

UMAP

**Modules in
Undergraduate
Mathematics
and Its
Applications**

**Published in
cooperation with**

**The Society for
Industrial and
Applied Mathematics,**

**The Mathematical
Association of America,**

**The National Council
of Teachers of
Mathematics,**

**The American
Mathematical
Association of
Two-Year Colleges,**

**The Institute for
Operations Research
and the Management
Sciences, and**

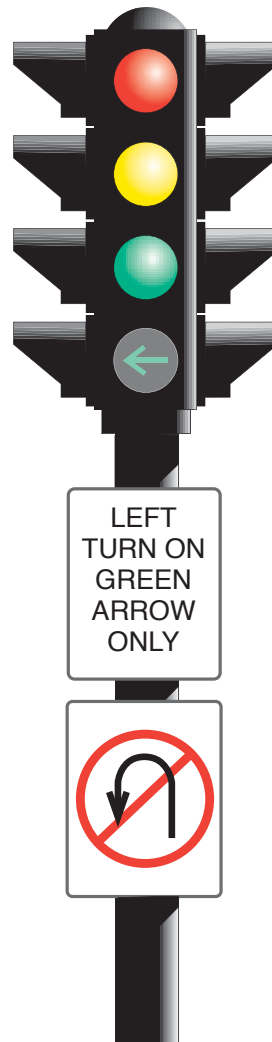
**The American
Statistical Association.**

The logo for COMAP, featuring the letters 'COMAP' in a stylized, cursive, black font.

Module 793

The Scheduling of Traffic Lights

Sara Kuplinsky
and
Julio Kuplinsky



Graph Theory, Optimization

INTERMODULAR DESCRIPTION SHEET:	UMAP Unit 793
TITLE:	The Scheduling of Traffic Lights
AUTHOR:	Sara Kuplinsky School of Theoretical and Applied Science Ramapo College of New Jersey Mahwah, NJ 07430 skuplins@ramapo.edu Julio Kuplinsky Dept. of Mathematics Yeshiva University New York, NY 10033 kuplinsk@yu.edu
MATHEMATICAL FIELD:	Graph theory, Optimization
APPLICATION FIELD:	Traffic management
TARGET AUDIENCE:	The first 8 sections are appropriate for Liberal Arts majors. A modeling course for mathematics majors could use all 12 sections.
ABSTRACT:	We consider the problem of scheduling traffic lights optimally, or more generally, the problem of allocating optimally the use of a facility to users having conflicting constraints. To this effect, we show how traffic at an intersection can be modeled by a graph with some additional structure and how to set up a linear program from which a solution to the problem can be obtained. An algorithm to carry out this task is discussed in detail.
PREREQUISITES:	Elementary graph theory, linear programming.

UMAP/ILAP Modules 2006: Tools for Teaching, 43–85.
©Copyright 2006 by COMAP, Inc. All rights reserved.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice. Abstracting with credit is permitted, but copyrights for components of this work owned by others than COMAP must be honored. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior permission from COMAP.

COMAP, Inc., Suite 3B, 175 Middlesex Tpke., Bedford, MA 01730
(800) 77-COMAP = (800) 772-6627, or (781) 862-7878; <http://www.comap.com>

The Scheduling of Traffic Lights

Sara Kuplinsky
School of Theoretical and Applied Science
Ramapo College of New Jersey
Mahwah, NJ 07430
skuplins@ramapo.edu

Julio Kuplinsky
Dept. of Mathematics
Yeshiva University
New York, NY 10033
kuplinsk@yu.edu

Table of Contents

1. FOREWORD	1
2. INTRODUCTION	1
3. MODELING THE PROBLEM	3
4. IMPROVING THE ASSIGNMENT	5
5. MAXIMAL CLIQUES	5
6. LINEAR PROGRAMMING	7
7. PHASINGS AND INTERSECTION ASSIGNMENTS	9
8. A MORE INVOLVED EXAMPLE	9
9. CONSECUTIVE ORDERINGS	12
10. FULL INTERSECTION ASSIGNMENTS	15
11. AN ALGORITHM	19
11.1 The Algorithm	19
11.2 Notes on the Algorithm	21
12. COMMENTS AND SUGGESTIONS FOR FURTHER READING	24
INDEX	26

13. SOLUTIONS TO THE EXERCISES	27
REFERENCES	38
ABOUT THE AUTHORS	39

MODULES AND MONOGRAPHS IN UNDERGRADUATE
MATHEMATICS AND ITS APPLICATIONS (UMAP) PROJECT

The goal of UMAP is to develop, through a community of users and developers, a system of instructional modules in undergraduate mathematics and its applications, to be used to supplement existing courses and from which complete courses may eventually be built.

The Project was guided by a National Advisory Board of mathematicians, scientists, and educators. UMAP was funded by a grant from the National Science Foundation and now is supported by the Consortium for Mathematics and Its Applications (COMAP), Inc., a nonprofit corporation engaged in research and development in mathematics education.

Paul J. Campbell
Solomon Garfunkel

Editor
Executive Director, COMAP

1. Foreword

We present a detailed discussion of a method for scheduling traffic lights. This choice should be understood as only one example of a number of problems that can be solved using the same techniques as those employed here.

To give a general description, these are problems in which a facility has to be shared by several users. Some pairs of users are allowed to use the facility simultaneously while others are not. Moreover, there are other restrictions, typically time related. The problem consists of assigning a time slot to each user, taking the restrictions into account, in such a way that a quantity, also involving time, is optimized.

Developing an early treatment due to Stoffers [1968], Opsut and Roberts [1979, 1981, 1983a, 1983b] obtained the main results of the theory that we present here. While basing this expository paper on their research, we try to make their results more accessible to the general student by making it self-contained and by stating formally some facts that they do not mention explicitly. We include a substantial number of exercises and provide answers. Further, we emphasize the applications and the extent to which the model captures some aspects of the real-life situation while intentionally ignoring others.

The tools required are elementary graph theory and linear programming. The first eight sections, which could be taught in two 75-minute lectures of a general education mathematics course, discuss the basic ideas and present the methods to solve simpler problems. Once the model is set up, software can be used to solve the associated linear program.

Sections **9. Consecutive Orderings** and **10. Full Intersection Assignments** spell out definitions and results needed to present the algorithm described in section **11. An Algorithm**. This algorithm explains how to obtain schedules, gives conditions under which schedules are best possible, and specifies those instances where we only know how to obtain a partial solution.

Proofs of all results mentioned in this work will be published separately. The authors would be most obliged to interested readers who send comments and suggestions.

2. Introduction

An intersection of traffic lanes is an example of a facility shared by several users. To control the traffic, we want to schedule a traffic-light system that assigns to each user a period during which that user has the right of way, i.e., may use the facility.

Consider the intersection of **Figure 1**. Some pairs of traffic streams, like y and z , cannot go through the intersection at the same time; we say that they are *incompatible*. Moreover, let us assume that traffic engineers have determined that y and w are also incompatible, due to the fact that the east-west lane is too narrow to handle the east-west and left-turning traffic simultaneously.

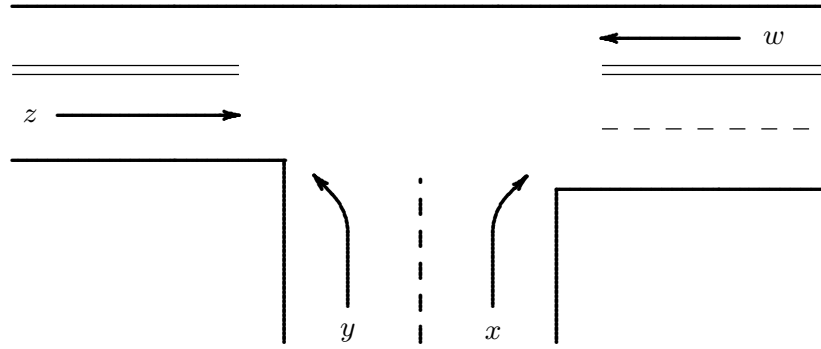


Figure 1. A simple intersection.

