T is the complete family of phases F' for G' (7)

Because of the second property G' is a graph whose family of phases has the consecutive-ones property. Such graphs are called interval graphs. Properties and characterizations of interval graphs are given in Lekkerkerker and Boland (1962), Fulkerson and Gross (1964) and Gilmore and Hoffman (1964), while Busacker and Saaty (1966) give a short summary.

It is important to distinguish between G and G' mainly because of their complementary graphs and their blocking groups. From $G' \subset \subset G$ follows $\bar{G}' \supset \supset \bar{G}$. The complementary graph of G' has more edges than the complementary graph of G. Therefore, the blocking groups of G' are larger in size or larger in number (or both) than those of G. It depends on the blocking groups of G', not on those of G, whether a phase sequence can be timed to satisfy the requirements (a) compatibilities, (b) cycle time and (c) minimal green times of a problem.

3. TRAFFIC LIGHT SCHEDULES

(3.a) *Characterization of schedules*

It is now desired to find values for the offset (displacement of the beginning of the first phase against zero reference time) and the phase durations which satisfy all the constraints and minimize $z = \sum_k |m_k| g_k$. The phase sequence is assumed to be given and to have consecutive ones.

The treatment of the optimization problem will be broken down into two parts:

1. A class of "feasible solutions" is introduced. The feasible solutions meet those requirements which are not related to coordination. A simple way to construct an initial feasible solution will be presented.
2. Ways to replace a known feasible solution by an improved one will be explored. An algorithm which must reach an optimal solution in a finite number of such exchanges will be developed.

A traffic light schedule shall be considered as "feasible" if it meets the constraints (a) compatibilities, (b) cycle time and (c) minimal green times of the problem statement. Since all the phases satisfy the compatibility matrix (a) two requirements for feasible schedules remain: (b) the sum of the phase durations must equal the cycle time

$$\sum_i d_i = T \tag{8}$$

and (c) for each light the minimal green time $f_{k\,\min}$ must be less than the sum over the durations d_i of those phases in which it shows green.

$$\sum_{i|k \in C_i} d_i \geqslant f_{k\,\min} \quad \text{for all } k \tag{9}$$

Feasible schedules satisfy a system of linear inequality constraints. The space of feasible solutions is therefore a polygonal convex set.

Whether the coordination achieved with a solution is satisfactory or not does not matter with respect to feasibility. Many solutions which are feasible in the sense introduced above will be unsatisfactory in practice, because of poor coordination. However, not every feasible schedule is intended for practical usage. A schedule that is used should be both "feasible" and "optimal".

The objective function (sum of weighted starting-point deviations) has the starting-point deviations as its variables and not the phase durations which appear in the constraint equations. The starting-point deviations are related to phase durations, offset and ideal starting points by a set of "auxiliary" equations. Each auxiliary equation produces one "evaluation variable" for the objective function. The formulation of the auxiliary equations is complicated by the necessity to account for the cyclic repetition of the ideal starting points.

If one discards the cyclic repetition for a moment and assumes that there is one actual and one ideal starting point for a light k, the actual starting point and the starting-point deviation can be found. The actual starting point $m_k\oplus$ is the sum of offset t_v and the durations d_i of those phases which occur before the phase C_{ek} in which the green interval for light k commences.

$$m_k\oplus = t_v + \sum_{i=1}^{ek-1} d_i \tag{10}$$

The starting-point deviation m_k results if the ideal starting point m_{0k} is subtracted from the actual starting point.

$$m_k = m_k\oplus = m_{0k} = t_v - m_{0k} + \sum_{i=1}^{ek-1} d_i \tag{11}$$

Consideration of the cyclic repetition of the schedule and the ideal starting points requires that the result of equation (11) be expressed modulo T. For the evaluation of the starting-point deviation with respect to the closest ideal starting point, it is further necessary to have $|m_k| \leqslant T/2$. The starting-point deviation will result correctly (magnitude and sign) if the cycle time is added or subtracted from (11) as necessary. One can write

$$m_k = t_v - m_{0k} + \sum_{1}^{ek-1} d_i + v_k T \tag{12}$$

where v_k is an unknown integral number. m_k will be positive if the ideal starting point occurs before the actual starting point. If the constraints

$$0 < |t_v| < T/2 \quad \text{and} \quad 0 < m_{0k} < T$$

are imposed it is sufficient to permit

$$v_k = (-1, 0, +1) \tag{13}$$

One could, in fact, write the relation such that two values of v_k would be sufficient, but it seems easier to consider the whole system of starting-point deviations of the problem simultaneously if the three values are permitted.

The contribution Δz of one starting-point deviation to the objective function is given by

$$\Delta z = |m_k| g_k \tag{14}$$

This can be expressed as the sum of two linear terms in non-negative variables. One can write

$$|m_k| = m_k' + m_k'' \tag{15}$$

where

$$m_k' = \begin{cases} m_k & m_k \geqslant 0 \\ 0 & m_k < 0 \end{cases}$$

$$m_k'' = \begin{cases} 0 & m_k \geqslant 0 \\ -m_k & m_k < 0 \end{cases}$$

(compare with Fig. 15). Equation (14) becomes

$$\Delta z = m_k' g_k + m_k'' g_k$$

and the objective function z for the whole problem assumes the form

$$z = \sum_k (m_k' g_k) + \sum_k (m_k'' g_k) \tag{16}$$

which is linear and has non-negative variables.

In Fig. 17a the coefficient tableau for an example is shown. It consists of the first five phases for problem 1—which were shown to give a phase sequence in Fig. 14—and minimal green times and starting points from Fig. 1. The first five columns of the tableau contain the phases. For convenience the phases have been renumbered; the new ordering corresponds to their sequence of occurrence in the schedule. To the right of the phases there are ten columns for the five starting-point deviations specified in Fig. 1 (five for positive and five for negative deviations). The last two columns on the left-hand side are for positive and negative values of offset.

The top row of the tableau contains the equation for the cycle time. It is followed by eight equations for minimal green times and five auxiliary equations for starting-point deviations. The effects of cyclic repetition have been indicated by ± 70 ($= T$) on the right-hand side in the auxiliary equations.