To model the compatibility and incompatibility relations, we use a graph whose vertices represent the streams and whose edges represent compatibilities between them. For the given intersection, graph G of **Figure 2** represents the streams and the presence or absence of conflicts between them.

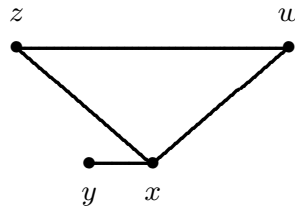


Figure 2. Graph to represent the intersection of **Figure 1**.

Each stream v is controlled by a traffic light with a green-light period that we denote by S_v .

The assignment of green-light periods should be such that incompatible streams get disjoint periods, that is,

$$u \neq v, \{u, v\} \notin E \implies S_u \cap S_v = \emptyset,$$

where E denotes the set of edges of G .

Of course, in a real situation, a stream like x , compatible with all others, would not need any signal. But, to illustrate the main ideas, we do not omit x from our arguments.

Traffic lights are scheduled cyclically: A pattern of light assignments repeats indefinitely. A *phase* is a time period delimited by light changes but during which there are no light changes on any stream. This concept is clarified by **Figure 3**, which depicts one possible assignment of red/green lights for the intersection of **Figure 1**. In **Figure 3** and in the sequel, we indicate green light using a partially shaded band and red light using a blank band.

The assignment has five phases: the first, lasting 2 units of time, and then four more, lasting 3, 18, 2, and 15 units. It is a rather inefficient assignment, one reason being that y should not have to wait 2 units to go through. A better assignment (with four phases) would be that in **Figure 4**.

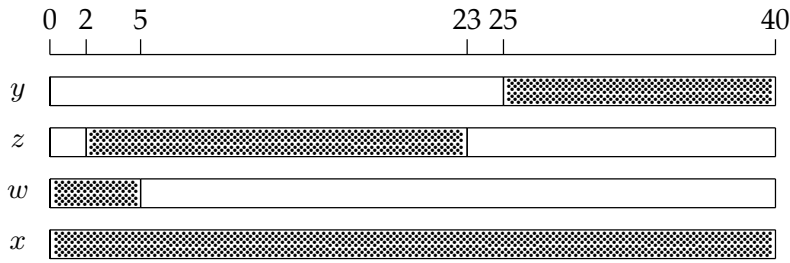


Figure 3. A possible assignment of red/green lights for the intersection of Figure 1.

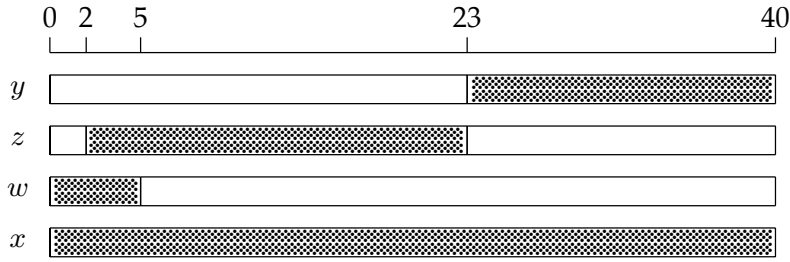


Figure 4. A better assignment of red/green lights for the intersection of Figure 1.

The four-phase cycle depicted in Figure 4 repeats indefinitely. We say that the *cycle length* equals $2 + 3 + 18 + 17 = 40$ units of time.

3. Modeling the Problem

We now point out conditions that arise naturally and yield a mathematical model whose solution determines a suitable schedule.

A yellow light is used to announce signal changes. Stoffers [1968, 201] points out that the length of these transitory states is in many countries fixed by regulations. This length may depend on the size or geometry of the crossing.

How should we incorporate yellow lights into our model? A yellow light at the end of a green period is an indication that the traffic should clear the intersection. Vehicles on a yellow light that are in or near the intersection are allowed to circulate, while streams incompatible with them still have a red light. However, we can consider that streams have only red or green lights: We translate a solution into the real world by changing to yellow an appropriate interval at the end of each green period. As a consequence, a scheduling like the one depicted in Figure 4 for the intersection of Figure 1 would look different on the road after yellow periods are added. For example, if we used solid shading for the yellow periods, we might have the result in Figure 5.

From now on, our models disregard yellow lights, so our diagrams will look like those in Figures 3 and 4.

The time that it takes a stopped vehicle to enter the intersection—which depends, among other things, on the position of the vehicle in the waiting line and the number of vehicles arriving at the intersection per unit of time—is im-

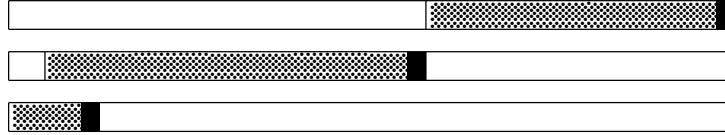


Figure 5. The scheduling of red/green lights from **Figure 4** with yellow periods added (solid shading).

portant for determining a minimum length of green for each stream. Moreover, since yellow periods (which are considered part of green periods) are “safety gaps” and thus have a minimum length, the green periods also have a minimum length. Traffic engineers determine these bounds by taking measurements at intersections. Van Vuren and Van Vliet [1992, 751], for example, give 2.5 s as a rough estimate for a starting delay, i.e., the time that it takes for a vehicle to start after the car in front has started. Another way of describing this delay is by measuring the speed of the motion wave. The authors give a value of 5 m/s for the speed of the wave moving back from the stop line.

So our model will require a given minimum amount of green time per stream per cycle. However, if we allow more than one green interval per cycle, we could obtain an assignment with at least the minimum amount of green but with some green intervals not long enough to contain yellow safety times. Another complication of multiple greens is noted by Stoffers [1968, 205]. At the beginning of each green period, “lost times” occur, until the traffic picks up speed. The total amount of green per stream then would be affected by an unknown number of these lead times. Hence, we prescribe just one green interval per cycle per stream.

Next, some reflections on cycle length. Long cycles, allowing for longer intervals of green, can accommodate more vehicles; but cycles longer than necessary to handle the traffic on hand may produce unacceptable delays. Van Vuren and Van Vliet suggest that it is best to use the shortest practical cycle length necessary to accommodate the traffic volume [1992].

Summarizing, we have

- a graph that indicates which pairs of streams are compatible and which are not; and
- for each stream v ,
 - a positive number r_v , the minimum length of an interval of green for that stream; and
 - another positive number N , the maximum length of the cycle (of course, N should be at least as large as the largest r_v .)

As seen in **Figures 3** and **4**, there can be many light assignments that satisfy the compatibility requirements and the restrictions imposed by r_v and N . Some assignments, however, are better than others: We ask that the total amount of

green (i.e., the sum of green times for all streams, for any one cycle) be as large as possible. We now return to our example to seek such improvement.

4. Improving the Assignment

Consider the graph of **Figure 2** on p. 2 and assume that the minimum amounts of green are $r_y = 15$, $r_z = 20$, $r_w = 5$, and $r_x = 20$ units of time, while the maximum cycle length is given by $N = 40$ units. In the assignment in **Figure 4**, the total amount of green is $17 + 21 + 5 + 40 = 83$.

Since we are interested in maximizing the total amount of green, that apportionment is clearly not optimal, because z and w could receive more green. The increase of green times is reasonable: Since x , w , and z are mutually compatible streams, they can share more green time without compromising cycle length, thus providing a longer total amount of green (**Figure 6**). The total amount of green is now 103 units.