The ambiguity in the right-hand sides of the auxiliary equations is present because of the choices for the deviation between the actual and the desired starting point, which is evaluated. If one specifies arbitrarily a set of ideal starting points that shall be used for the evaluations, the ambiguity is eliminated. The problem is then a linear program. If by accident the right set of ideal starting points was selected, an optimal solution of the linear program would be an optimal schedule.

The evaluation variables in the auxiliary equations contain no restriction for the deviation from a tight constraint. Therefore, the auxiliary equations do not influence the feasibility of a solution of any such linear program, and the set of feasible solutions for all these linear programs (thirty-five different linear programs are possible for the example) must be the same. It is given by relations (8) and (9). Nevertheless, the solution space of any one of these linear programs has additional dimensions for the evaluation variables and additional extreme points. The additional extreme points are located at the zeros of the evaluation variables.

A contribution Δz to the objective function as function of a continuous variation in offset describes a triangular wave with cycle time T and peaks $(T/2)g_k$ and 0. This function is piecewise linear and not convex. One may therefore feel discouraged from attempting a linear programming-type approach, which requires a convex solution space. However, if the problem is formulated such that each linear segment of the triangular wave involves a different variable, the difficulty can be circumvented. For the evaluation with respect to three ideal starting points in three successive cycles, one must then have six evaluation variables (three for positive and three for negative deviations).

(3.b) *A linear program for the optimization*

Correct evaluation of a starting point is possible if three successive starting points $(m_{0k}-T_{0k})$, m and $(m_{0k}+T)$ are available for the evaluation. One needs three auxiliary equations instead of one and can then have completely specified right-hand sides. Since only one deviation for each traffic light will be evaluated, one must provide that the ideal starting point for the evaluation can be changed by adding or subtracting the cycle time from the evaluation variable.

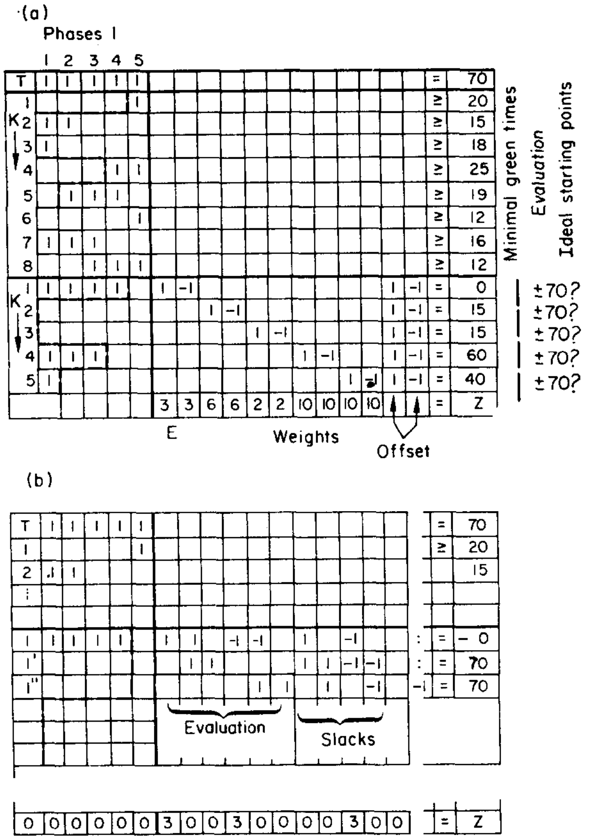


FIG. 17. Optimization for a phase sequence formulated as a linear program.

Figure 17b shows how the auxiliary equations for the evaluation of the starting point of light 1 must be written. The first equation has a positive and a negative evaluation variable, which has the weight in its column in the objective row. A second positive (negative) variable provides that the cycle time is added to (subtracted from) the evaluation variable unless it equals the slack in equation 1′(1″). The other equations for the starting point evaluations of the problem (Fig. 17) have the same form. The complete tableau contains fifteen auxiliary equations.

All the basic feasible solutions to the scheduling problem are extreme points of the convex solution space of this linear program. The converse is not true. The linear program also possesses extreme points which are not basic feasible solutions to the scheduling problem. The algorithm in (3d) will automatically exclude these.

(3.c) *Existence condition and construction for a feasible solution*

Whether a feasible solution exists depends on the blocking groups and the cycle time. In Section (2.c) it has been found that the lengths of the blocking groups yield a set of lower bounds on the cycle times for which feasible solutions might exist. Now a stronger claim will be made.

Theorem

For a unimodular phase sequence, a feasible solution exists if and only if no blocking group is longer than the required cycle time. (17)

The theorem postulates the existence of a "critical path" in the schedule. Before it is proven, an illustration of the claim will be given in Fig. 18. Figure 18a repeats the minimal green times (from Fig. 1). Figure 18b lists the blocking groups for the phase sequence in Fig. 17 and shows the evaluation of their length. In the illustration of the lengths of the blocking groups in Fig. 18c, the sequence of the elements in each blocking group has been chosen to correspond to the sequence of their (actual) starting points in the schedule. The first ten blocking groups are those shown in Fig. 10, the last blocking group (traffic streams 6 and 7) appears [see Section (2.d)] because phase 6 is not in the sequence. The highest lower bound on T is given by the group $(1, 3, 5)$ with 57 sec. Theorem (17) states that there must exist feasible solutions for all $T \geqslant 57$ sec. If the theorem is true, it must be possible to bring the green intervals in the blocking groups in Fig. 18c into such locations that relative to time each green interval has the same position in all blocking groups which contain it and such that there are no overlaps between incompatible elements of different blocking groups. For the minimal cycle time (57 sec in the example) the largest ("critical") blocking group has no slack and the positions of its elements are fixed. In all other blocking groups, there is some slack and repositioning of green intervals is possible. Figure 18d shows for the example a feasible arrangement of the green times. Figure 18e shows how a feasible schedule for problem 1 is obtained from Fig. 18d by filling in slack to meet the required cycle time of 70 sec.

(a) Minimal green times (From Fig. 1)

K	1	2	3	4	5	6	7	8	·/·
f	20	15	18	25	19	12	16	12	sec

Blocking groups (From Fig. 16)

a	1,2	20+15	35
b	1,3,5	20+18+19	57
c	1,7	20+16	36
d	2,4	15+25	40
c	2,6	15+12	27
f	2,8	15+12	27
g	3,4	18+25	43
h	3,5,6	18+19HZ	49
i	3,8	18+12	30
j	4,7	25+16	41
k	6,7	12+16	28

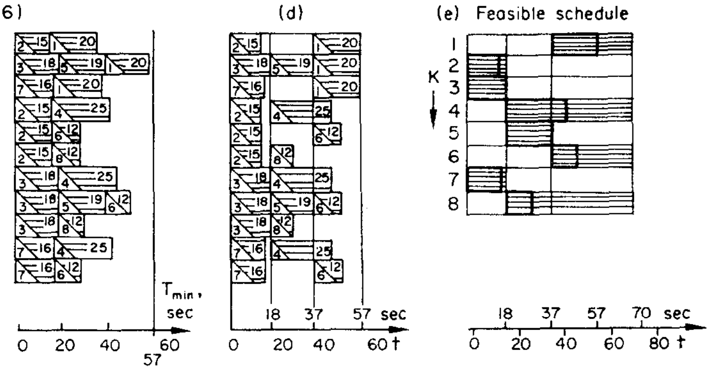


FIG. 18. Blocking groups—minimal cycle time—feasible schedule.

The proof for theorem (17) shall be given by means of a linear program which is related to the one in Fig. 17, but has a different objective: for the given phases and minimal green times, minimization of cycle time is requested. Figure 19 shows this program for the example. It should be noted that it is again a unimodular linear program since the matrix **A** appears on the left-hand side. In the proof the dual of this program is used. According to the duality theorem, the dual must have the same optimal value of its objective function *w* than the primal has for the objective *z*. The coefficient tableau for the dual is obtained from the primal if its whole tableau (including objective row and right-hand side) is transposed. The ⩾ of the restrictions in the primal is replaced by ⩽ in the restrictions of the dual.

d_i

		1	2	3	4	5			
	1					1	≥	20	
	2	1	1				≥	15	
	3	1					≥	18	
	4				1	1	≥	25	Primal
K	5		1	1	1		≥	19	
	6					1	≥	12	
	7	1	1	1			≥	16	
	8			1	1	1	≥	12	
		1	1	1	1	1	=	Z	Minimize !

K ⟶

		1	2	3	4	5	6	7	8			
	1		1	1				1		≤	1	
	2		1			1		1		≤	1	
d_i	3					1		1	1	≤	1	Dual
	4				1	1			1	≤	1	
	5	1			1		1		1	≤	1	
		20	15	18	25	19	12	16	12	=	W	

Maximize !

FIG. 19. Minimization of cycle time.

The dual problem has a row for each phase and a column for each traffic stream. If two traffic streams occur together in a row of the dual, they are compatible in G' [see Section (2.c)] if they do not occur together in some row, they are incompatible.

Feasible solutions of the dual must possess the following properties:

1. The variables have integral values. This must be the case because the coefficient matrix of the left-hand side is unimodular, and the right-hand side is integral.
2. The variables are non-negative (by virtue of the duality theorem).
3. The variables are ⩽ 1. This is required in the constraint equations.

It follows that the variables in feasible solutions of the dual can only assume the values 0 and 1. Further, each row can only contain one non-zero variable. Variables, which have the value 1 in a feasible solution must be pair wise incompatible in G'. The value of the objective function that goes with such a set is the sum of the minimal green times for the non-zero variables. Thus, the maximal value of the objective function must be the length of the longest blocking group of G'.

A simple construction of an initial basic feasible solution based on (17) is as follows:

> Beginning with phase 1, make each phase C_i as long as necessary to satisfy the green requirements for those lights in phase C_i which do not remain green in phase C_{i+1}. In doing so, treat the cycle time as a green requirement. (18)

If this leads to a schedule with

$$\sum d_i = T \tag{8}$$

a feasible solution has been obtained; if the result exhibits

$$\sum d_i > T$$

no feasible solution exists.

(3.d) *Optimization*

Optimization of the expanded linear program (example Fig. 17b) by means of the simplex algorithm is possible; however, due to the unimodularity of the problem, a more efficient algorithm is possible. To show this, use of the simplex algorithm shall be considered.

In the optimization process for the linear program (Fig. 17b) one can arbitrarily in the beginning reduce the evaluation of the deviations from the ideal starting points to one single ideal starting point. This is done by replacing the other non-zero variables by their corresponding slacks. Of course, there might be deviations $>T/2$ which now contribute to the objective z, if no attention is given to the choice of the one deviation that will be kept. Nevertheless, once there is only one ideal starting point that goes into the evaluation simplex need never go back to a basis which is not a schedule. All feasible solutions that follow in the optimization procedure can easily be sketched as traffic light schedules and give readily an interpretation of the objective function (as, for example, in Figs. 5b and 24). From such consideration of successive feasible solutions, it is then apparent that each transition must correspond to one of three possibilities:

1. Exchange of one ideal starting point that enters the evaluation of z (symbol for this change: W). The W-step is possible and improves the objective function if previously $m_k > T/2$ for some k.
2. Change in offset (step Zv). This shifts the whole schedule relative to reference time.
3. Exchange of time between two phases (step A). In this step, the duration of one phase is shortened in favor of some other phase which becomes longer. The beginnings of some phases shift relative to reference time.

Each of the three steps can easily be performed without reference to the simplex algorithm provided one knows: (a) which steps would improve the solutions and (b) the maximal feasible change width of the steps.

For the W-step, the answer to both (a) and (b) is simple: a W-step helps whenever $|m_k| > T/2$. The width of the step is one cycle time T. T has to be added or subtracted such that $|m_k|$ decreases. For convenience, the offset should also be handled in this fashion even though the objective is not affected by it.