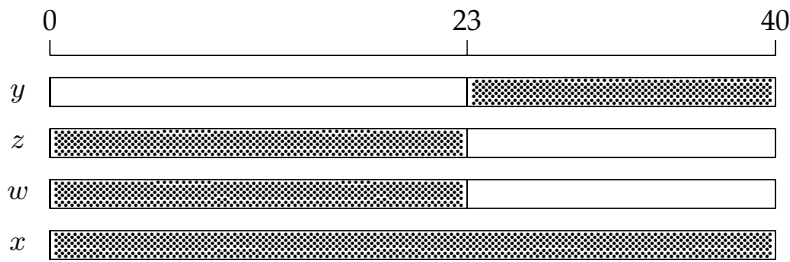


Figure 6. Sharing green time between traffic streams.

5. Maximal Cliques

The set of mutually compatible streams corresponds to a set of mutually adjacent vertices in our graph. We define a concept that is crucial in what follows:

Definition. A *maximal clique* of a graph G is a set K of mutually adjacent vertices of G not contained in any larger set with this property. In other words:

- any two vertices of K are adjacent, and
- a vertex not in K (if there are any) cannot be adjacent to every vertex in K .

For example, the graph in **Figure 2** has two maximal cliques: $K_1 = \{x, z, w\}$ and $K_2 = \{x, y\}$.

Exercise

1. Prove that every clique in a graph is contained in some maximal clique.

The sharing of green time by members of a maximal clique may increase the total amount of green. To discuss this in more detail, we assign to each maximal clique K_i a length of green time d_i , called the *duration* of the clique. It represents the length of the period during which all members of K_i receive green light simultaneously. Each vertex receives d_i units of green time for each maximal clique K_i to which it belongs. Hence, vertices in K_i belonging to other maximal cliques in general receive more green than elements that belong to K_i only.

The assignments in both **Figures 7a** and **7b**, for example, show how x benefits from being a member of two maximal cliques. The vertices z , w , and y each belong to only one of the cliques and each receives less green than x does.

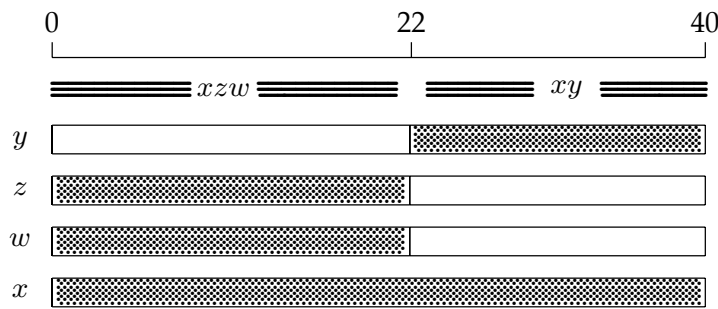


Figure 7a. Vertex x belongs to both the maximal cliques $K_1 = \{x, z, w\}$ (22 units of green) and $K_2 = \{x, y\}$ (18 units of green), hence receives 40 units of green.

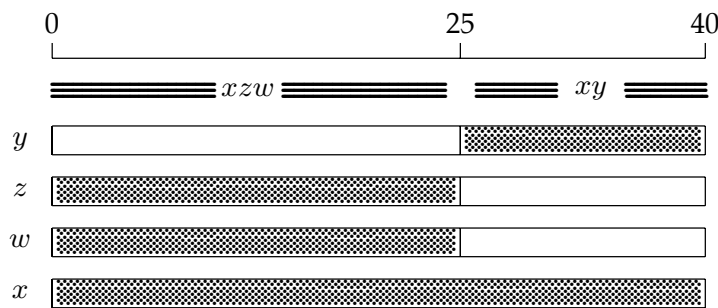


Figure 7b. The same maximal cliques as in **Figure 7a**, but with K_1 getting 25 units of green and K_2 getting 15 units of green; vertex x still get 40 units.

In these figures, we indicate graphically the members and durations of the maximal cliques. In **Figure 7a**, for example, clique $K_1 = \{x, z, w\}$ gets $d_1 = 22$ units and clique $K_2 = \{x, y\}$ gets $d_2 = 18$ units. The cliques are the same in **Figure 7b**, where $d_1 = 25$ units and $d_2 = 15$ units. The total amounts of green are respectively $18 + 22 + 22 + 40 = 102$ units and $15 + 25 + 25 + 40 = 105$ units.

Our model will have as unknowns the durations. We will specify an explicit method to obtain an assignment of green-light intervals that respects incompatibilities and satisfies the constraints of minimum green time per stream and

maximum cycle length. However, since some intersections don't have an assignment satisfying all the restrictions, we give conditions that guarantee the existence of such an assignment.

Let J_i be the interval during which all members of the maximal clique K_i receive green simultaneously. For convenience, we choose the phases as left-closed, right-open intervals. Referring again to the graph of **Figure 2**, we set $J_1 = [0, d_1)$. Then J_2 has to be such that $J_1 \cap J_2 = \emptyset$. Since we have an upper bound on cycle length but want to maximize the total amount of green, we must take $J_2 = [d_1, d_1 + d_2)$. So, for the two-phase assignments in **Figures 7ab**, we have $J_1 = [0, 22)$, $J_2 = [22, 40)$ and $J_1 = [0, 25)$, $J_2 = [25, 40)$ respectively. Recalling that x can receive green simultaneously with w , z , and y , we next define the interval of green that each vertex gets as follows:

$$S_y = J_2, \quad S_z = J_1, \quad S_w = J_1, \quad S_x = J_1 \cup J_2.$$

We express the restrictions of minimum green length and maximum cycle length in terms of the previously introduced unknowns d_i . Since the lengths of S_y , S_z , S_w , and S_x are d_2 , d_1 , d_1 , and $d_1 + d_2$, recalling the values of r_y , r_z , r_w , r_x , we conclude that the restrictions on minimum green length are

$$\begin{aligned} d_2 &\geq 15 \\ d_1 &\geq 20 \\ d_1 &\geq 5 \\ d_1 + d_2 &\geq 20. \end{aligned}$$

The cycle length is the length of $S_y \cup S_z \cup S_w \cup S_x$, so considering the value of N , the restriction on cycle length reads

$$d_1 + d_2 \leq 40,$$

and obviously

$$d_i \geq 0.$$

The optimality condition is: maximize the total amount of green, which is $d_2 + d_1 + d_1 + (d_1 + d_2)$, that is,

$$\text{maximize } 3d_1 + 2d_2.$$

6. Linear Programming

Our model is an optimization problem: We want to find the maximum (or the minimum) of a certain function, the variables of which must meet certain restrictions. In our example, the function to maximize is $3d_1 + 2d_2$, and the restrictions on the variables are the seven inequalities above. Our model in fact is a *linear program (LP)*, because the restrictions (or constraints) and the function

to maximize or minimize are linear. The restrictions can be of different types—some can use \leq and others \geq or even $=$ —but strict inequalities ($<$ or $>$) are *not* allowed. It is usually the case (but not mandatory) that all variables be required to be nonnegative.

For students who know how to find the optimal solution of such a problem graphically, **Figure 8** will show that the optimal solution is $d_1 = 25$, $d_2 = 15$.

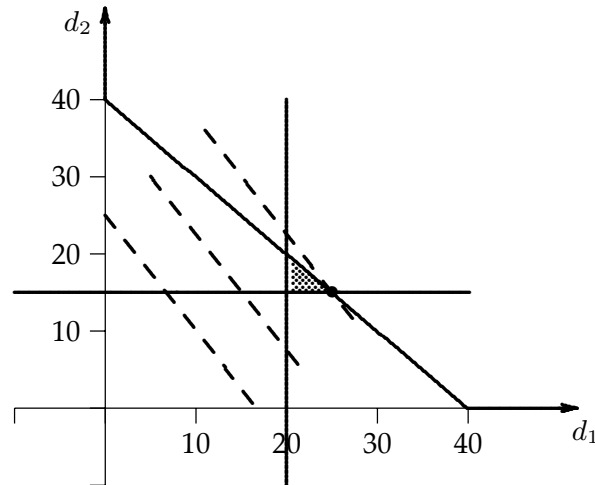


Figure 8. Graphical representation of a sample linear program and its solution.