For step A and step Zv the partial derivative of the objective function with respect to the intended change can be used to determine whether improvement would result. The width of the steps results from the requirement to keep the variables of the expanded program non-negative. Phase duration, offset, minimal green times and starting-point deviations have to be considered in this respect.

In the determination of the partial derivative of z with respect to a step, one can start out with the partial derivative of z for a change in the beginning of the individual phases and then sum over the derivatives of those phases that would be affected. The partial derivative of z with respect to a displacement in the beginning of one phase will in general be different for displacements towards $+t$ and towards $-t$. In either case it is obtained by adding (or subtracting) the weights of those starting points which occur in the phase. Let $\delta = (\delta_{ik})$ be a $(0, 1)$ matrix where $\delta_{ik} = 1$, if the starting point for light k is in phase i and p_i^+ and p_i^- the partial derivatives for displacement of the phase in the direction $+t$ respectively $-t$. One obtains

$$\begin{aligned} p_i^+ &= \sum_{k\mid m_k\geqslant 0} (\delta_{ik}g_k) - \sum_{k\mid m_k<0} (\delta_{ik}g_k) \\ \text{and} \\ p_i^- &= \sum_{k\mid m_k\leqslant 0} (\delta_{ik}g_k) - \sum_{k\mid m_k>0} (\delta_{ik}g_k) \end{aligned} \tag{19}$$

the slopes + and − for positive and negative changes in offset (step zv) follow to

$$\Delta^+ = \sum_i p_i^+ \quad \text{and} \quad \Delta^- = \sum_i p_i^- \tag{20}$$

For the exchange of green time between two phases (step A) formulas analogous to (19) will give the slope for z. However, phases, the beginnings of which do not move, have to be excluded from the summation. Furthermore, because of the cyclic repetition of the schedule the summation may run across the on–off point. Consideration of whether some phases will travel towards $+t$ (A_r) or towards $-t$ (A_l) and whether or not the beginning of the first phase (on–off) point will shift leads to four cases. These are summarized in Fig. 20. The indices α and β stand for the increasing and the decreasing phase respectively, r stands for right, l for left and Δ with indices is the derivative of z with respect to the step.

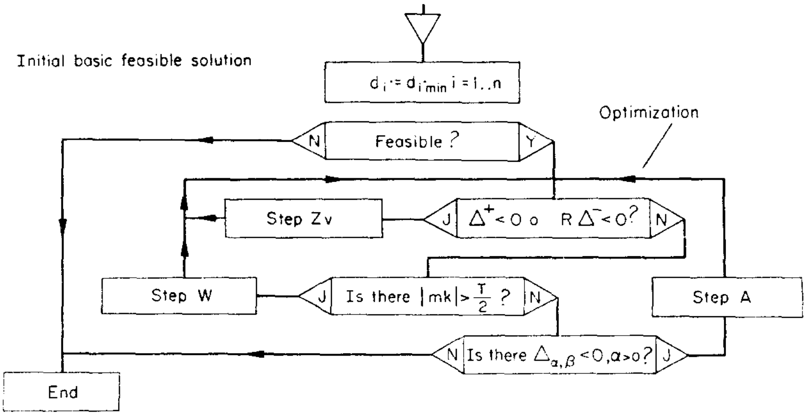


FIG. 20. Optimization of phase durations and offset.

The width of the steps Z_v and $A_r(A_l)$ is determined by the requirement that the excess of green time for the lights above the minimum, the phase durations, and the starting-point deviations of the expanded program must remain non-negative.

In a change of offset (step Z_v) only the offset itself and the starting-point deviations change. The step is limited by the first coincidence of an actual and ideal starting point that would occur if the transition between the two solutions is performed gradually. The width a of the step is therefore

$$a = \min(m_k'' \mid m_k'' > 0) \tag{21}$$

for shifts of the schedule towards $+t$ and

$$a = \min(m_k' \mid m_k' > 0)$$

for shifts towards $-t$.

In the exchange of time between two phases (step A_r or A_l) the duration of the losing phase C and the minimal green times have also to be watched for.

$$a = \min[(a_{k\alpha}\overline{a_{k\beta}})(f_k - f_{k\,\min}),\ d_\alpha,\ (\delta_{ik} m_k^*)]$$

with

$$a_{k\alpha}\overline{a_{k\beta}} = 1,$$

where

$$\overline{a_{k\beta}} = 0(1) \quad \text{if} \quad a_{k\beta} = 1(0)$$

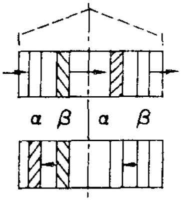
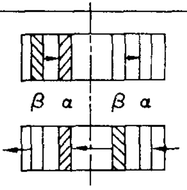
$$m_k^* = m_k'(m_k'') \quad \text{for shifts towards } -t\ (+t)$$

and i running over the phases which experience a translation of their beginning.

The term $(a_{k\alpha}\overline{a_{k\beta}})$ serves to bring only those green times into the expression which are present in phase C_α and not in phase C_β. With respect to the evaluation of $\delta_{ik} m_k^*$ there are again four cases (in analogy to the evaluation of the slopes). The detailed expressions are given in Fig. 20.

A step Zv or A which has a negative slope of the objective function, and a non-zero width of the step leads from a known solution to a better one. A sequence of such steps together with W-steps whenever possible must lead to an optimal schedule in a finite number of steps. The optimization is completed when there is (a) no $|m_k| > T/2$ and (b) no slope $\Delta_{\alpha\beta} < 0$ with width $a > 0$ for the corresponding step.

A flow chart for the scheduling algorithm is given in Fig. 20.