If you are not familiar with the methods used to solve linear programs, you can rely on widely available software packages. We will use Student Lindo / PC Release 6.00 for our examples. You should enter the following input.

```
max 3d1 + 2d2 st
d2 > 15
d1 > 20
d1 > 5
d1 + d2 > 20
d1 + d2 < 40
```

Note that:

- Spaces are largely irrelevant, in particular between constants and variable names.
- For typing convenience, you enter $<$ or $>$, but the meaning is \leq or \geq .
- Variables are assumed nonnegative by default.
- Clicking the `Solve` button gives the solution.

That optimal solution gives the intervals $J_1 = [0, 25)$ and $J_2 = [25, 40)$, and therefore $S_y = [25, 40)$, $S_z = S_w = [0, 25)$, and $S_x = [0, 40)$. The total amount of green is 105 units, and the cycle length is 40 units. This optimal solution yields the schedule depicted in **Figure 7b**.

Exercises

2. Reverse the order of the cliques K_1 and K_2 and solve the previous problem again. Compare the solutions obtained.
3. Name local factors (i.e., factors pertaining to a particular intersection) that may contribute to determine the minimum amount of green each stream should receive.
4. Name quantities pertaining to a traffic intersection, different from the total amount of green, that it would be meaningful to maximize or minimize.

7. Phasings and Intersection Assignments

We sought assignments S such that $S_v \cap S_w = \emptyset$ whenever v and w are distinct nonadjacent vertices, but we got more than that. In fact, our optimal solution satisfies the converse; that is, if two vertices are adjacent, then the corresponding intervals intersect. To put it in formal terms, we distinguish the following conditions:

$$v \neq w, \{v, w\} \notin E \implies S_v \cap S_w = \emptyset, \quad (1)$$

and

$$\{v, w\} \in E \implies S_v \cap S_w \neq \emptyset. \quad (2)$$

Interval assignments that satisfy **(1)** are called *phasings*, and those that satisfy both **(1)** and **(2)** are called *intersection assignments*.

Now is a good time to recapitulate what we have done so far and to take a look at the road ahead. We have a model of a traffic intersection, which consists of a graph and some numerical data. The problem that we set out to solve is how to obtain either a phasing or an intersection assignment that satisfies the conditions of the model and is also optimal in providing maximum green time. We used a linear program with the data and the structure of the graph to obtain a phasing or intersection assignment. However, it is not at all clear that we could not obtain an even better scheduling by using other techniques, perhaps unrelated to linear programming, and independent of the maximal cliques.

Another question arises naturally in this context. Since the condition for phasing is less restrictive than the one for intersection assignment, it is conceivable that we can do better with phasings than with intersection assignments. In other words, maybe we could find a phasing with a larger total green time than the intersection assignment just obtained. We address these fundamental issues later.

In the meantime, we consider a second, more complicated intersection.

8. A More Involved Example

In **Figure 9**, stream p represents pedestrians, which, from the point of view of our mathematical model, can be treated like any vehicle stream.

Figure 10 depicts compatibilities and incompatibilities of the streams.

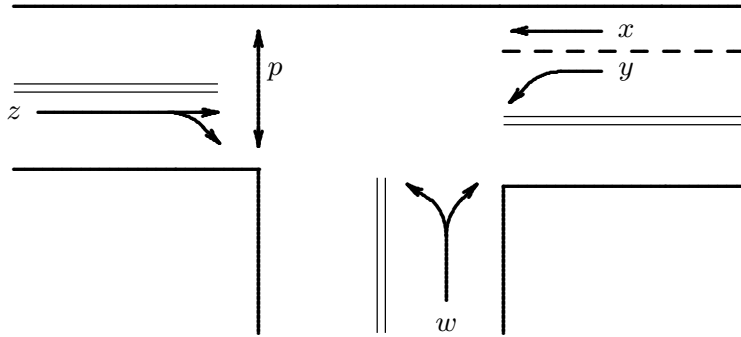


Figure 9. Intersection with a pedestrian stream p .

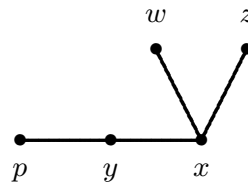


Figure 10. Graph corresponding to the intersection of **Figure 9**.

The graph in the earlier **Figure 2** on p. 2 had two maximal cliques, $\{x, z, w\}$ and $\{x, y\}$. We assigned an interval of green to $\{x, z, w\}$ to begin the cycle, and then another abutting interval, disjoint from the first one, to $\{x, y\}$. We also asked you in **Exercise 2** to reschedule the intersection, reversing the order in which the cliques are numbered. You must have noticed then that this too resulted in an admissible assignment, with each stream receiving one green interval per cycle.

The new example differs from the earlier one in a fundamental way: The ordering of the cliques matters.

Assume that we assign green intervals to the maximal cliques in the order $\{x, z\}$, $\{p, y\}$, $\{x, y\}$, and $\{x, w\}$. The assignment in **Figure 11** depicts a schedule obtained that way.

Clearly, the assignment in **Figure 11** is not an acceptable solution for our problem. But, a change in the order in which the intervals of green are assigned to the maximal cliques gives the acceptable assignment shown in **Figure 12**.

The schedule is obtained by assigning consecutive green intervals to cliques $\{x, z\}$, $\{x, w\}$, $\{x, y\}$, and $\{p, y\}$, in that order. Following the notation introduced in section 5. **Maximal Cliques**, this assignment can be obtained from the graph in **Figure 10** by listing its maximal cliques as

$$K_1 = \{x, z\}, \quad K_2 = \{x, w\}, \quad K_3 = \{x, y\}, \quad K_4 = \{p, y\},$$

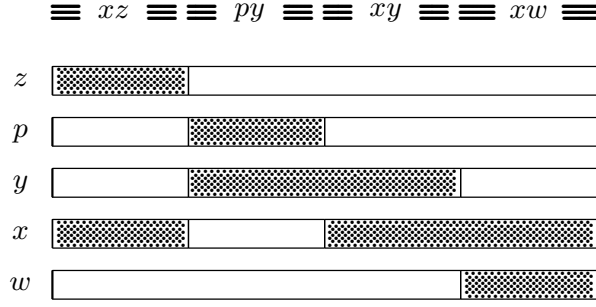


Figure 11. Schedule obtained by assigning green intervals to the maximal cliques in the order $\{x, z\}$, $\{p, y\}$, $\{x, y\}$, and $\{x, w\}$.

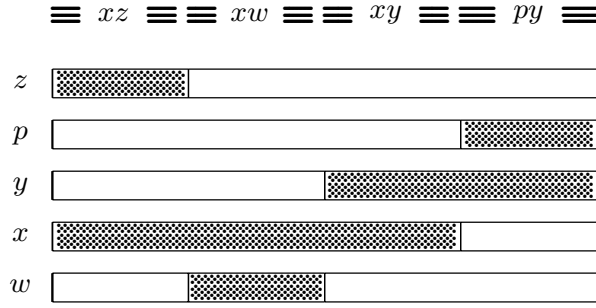


Figure 12. Schedule obtained by assigning green intervals to the maximal cliques in the order $\{x, z\}$, $\{x, w\}$, $\{x, y\}$, and $\{p, y\}$.

by defining

$$J_1 = [0, d_1), \quad J_2 = [d_1, d_1 + d_2), \quad J_3 = [d_1 + d_2, d_1 + d_2 + d_3),$$

$$J_4 = [d_1 + d_2 + d_3, d_1 + d_2 + d_3 + d_4),$$

and, finally, by assigning

$$S_z = J_1, \quad S_p = J_4, \quad S_y = J_3 \cup J_4, \quad S_x = J_1 \cup J_2 \cup J_3, \quad S_w = J_2.$$

So, as in our first example,

$$S_v = \bigcup_{i:v \in K_i} J_i.$$

That is, we assign to each vertex v the union of those J_i corresponding to the maximal cliques to which v belongs.

We now work out this example. Suppose that $r_z = 25$, $r_p = 10$, $r_y = 15$, $r_x = 45$, and $r_w = 15$, while $N = 70$ units of time. We keep the ordering of the maximal cliques introduced above, and find that the linear program (LP) to consider is

$$\begin{aligned} &\text{maximize } 2d_1 + 2d_2 + 2d_3 + 2d_4 \text{ subject to} \\ &d_1 \geq 25 \\ &d_4 \geq 10 \end{aligned}$$

$$\begin{aligned}
 d_3 + d_4 &\geq 15 \\
 d_1 + d_2 + d_3 &\geq 45 \\
 d_2 &\geq 15 \\
 d_1 + d_2 + d_3 + d_4 &\leq 70 \\
 d_1, d_2, d_3, d_4 &\geq 0.
 \end{aligned}$$

Lindo returns the optimal solution $d_1 = 40, d_2 = 15, d_3 = 0, d_4 = 15$, for which the total amount of green is 140 units. The corresponding phasing is given in **Figure 13**. Note that this three-phase phasing is not an intersection assignment.

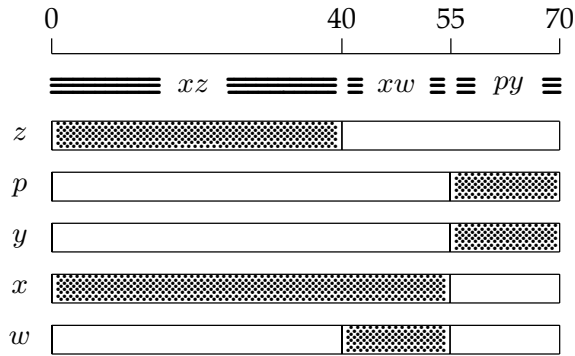


Figure 13. The optimal solution, with 140 units of green time.

Exercise

- For the example and the r_v 's and N given above, find an intersection assignment with the same optimal total amount of green of the previous phasing.

9. Consecutive Orderings

For our method to work, we have to make sure that each vertex is assigned at most one interval of green time per cycle. As **Figure 11** shows, that will in general not occur unless green light is assigned consecutively to the cliques to which vertex v belongs. In fact, in the schedule of **Figure 11**, x receives a first interval of green as a member of $\{x, z\}$, no green next because it is not a member of $\{p, y\}$, and then green again as a member of $\{x, y\}$ and $\{x, w\}$.

Definition. An ordering K_1, K_2, \dots, K_n of the maximal cliques of a graph is *consecutive* if whenever a vertex belongs to cliques K_i and K_j with $i < j$, it necessarily belongs to all intermediate cliques, that is, to cliques K_h with $i < h < j$.

Given that the graph of **Figure 2** on p. 2 has only two maximal cliques, labeling any of them K_1 and the other one K_2 results in a consecutive ordering

of them. The graph of **Figure 10**, on the other hand, admits orderings of its maximal cliques that are not consecutive, as well as consecutive orderings different from the one that we have used.

Exercise

6. Find all consecutive orderings of the maximal cliques of the graph of **Figure 10**.

There are graphs whose maximal cliques do not admit any consecutive ordering; **Figure 14** shows one.

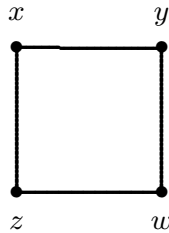


Figure 14. A graph with maximal cliques that do not admit any consecutive ordering.

The maximal cliques are $\{x, y\}$, $\{y, w\}$, $\{w, z\}$, and $\{z, x\}$. Since the first two contain y and the others don't, the first two must be consecutive in the ordering. If, say, $\{x, y\}$ immediately precedes $\{y, w\}$, then $\{z, x\}$ must precede them immediately and $\{w, z\}$ follow them immediately. But this is not a consecutive ordering of the four cliques. A similar argument applies if $\{y, w\}$ immediately precedes $\{x, y\}$.

It can be shown that the graph of **Figure 14** does not admit an intersection assignment, either. In fact, Fulkerson and Gross [1965] have given a complete characterization of graphs whose maximal cliques admit a consecutive ordering. To state their result, we introduce some terminology.

Definition. An *interval graph* is a graph whose vertices can be put in correspondence with intervals of the real line in such a way that two vertices are adjacent if and only if the associated intervals intersect.

In other words, an interval graph is one that admits an intersection assignment. The Fulkerson and Gross result claims that a graph is an interval graph if and only if its cliques can be ordered consecutively. Applying this result to the graph of **Figure 14**, which as we know does not admit a consecutive ordering of its maximal cliques, we can say that it is not an interval graph. Simply put, it is impossible to find four intervals of the real line such that each one intersects exactly two of the others.

Exercises

7. Consider graph G in **Figure 15**. Find a consecutive ordering of its maximal cliques different from the one used in **Section 8**. For the same rs and N given there, find a schedule for the traffic graph and compare it with the one in the text.

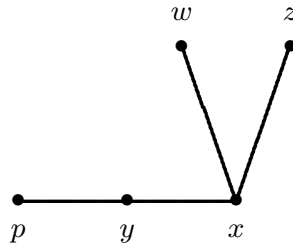


Figure 15. Graph for Exercise 7.

8. Schedule the intersection in **Figure 16** subject to the compatibilities depicted by the graph in **Figure 17**. Assume that $r_w = r_z = 18$, $r_v = 10$, $r_x = 15$, $r_y = r_t = 20$, and $N = 60$ units of time. What kind of schedule do you get?

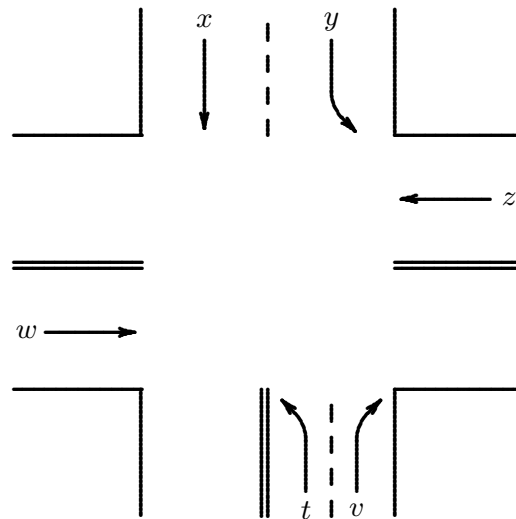


Figure 16. Intersection for Exercise 8.

10. Full Intersection Assignments

How can we be sure that the method that we have used gives best-possible phasings or intersection assignments? To clarify these matters, we give careful definitions of the concepts involved and state a number of fundamental results.

We note first that by *interval* we mean an ordinary interval of the real line, with the proviso that we consider only bounded intervals (open, closed, or

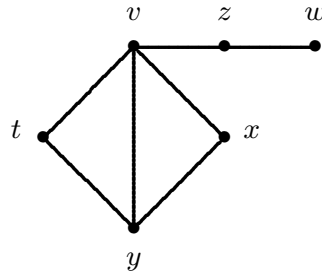


Figure 17. Compatibilities graph for the intersection of Figure 16 and Exercise 8.

half-open). An interval has a left endpoint a and a right endpoint b , with $a \leq b$. An interval may be empty. Following the usual terminology, we call $b - a$ the *length* of the interval.

Definition. A *traffic graph* is a graph G together with real numbers $r_v > 0$ associated with vertices v and a real number $N > 0$. In the sequel, we say for brevity that G, r, N is a traffic graph. A *phasing* of a traffic graph is a phasing S such that

- a) $|S_v| \geq r_v$ for each vertex v , and
- b) $|\bigcup_v S_v| \leq N$.