		I	m*	m**	t_v	On/off point
$\beta > \alpha$ Shift right	$\Delta^r_{\alpha\beta}$	$(\beta+1)..n$ and $1...\alpha$	m''	m'	a	
$\beta > \alpha$ Shift left	$\Delta^l_{\alpha\beta}$	$(\alpha+1)..p$	m'	m''	·/·	
$\beta < \alpha$ Shift right	$\Delta^r_{\alpha\beta}$	$(\beta+1)..\alpha$	m''	m'	·/·	
$\beta < \alpha$ Shift left	$\Delta^l_{\alpha\beta}$	$(\alpha+1)..n$ and $1...\beta$	m'	m''	−a	

General formulas:

$$\Delta_{\alpha,\beta} = \sum_{K \mid m^{**} \geqslant 0} \delta_{iK} \cdot g_K - \sum_{K \mid m^* > 0} \delta_{iK} \cdot g_K$$

$$a = \min\left((a_{K\alpha} \cdot \overline{a_{K\beta}})\,(f_K - f_{K_{\min}}),\ d_\alpha\, \delta_{iK} \cdot m_K^{**}\right) \quad \text{with} \quad \alpha_{K\alpha} \cdot \overline{\alpha_{K\beta}} = 1,\ \delta_{iK} = 1,\ i \in l$$

Notation: α = Decreasing phase β = Increasing phase t_v = Offset

$$\left.\begin{matrix} l \\ r \end{matrix}\right\} \text{ Some phases shift toward } \left\{\begin{matrix} \text{left}(-t) \\ \text{right}(+t) \end{matrix}\right.$$

$$i \in C_i \Leftrightarrow \text{Phases } C_i \text{ Shifts} \mid \overline{\alpha_{iK}} = \text{Complement of } \alpha_{iK}$$

FIG. 21. Cases for step A.

A possible tabular form for the necessary computations is shown in Figs. 22 and 23. Figure 22 repeats in the upper part the data arrangement of Fig. 17; however, the system has been transposed so that the right-hand side appears now below the "head" of the table.

	d_i						f_k															
g_k															3	6	2	10	10	-	-	-
i	1	2	3	4	5	T	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
															m_k							
1	1					1		1	1				1		1			1	1			
2		1				1		1			1		1		1			1				
3			1			1					1		1	1	1			1				
4				1		1				1	1			1	1							
5					1	1	1			1		1		1								

	t_v																			
A_0	0	0	0	0	0	0	70	20	15	18	25	19	12	16	12	0	15	15	60	40
A_1	0	-18	0	0	0	0	52	20	-3	0	25	19	12	-2	12	-18	15	15	42	22
A_2	0	-18	0	0	-19	0	33	20	-3	0	6	0	12	-2	-7	-37	-15	-15	42	22
A	0	-18	0	0	-19	33	0	-13	-3	0	-27	0	-21	-2	-40	-37	15	15	42	22
B	15	-18	0	0	-19	-33		-13	-3	0	-27	0	-21	-2	-40	-52	0	0	27	7
C	22	-18	0	0	-19	-33		-13	-3	0	-27	0	-21	-40	-40	-59	-7	-7	20	0
D	22	-18	0	-19	0	-33		-13	-3	0	-8	0	-21	-21	-40	-59	-7	-7	-1	0
E	22	-18	0	-20	0	-32		-12	-3	0	-7	-1	-20	-22	-40	-60	-7	-7	0	0
F	22	-18	0	-20	0	-32		-12	-3	0	-7	-1	-20	-22	-40	10	-7	-7	0	0
G	22	-18	0	-20	10	-22		-2	-3	0	-7	-11	-10	-22	-40	0	-7	-7	0	0
H	15	-25	0	-20	-10	-22		-2	-3	-7	0	-4	-10	-29	-33	7	0	0	0	0
J	15	-25	0	-20	-10	-20		0	-3	-7	0	-6	-8	-29	-33	5	0	0	0	0

					P_i^+						P_i^-					
	BEM		Z		1	2	3	4	5	Δ^+	1	2	3	4	5	Δ^-
A	ZV	8	7	1	-8	-10	0	-10	3	-25	8	10	0	10	-3	25
B	ZV	4	9	6	8	-10	0	-10	3	-9	8	10	0	10	-3	25
C	A_r	4	3	3	8	10	0	-10	3	11	-8	10	0	10	-3	9
D	A_r	2	4	3	8	10	0	-10	3	11	-8	10	0	10	-3	9
E	W	2	3	6	8	10	0	10	3	31	-8	10	0	10	-3	9
F	A_r		8	6	8	10	0	10	-3	25	-8	10	0	10	3	15
G	A_l		5	6	8	10	0	10	3	31	-8	10	0	10	3	15
H	A_r		2	1	8	10	0	0	-3	25	8	10	0	10	3	31
J			1	5	8	10	0	10	-3	25	8	10	0	10	3	31

FIG. 22. Optimization of a schedule for example 1.

Right shift, β

C,D	1	2	3	4	5
α 1		1	1	11	8
2	10		11	21	18
3	10	0		21	18
4	0	-10	-10		8
5	3	-7	-7	3	

Left shift, β

C,D,E	1	2	3	4	5
α 1		10	10	20	17
2	-1		0	10	7
3	-1	9		10	7
4	-11	-1	-1		-3
5	-8	2	2	12	

β

E,G	1	2	3	4	5
α 1		21	21	11	8
2	10		31	21	18
3	10	0		21	18
4	20	10	10		28
5	23	13	13	3	

β

H	1	2	3	4	5
α 1		10	10	20	23
2	21		0	10	13
3	21	31		10	13
4	11	21	21		3
5	8	18	18	28	

F,H Increasing, β

Decreasing, α	1	2	3	4	5
1		15	15	5	8
2	10		25	15	18
3	10	0		15	18
4	20	10	10		28
5	17	7	7	-3	

β

F,G	1	2	3	4	5
α 1		10	16	26	23
2	5		0	10	13
3	5	15		10	13
4	-5	5	5		3
5	-8	2	2	12	

FIG. 23. Gradient matrices.

In the upper left corner a unit matrix for the phase durations has been added to the head. At the extreme left below the head a column t_v is provided for the offset. The columns T for the cycle time and f_k for the green times of the traffic streams form the center part of the tableau. To the right of the green times are columns for the starting-point deviations.