An *intersection assignment* of a traffic graph is defined in a similar manner. The quantity $|\bigcup_v S_v|$ in b) is what we previously called the “cycle length.” Following Roberts [1979], we call this quantity the *measure* of the phasing. Analogously, $\sum_v |S_v|$ was previously called the “total amount of green,” but is referred to here as the *measure score* of the phasing. We apply the terms “measure” and “measure score” also to intersection assignments.

The *phasing number* (*intersection number*) of a traffic graph is the supremum of the measure scores taken over allphasings (intersection assignments) of the traffic graph. These numbers are not defined if the traffic graph does not admit a phasing or intersection assignment (see Exercise 10.) On the other hand, if a traffic graph admits a phasing (intersection assignment), then this supremum is finite (see Exercise 9.)

Exercises

- 9. Prove that if a traffic graph hasphasings, then the supremum in the definition of phasing number is finite, as is the intersection number.
- 10. A traffic graph G, r, N may have no phasing, for example if N is too small with respect to the r_v s. Indeed, we clearly need N to be at least as large as the largest r_v . However, even if this condition is met, the existence of a phasing depends on the underlying graph itself. Show this by considering the values $r_x = 1, r_y = r_z = 2, r_t = 1, N = 4$, for both graphs in Figure 18.

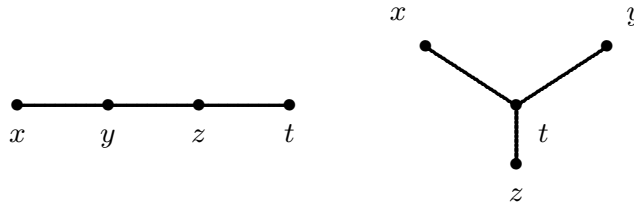


Figure 18. Graphs for Exercise 10.

11. Let G be a connected interval graph, S an intersection assignment. Show that $\bigcup_v S_v$ is an interval. Show by example that the converse of this statement is not true.

We have seen how the traffic-light scheduling problem gives rise to an LP, and showed with an example how a solution of the LP determines a phasing of the associated traffic graph. The question addressed at the beginning of this section can now be rephrased as:

Is the measure score of the phasing that we obtained equal to the phasing number of the graph?

We discuss this issue in the sequel.

When dealing with intersection assignments, the ones with the properties that we need are those in which the intervals of the assignment (i.e., the green-light intervals of the traffic-light scheduling) are as large as possible. More precisely, we consider assignments S such that the enlargement of an S_v results in incompatible streams being given green light simultaneously. We give a formal definition:

Definition. An intersection assignment S is called *full* when, for every vertex w and interval $I \subseteq \bigcup S_v$ such that $S_w \subset I$, there exists a vertex z for which $S_w \cap S_z = \emptyset$ but $I \cap S_z \neq \emptyset$.

Exercise 12 asks you to prove that, given an intersection assignment S that is not full, there exist full intersection assignments with the same measure as S and at least as large a measure score. The example below discusses an idea that you might find useful in solving this exercise. Consider the graph and intersection assignment in **Figures 19** and **20**.

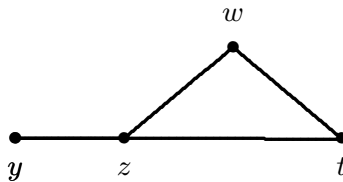


Figure 19. A graph.

This assignment is not full. We could extend S_z without reaching beyond $\bigcup S_v$, for example to $[0, 120)$, not creating forbidden green overlap. After that,

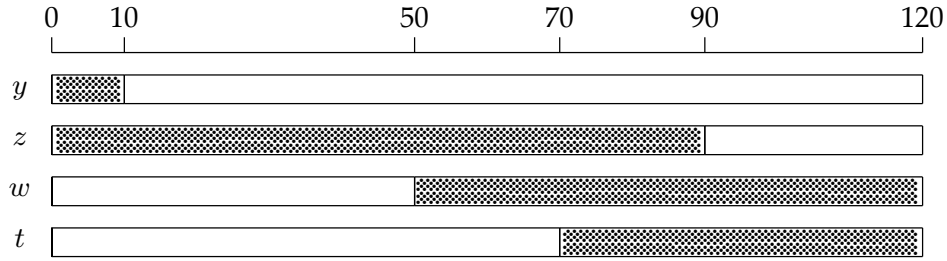


Figure 20. A non-full intersection assignment for the graph of Figure 19.

it would be possible to extend both S_w and S_t to $[10, 120)$. This would be a full intersection assignment. The measure stays the same, while the measure score has increased.

To justify the implicit assumption that this process can always be carried out, we have to convince ourselves that we will not have to continue indefinitely with larger and larger I s. This convincing is left for **Exercise 12**.

Exercises

- 12. Suppose that graph G has an intersection assignment S . Prove that there exists a full intersection assignment T such that $|T_v| \geq |S_v|$ for all vertices v and $\bigcup T_v = \bigcup S_v$.
- 13. Let G be an interval graph, S a phasing, and $\alpha \in \bigcup S_v$. Prove that the set of vertices v such that $\alpha \in S_v$ is a clique.
- 14. Find a traffic graph with an intersection assignment and an instant α so that the set of all streams green at α is not a maximal clique.

Lemma. *Let G be an interval graph, S a full intersection assignment. If $\alpha \in \bigcup S_v$, then the set of vertices v such that $\alpha \in S_v$ is a maximal clique. (Exercise 14 asked you to find an example of an intersection assignment for which this conclusion does not hold.)*

Proof: Assume first that G is connected. Let K be the set of vertices v such that $\alpha \in S_v$. We know by **Exercise 13** that K is a clique. Suppose that it is not maximal; then there exists a vertex $w \notin K$ which is adjacent to all vertices in K . This means that $\alpha \notin S_w$, and for all v such that $\alpha \in S_v$, we have $S_v \cap S_w \neq \emptyset$.

Since $\alpha \notin S_w$, α is either smaller than all elements in S_w or larger than all elements in S_w . We assume the former, since the proof is similar in the latter case.

We can also assume that of all vertices w satisfying the above properties, we choose one such that S_w has the smallest left endpoint. Further, we choose w , if possible, so that S_w is left-closed. In other words, we can assume that no other S_z satisfying the same properties as S_w has elements smaller than all of S_w . (**Figure 21** will help you follow this proof.)

Consider an interval I defined as follows. Its left endpoint is α and its right endpoint is the same as that of S_w . Moreover, I is left-closed and is right-open

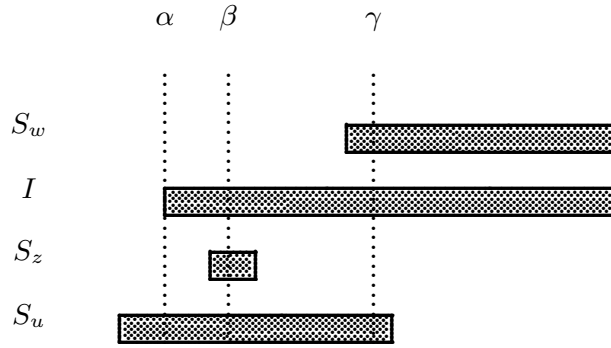


Figure 21. Illustrative interval graph for the proof of the **Lemma**.

or right-closed depending on whether S_w is right-open or right-closed. Then $S_w \subset I$. We claim that $I \subseteq \bigcup S_v$. In fact, any point in I is between α and a point in S_w . But $\alpha \in \bigcup S_v$ and $S_w \subseteq \bigcup S_v$, so any point in I is between points in $\bigcup S_v$. Since this set is an interval by **Exercise 11**, any point in I is also in $\bigcup S_v$.

Since S is a full intersection assignment, there exists a vertex z such that $S_z \cap S_w = \emptyset$ but $S_z \cap I \neq \emptyset$. This implies in particular that w and z are not adjacent, and by the hypothesis on w , $z \notin K$. Moreover, by choosing any $\beta \in S_z \cap I$, we have $\beta \in S_z$, $\alpha \leq \beta$, and β strictly less than all the elements in S_w .

We claim that z is adjacent to every vertex in K . In fact, let $u \in K$, so that $\alpha \in S_u$. By the assumption on w , we know that u and w are adjacent, so there exists $\gamma \in S_u \cap S_w$. Now we have $\alpha \leq \beta < \gamma$. Since $\alpha, \gamma \in S_u$ we have $\beta \in S_u \cap S_z$, so u and z are adjacent.

To summarize, we have found a vertex $z \notin K$ (so $\alpha \notin S_z$) adjacent to every vertex in K . Since $\alpha \notin S_z$, α is either less than every element in S_z or greater than every element in S_z . Now $\beta \in S_z$, so necessarily α is less than every element in S_z . This shows that z satisfies the same conditions as w , and moreover S_z contains β , which is smaller than all the elements in S_w . This contradicts the choice of w , so we conclude that K is maximal. \square

The general case, when G is not necessarily connected, can be easily reduced to the case just proved. We leave the details as **Exercise 16**.

Exercises

15. Extend the partial proof in the **Lemma** to the case where α is arbitrary. You have to show that, if G is a connected graph having a full intersection assignment S , $\alpha \in \bigcup S_v$, K is the set of vertices v such that $\alpha \in S_v$, and w is a vertex adjacent to all vertices in K , then $w \in K$.
16. Complete the proof of the **Lemma**. You have to show that the result holds also when G is not necessarily connected.

It is convenient to introduce some terminology. Suppose that G, r, N is a traffic graph with maximal cliques K_1, \dots, K_s , and denote by p_i the number

of vertices of K_i . We say that

$$\begin{aligned} &\text{maximize } \sum p_i d_i \text{ subject to} \\ &\sum_{i:v \in K_i} d_i \geq r_v \text{ for all vertices } v \\ &\sum d_i \leq N \\ &d_i \geq 0 \text{ for all } i \end{aligned}$$

is the LP *associated* with G, r, N , and refer to it as L when the meaning is clear from the context. For each ordering of the maximal cliques of the traffic graph, there exists an LP. We argued, however, that the existence of a solution is independent of these orderings, so we do not concern ourselves with distinguishing the various LPs. On the other hand, in order to build a phasing, we need to assume that the maximal cliques are consecutively ordered.

11. An Algorithm

We summarize here the ideas discussed so far that can be used to find phasings, intersection assignments, and associated parameters of a given traffic graph. In some cases, this algorithm will not give a definite answer; we point this out where appropriate. We also provide notes that clarify some aspects of the procedure and state theorems that justify the claims made. Proofs of these theorems will be published elsewhere.

11.1 The Algorithm

Consider the traffic graph G, r, N .

1. Obtain the maximal cliques of G and determine whether they admit a consecutive ordering. (For the small examples that we discuss here, this determination can be done by considering a few simple possibilities.¹) If the maximal cliques admit a consecutive ordering (i.e., G is an interval graph), go to 2), otherwise (G is not an interval graph) go to 5).
2. Set up L in the following manner.
 - (a) The variables are, say, d_1, \dots, d_s (one per maximal clique.)
 - (b) For each vertex v the sum of d s corresponding to all cliques that contain v must be $\geq r_v$.
 - (c) The sum of all d s must be $\leq N$.

¹Raised numerals in this section indicate notes on the algorithm, which are collected in a subsection after it.

- (d) All d_s are nonnegative.
- (e) The objective function (to be maximized) is the sum of all terms $p_i d_i$, where p_i is the number of vertices of the i -th maximal clique.

Go to 3).

3. If L is not feasible, the traffic graph does not admit a phasing or an intersection assignment.^{2,3} Exit. Otherwise, compute an optimal solution d_1, \dots, d_s and the value of L . Define as many intervals as maximal cliques as follows. Set $J_1 = [0, d_1)$, $J_2 = [d_1, d_1 + d_2)$, $J_3 = [d_1 + d_2, d_1 + d_2 + d_3) \dots$; then, for each vertex v , set S_v equal to the union of the intervals corresponding to the maximal cliques to which v belongs (i.e., $S_v = \bigcup_{i: v \in K_i} J_i$.) This defines an optimal phasing of the traffic graph, and its phasing number is the value of L . Go to 4).
4. If $d_i > 0$ for all i , the procedure described in 3) gives an optimal intersection assignment.⁴ The intersection number, phasing number, and the value of L are all the same. Exit. If some $d_i = 0$, solve the following LP (which in this case has an optimal solution.)

$$\begin{aligned} \text{minimize } & \sum d_i \text{ subject to} \\ & \sum_{i: v \in K_i} d_i \geq r_v \text{ for all vertices } v \\ & d_i \geq 0 \text{ for all } i. \end{aligned}$$

In the sequel, we refer to this LP as L' . Its optimal solution, if it exists, is denoted by N_0 .

We necessarily have $N_0 \leq N$. If $N_0 < N$, then the traffic graph admits an intersection assignment, and its phasing and intersection numbers are equal to the value of L .^{5,6} Exit. If $N_0 = N$, the algorithm is inconclusive.⁷ Exit.

5. Determine subgraphs H of G with the same vertex set as G in the following way. Start with the empty graph (with no edges) and keep adding edges of G as long as the resulting subgraph is an interval graph (note that the empty graph is an interval graph.) Stop when this is no longer possible. Consider every subgraph that can be obtained this way⁸, and assign to the vertices the same values r_v and N as the original graph has. Go to 6).
6. For each subgraph found applying the procedure just described, determine N_0 as defined in 4) above.
 - (a) If for any of these it happens that its $N_0 = N$, the algorithm is inconclusive.⁷ Exit.
 - (b) If all subgraphs have their $N_0 > N$, then G, r, N does not admit a phasing.^{3,9,10} Exit.

- (c) Otherwise, for those subgraphs that have $N_0 < N$, compute their intersection number following 3) and 4). Then the maximum of these numbers is the phasing number of the traffic graph.^{9,10} A phasing of the subgraph realizing this maximum can be obtained as in 3). This is a phasing of the original traffic graph realizing its phasing number. Exit.

11.2 Notes on the Algorithm

¹ Larger examples require more-sophisticated techniques. See details in section 12. **Comments**

² **Definition.** An *interval traffic graph* is a traffic graph G, r, N where G is an interval graph.

³ **Theorem 1 (Phasing Existence).** An interval traffic graph admits a phasing iff L is feasible (the feasibility of L does not depend on the ordering of the maximal cliques.) In this case, L has an optimal solution, and the phasing number is equal to the value of L .

⁴ **Proposition (Phasing as an Intersection Assignment).** Suppose that L has a feasible solution giving rise to a phasing as described in 3). Then this phasing is an intersection assignment iff all the variables are positive. In this case moreover, the phasing and intersection numbers of the interval traffic graph are equal to the value of L .

⁵ **Theorem 2 (Phasing Number).** Suppose that L' is feasible and has value N_0 (see **Exercise 20.**) If $N_0 < N$, the phasing and intersection numbers of the interval traffic graph are equal to the value of L .

⁶ In this case, we don't know if an intersection assignment realizing the intersection number exists, nor how to find it if it does.

⁷ This is of no consequence for practical applications, for in this case we can just increase N by an arbitrarily small amount to obtain a traffic graph (or traffic subgraphs) for which an intersection assignment can be determined.

⁸ In building one of these subgraphs, there may be several possible choices of the next edge to add. All these choices have to be considered, thus obtaining in general several different subgraphs.

⁹ **Definition.** Given a graph G , a *maximal interval spanning subgraph* is a spanning subgraph of G that is an interval graph and is not included in another such subgraph.