As a basis for the computation of the various derivatives the $p_i\pm$ are computed in a separate table on the right-hand side of Fig. 25. From this table the slopes Δ^+, Δ^- and $\Delta_{\alpha\beta}^{(\cdot)}$ which are needed for the selection of the next step can be computed. For example, these data have been arranged to form square matrices ("gradient matrices") in Fig. 23.

In Fig. 22 there will be one row corresponding to each cycle of the simplex algorithm. The head—the left-hand side of the problem— is not changed in the computational process. It is used solely as a guide in finding the variables affected by the transition from one row to the next one. Row A contains the right-hand side of the problem posed in Fig. 17. The rows A_1, A_2, A show the construction of an initial basic feasible solution according to (18). In each row the element that determines the width of the next step is underlined. The next row is formed by subtracting the duration of the phase which is entering the basis from all elements that have 1 in the row of that phase in the head. Therefore phase durations show up with negative values and negative values in column f_k indicate the slack green time (green time in excess of the required minimal green). In the columns m_k the separation of positive and negative values into m_k' and m_k'' has been given up. This leads to a more compact arrangement, but necessitates of course some extra care in the choice of the sign for adding or subtracting in the m_k columns during the optimization.

With respect to the gradient matrices in Fig. 23 it should be noted that not all the gradients for both shifts to the right and to the left have been shown. Also some gradient matrices correspond to more than one row of the table.

The feasible schedules corresponding to rows A to I of Fig. 22 are sketched in Fig. 24.

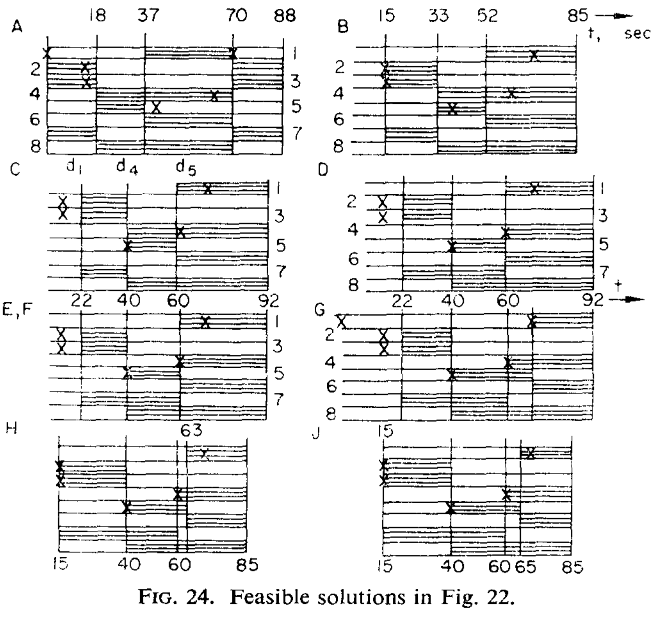


FIG. 24. Feasible solutions in Fig. 22.

Remarks

1. During the optimization there will often be more than one possible next step; the choice taken may then influence the number of steps necessary to reach an optimum. However, the true optimum of the objective function will be reached irrespective of the choices made. The sequence Z_v, W, A which is shown in the flow chart Fig. 21 is not mandatory.

2. One might suspect that the problem has local optima. For example, optimization might be carried out with respect to a fixed set of ideal starting points first. Thereafter an exchange in reference for some starting-point deviation must follow unless the global optimum is reached. One might doubt that such a W-step with a decrease in z will always be possible, if the global optimum was not yet reached. However, from the comparison with the expanded linear program, which has convex solution space, it is clear that local optima cannot exist, and therefore an improving w-step must be possible.

3. No discussion of "degeneracy" has been given. In principle any complication that might arise from this can be overcome by perturbation ot constants (see Dantzig, 1963, chapter 10).

4. EXAMPLES AND CONCLUSIONS

(4.a) *A conventional schedule for example* 1

Current practice is to adopt a phase sequence on an intuitive basis. Example 1 contains a strong left turning traffic stream 5 in Fig. 1 and an opposing traffic stream 1 which needs less green time than the confluent traffic stream 4. A "split-phase system" is usually recommended for such a situation. For example, the phase sequence shown in Fig. 5b has then to be used.

The selection of offset and the freedom in selecting phase durations that are left by the minimal green times must be used for coordination. If the priorities expressed by the weights and the figure of merit $z = \sum |m_k| g_k$ are adopted, the schedule shown in Fig. 5b is the best possible result. It yields $z = 442$.

The optimal schedule found with the methods shown in this paper was given in Fig. 27h. It has the considerably better value of $z = 15$.

Admittedly the criterion $z = \sum |m_k| g_k$ is a subjective criterion, especially since the weights are a matter of choice. Nevertheless the relative magnitudes 442 : 15 indicate that the schedule found in the paper is much superior with respect to coordination. This should be credited mainly to the unconventional phase sequence that has been used.

(4.b) *An example of progressive timing*

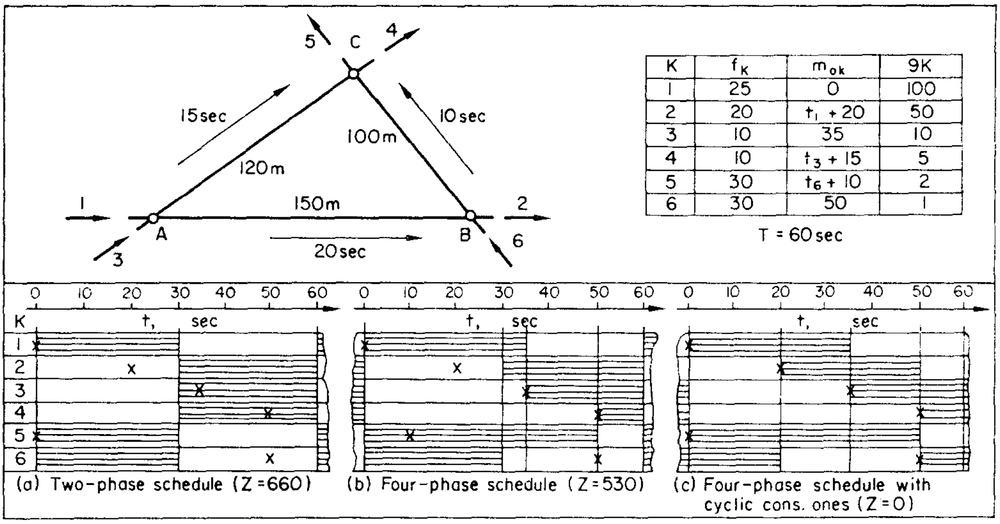
Figure 25 shows a network composed of three one-way streets with three conflict areas. The specification contains three ideal starting points for the entrance of traffic streams (1, 3 and 6) into the network and three specifications of travel time between successive conflict areas. The latter specifications mean "ideal differences" between starting points instead of ideal starting points specified relative to a fixed reference time. These ideal differences and the deviations therefrom can be expressed in terms of a summation over some phase durations. Example: in Fig. 25 the difference between the starting points for traffic streams 1 and 2 is specified as 20 sec. For the phase sequence in Fig. 26 this means a desired duration of 20 sec for phase 1. Similarly the duration of phase 3 is desired to be 15 sec from the ideal difference of 15 sec from the starting points of streams 3 and 4.

Three phase sequences will be considered for the network.

1. An ordinary two-phase schedule is given in Fig. 25a. In this schedule only two out of the six starting points have their ideal locations.

2. A schedule with four phases and consecutive ones is shown in Fig. 25b. This phase sequence has been optimized with the algorithm (as shown in Fig. 26). It should be noted that the weight associated with an ideal starting-point difference enters at both the starting points which determine the difference.

3. In Fig. 27 a phase sequence with cyclic consecutive one is considered. This example goes beyond the computational basis established in the paper. The optimization of this



K	f_K	m_{ok}	g_K
1	25	0	100
2	20	t_1+20	50
3	10	35	10
4	10	t_3+15	5
5	30	t_6+10	2
6	30	50	1

T = 60 sec

(a) Two-phase schedule (Z = 660) (b) Four-phase schedule (Z = 530) (c) Four-phase schedule with cyclic cons. ones (Z = 0)

FIG. 25. An example of progressive timing.

sequence is carried out in Fig. 27 with the second schedule as an initial feasible solution. The result is a schedule with ideal coordination (see Fig. 26c).

It is worth noting that all three schedules in Fig. 25 show the same total amount of green time (180 sec of green per cycle). This demonstrates by example that the traditional reasoning against a large number of phases is not valid in case of overlapping greens (and if the definition of phase given in this paper is used.)

						f_k						m_k							
	T	d_1	d_2	d_3	d_4	f	2	3	4	5	6	m_1	2	3	4	5	6		
i 1	1	1				1				1	1		1	1					
2	1		1			1	1			1				1					
3	1			1			1	1		1					1				
4	1				1		1	1	1										
											g_k	100	50	10	5	2	1		Z
	60					25	20	10	10	30	30	0	20	35	15	10	50		
	30	-30				-5	20	10	10	0	0	0	-10	5	15	10	50		
A	0	-30			-30	-5	-10	-20	-20	0	0	0	-10	5	15	10	50	W	
B	–	-30			-30	-5	-10	-20	-20	0	0	0	-10	5	15	10	-10	A_r	655
C	–	-30	-5		-25	-5	-10	-15	-15	-5	0	0	-10	5	0	10	-10	A_r	605
D	–	-30	-5	-15	-10	-10	-10	-15	0	-20	0	0	-10	0	0	10	-10		530

	p_1^+	p_2^+	p_3^+	p_4^+	Δ^+	p_1^-	p_2^-	B_3^-	p_4^-	Δ^-
B	51	50	-5	-5	91	149	-50	15	5	119
C	51	50	15	-5	111	149	50	15	5	119
D	51	50	15	5	121	149	50	15	5	119

	After B for A_r:				After C for A_r:				
	1	2	3	4	1	2	3	4	
1	·/·	41	46	51	·/·	41	46	51	1
2	50	·/·	94	101	50	·/·	96	101	2
3	45	-5	·/·	96	65	15	·/·	116	3
4	40	-10	5	·/·	55	110	-5	·/·	4

Offset omitted since = 0   Starting point differences enter twice

FIG. 26. Example 3: phase sequence with consecutive ones.

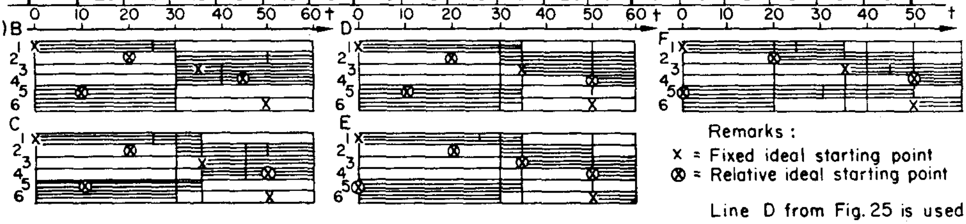
(4.c) *Critical remarks*

1. *Optimality.* The problem as stated at the outset requires that traffic streams which obtain green simultaneously must be pairwise compatible. It does not require the sets of traffic streams which get green simultaneously to be maximal sets of pairwise compatible

streams. As long as no restrictions for the use of phases in a phase sequence are introduced, one can without loss of generality assume that only such maximal sets of pairwise compatible traffic streams will be used as phases. However, as soon as the consecutive-ones requirement for the phases is imposed, this need no longer be true. Therefore the best solution found in the way described need not be the best possible one and optimality cannot be claimed.

i ↓	T	d_1	2	3	4	5	6	g_1	2	3	4	5	6	m_1	2	3	4	5	6
1	1	1						1				1	1		1	1		1	
2	1		1					1	1			1				1		1	
3	1			1					1	1		1					1	1	
4	1				1				1	1	1							1	
5	1					1				1	1		1						1
6	1						1				1		1					1	1

D	–	−30	−5	−15	−10	0	0	−10	−10	−15	0	−20	0	0	−10	0	0	10	−10
E	–	−30	−5	−15	0	−10	0	−10	0	−15	0	−20	−10	0	−10	0	0	0	0
F	–	−20	−15	−15	0	−10	0	−10	−10	−15	0	−20	0	0	0	0	0	0	0



Remarks :
X = Fixed ideal starting point
⊗ = Relative ideal starting point

Line D from Fig. 25 is used as a starting solution

FIG. 27. Example 3: phase sequence with cyclic consecutive ones.

On the other hand, if the optimal solution for all phase sequences is found, which have consecutive ones and maximal sets of traffic streams in the phases, usually an approximation to the optimum which is better than what can be expected from currently used scheduling techniques will result. Therefore near optimality is claimed for the approach.

To be able to guarantee optimality rigorously one must permit all sets of mutually compatible streams, not only the maximal such sets, as phases and then consider all maximal phase sequences for this larger family of phases.

2. *Effect of on–off point upon coordination.* In Fig. 25b—which has consecutive ones but not cyclic consecutive ones—a starting-point deviation remains in the optimal solution. On the other hand, the solution with cyclic consecutive ones is ideal. Practical experience speaks for the assumption that the finding in Fig. 25 is typical rather than accidental. It seems desirable not to have a condition of the form $C_a \cap C_n = \phi$ for the phase sequence when coordination is important.

3. *Extension to larger networks.* The example of a network in Fig. 25 has shown that the method of attack developed in this paper can in principle be used for networks. However, from the point of view of computational efficiency, the approach is not very satisfactory. If one sets up phases and phase sequences not for individual intersections but for networks containing several separate intersections, the number of phases and phase sequences increases sharply. Even with high-speed computers the method seems economically unfeasible for large networks.

Acknowledgements—A first study of the relationship between compatibilities and phases was made by the author in 1963 while working in the laboratory for traffic theory of Siemens and Halske in Munich, Germany. The algorithm for the determination of the dominant cliques given in Section (2.b) was developed at that time. Most of the material presented was developed by the author for his doctoral thesis at the University Fredericiana in Karlsruhe, Germany. Encouragement and criticism from Professors Leutzbach and Knodel are gratefully acknowledged. Discussions with them influenced the final form of the study.

The author is indebted to Dr. T. C. Hu for his referral to the paper on interval graphs by Fulkerson and Gross. The State College in Sacramento liberally supplied computer time.

REFERENCES

BERGE C. (1966). *The Theory of Graphs and Its Applications.* Methuen, London.

BUSACKER R. G. and SAATY T. L. (1965). *Finite Graphs and Networks.* McGraw-Hill, New York.

DANTZIG G. B. (1963). *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey.

ENGEL E. (1961). *Grundlagen der Strassenverkehrstechnik.* Goeschen series, Vol. 1198.

FEUCHTINGER M. E. (1954). Die Berechnung signalgesteuerter Knotenpunkte des Strassenverkehrs Forschungsarbeiten aus dem Strassenwesen. Neue Folge Heft 12. Kirschbaum, Bad Godesberg (formerly Bielefeld).

FORSCHUNGSGESELLSCHAFT FÜR DAS STRASSENWESEN E. V. (1966). Richtlinien für den Bau, Entwurf und Betrieb von Lichtsignalanlagen. Cologne.

FULKERSON D. R. and GROSS O. A. (1964). Incidence matrices and interval graphs. The Rand Corporation, Memorandum RM-3994-PR.

GILMORE P. C. and HOFFMAN A. J. (1964). A characterization of comparability graphs and of interval graphs. *Can. J. Math.* **16**, 539.

GLÜCK K. and SCHELZKE R. (1960). Startzeit und Zeitlücken an Signalgesteuerten Knoten. *Strassenverkehrstechnick* **4** (8), 357.

GREENSHIELDS B. D., SHAPIRO D. and ERICKSEN E. L. (1947). Traffic performance at urban intersections. Technical Report No. 1. Yale Bureau of Highway Traffic, New Haven, Conn.

HAIGHT F. A. (1963). *Mathematical Theories of Traffic Flow.* Academic Press, New York.

HOFFMAN A. J. and KRUSKAL J. B. (1956). *Integral Boundary Points of Convex Polyhedra.* p. 223. Princeton University Press, Princeton, N.J.

KIRCHGÄSSNER K. (1965). Die graphentheoretische Lösung eines nichtlinearen Zuteilungsproblems. *Unternehmensforschung* **9**, 216. (German O.R.).

KORTE J. W. (1960). *Grundlagen der Strassenverkehrstechnik in Stadt und Land,* 2nd edn. Bauverlag, Wiesbaden.

KNÖDEL W. (1960). Verkehrsplanung und Mathematik III. *Netzmodelle Mathematik, Technik Wirtschaft (MTW)* 7 (4), 147.

KÜNZI H. P. and KRELLE W. (1962). *Nichtlineare Programmierung.* Springer, Berlin.

LEKKERKERKER C. G. and BOLAND J. C. (1962). Representation of a finite graph by a set of intervals on the real line. *Fundam. Math.* **51**, 45–64.

LEUTZBACH W. (1966). Probleme der Kolonnenfahrt. *StrBau und StrVerk.* No. 44.

MAGHOUT M. K. (1959). Sur la détermination des nombres de stabilité et du nombre chromatique d'un graphe. *C.R.hebd. Séanc. Acad. Sci., Paris* **248**, 3522.

MORGAN J. T. and LITTLE J. O. C. (1964). Synchronizing traffic signals for maximal bandwidth. *Ops. Res.* **12**, 896.

VON STEIN W. (1961). Traffic flow with presignals and the signal funnel, in *Theory of Traffic Flow.* Elsevier, New York.

WARDROP J. G. and CHARLESWORTH G. (1952). A method of estimating speed and flow of traffic from a moving vehicle. Paper 5925, Institute of Civil Engineers, London.

WEBSTER F. V. (1958). Traffic signal settings. Technical Paper No. 39, Road Research Laboratory, England.