¹⁰ **Theorem 3 (Phasing in Terms of Spanning Subgraphs).** Let G, r, N be a traffic graph, and let \mathcal{M} be the set of maximal interval spanning subgraphs of

G . Assume that for every element H in \mathcal{M} such that the LP associated with H, r, N is feasible, this LP also has a feasible solution d_1, \dots, d_s with $\sum d_i < N$. Then G, r, N admits a phasing iff \mathcal{M} has elements H such that H, r, N has an intersection assignment. In this case, the phasing number of G, r, N is the maximum intersection number of such H, r, N .

Exercises

Note. The following exercises may require application of the notes above.

17. Suppose that L has an optimal solution d_1, \dots, d_s . Show that $\sum d_i = N$.
18. Show that a graph such that its maximal cliques can be consecutively ordered is an interval graph.
19. Prove that if L is feasible, then it has an optimal solution.
20. a) If L' is feasible, then it has an optimal solution.
 b) If L is feasible, then L' is feasible and $N_0 \leq N$. In this case, given a number l , there is a feasible solution d_1, \dots, d_s of L such that $\sum d_i = l$ iff $N_0 \leq l \leq N$.
21. The **Proposition (Phasing as an Intersection Assignment)** and **Theorem 2 (Phasing Number)** give conditions ensuring that the intersection number is equal to the value of L . Prove that those conditions are independent. More specifically, consider the following:
 i) L has an optimal solution d_1, \dots, d_s with $d_i > 0$ for all i .
 ii) L has a feasible solution d_1, \dots, d_s such that $\sum d_i < N$.
 You are asked to find a traffic graph that satisfies i) but not ii) and a traffic graph that satisfies ii) but not i). Hint: See the graphs in **Figure 22**.

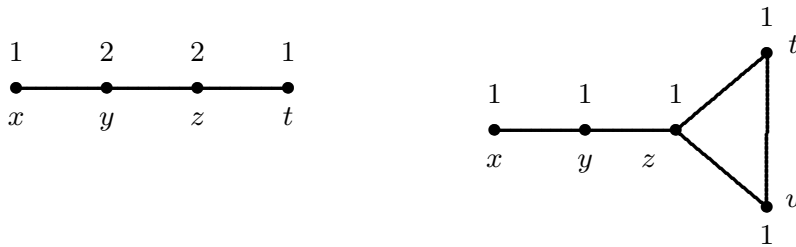


Figure 22. Graphs for Exercise 21.

22. Let G, r, N be an interval traffic graph admitting a phasing. Show that its phasing number is a maximum, i.e., there exists a phasing of G, r, N whose measure score is the phasing number.

23. (This exercise requires a substantial effort, hence may be appropriate for a class project.) Find the phasing number of the traffic graph associated with the intersection in **Figure 23**. Apart from the obvious incompatibilities, we specify that stream a is incompatible with d and f ; b is incompatible with c and e ; c is incompatible with e ; and d is incompatible with f . The minimum green times are: 45 s for a and b , 21 s for c , 30 s for d and f , and 36 s for e . Assume that $N = 135$.

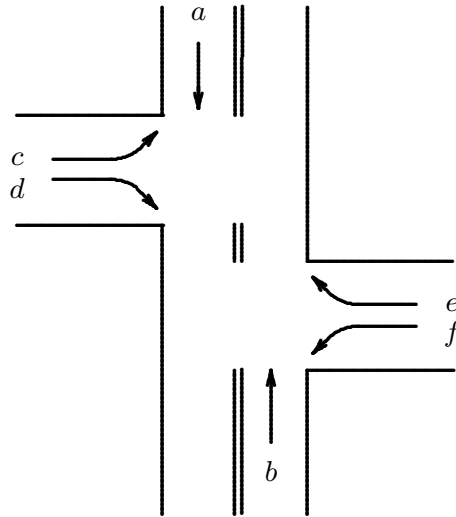


Figure 23. Intersection for Exercise 23.

24. Find a phasing for the traffic intersection of **Exercise 23** that realizes its phasing number.
25. Consider the intersection of **Figure 24**.

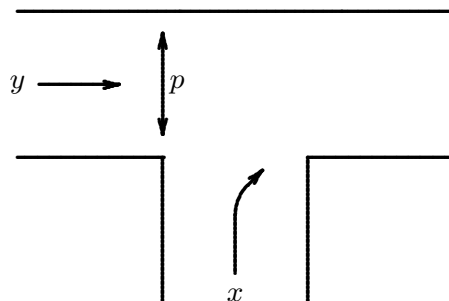


Figure 24. Intersection for Exercise 25.

- a) Assume that x and y are compatible, as well as p and x . Let $r_p = 65$, $r_x = 45$, $r_y = 50$, $N = 110$ units of time. Can you find a schedule?
- b) Solve L' to find N_0 .

- c) Find an optimal schedule using $N = 115$ units of time. Determine if you obtained a phasing or an intersection assignment.
- d) Let $N = 120$. Find two different optimal schedules with the same total amount of green and determine if they are phasings or intersection assignments.
- e) Is it possible to improve upon the total green for cases c) and d)?
26. Consider the intersection in **Figure 25**.

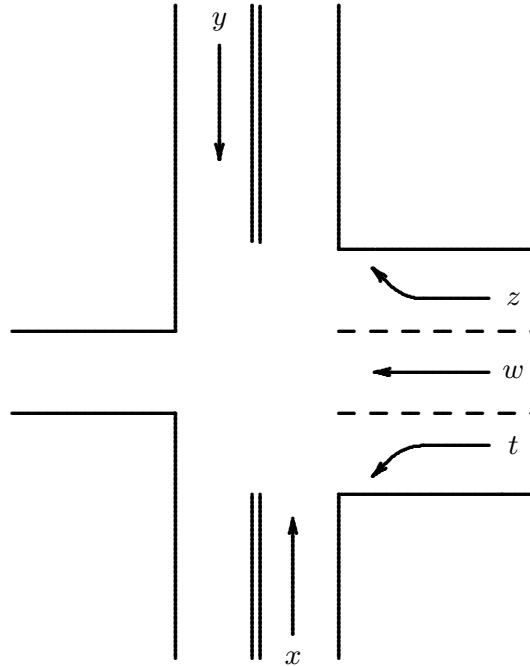


Figure 25. Intersection for **Exercise 26**.

- a) Assume that the only incompatible streams are x and w , x and t , y and t , and y and w . Draw the associated graph, give reasonable minimum amounts of green for the streams as well as for the cycle length, and find a schedule. What kind of schedule do you obtain?
- b) Now consider that x and z are incompatible and redo the problem.
- c) Could your solutions to a) and b) be improved under the stated conditions?

12. Comments and Suggestions for Further Reading

The phasing of traffic lights is a complex problem, involving empirical determinations as well as theoretical considerations. The model that we present

here abstracts, from all the relevant aspects, a few that lend themselves to treatment using standard mathematical tools. Other issues are treated at length in the literature. Hurd et al. [1955] discuss various parameters that we take for granted. The determination of appropriate values for these parameters is in part theoretical and in part observational. For example, Hurd et al. give ways to determine the length of yellow light that take into consideration the stopping time (the time to stop a moving vehicle) and the clearance time (which depends, among other things, on the width of the intersection and stopping time.) They also give a formula for the total delay (the time for a vehicle to start after the light turns green, summed over all vehicles.) This formula, a function of the length of red light and the number of vehicles stopped, involves parameters that are observed empirically. They argue that the delay is minimum when the red intervals are as short as possible.

It is also possible to introduce probabilities into the model. The arrival of cars can be modeled by a Poisson process, so we can determine the cycles when the number of cars is much larger than average, producing accumulation of vehicles. In this way, one can strike a balance between a cycle short enough to avoid excessive delays and one long enough to keep the number of cycles with accumulation at a tolerable level.

As a result of our lengthy discussion, we can in principle find optimal phasings and intersection assignments of some kinds of graphs. Nevertheless, some aspects that we left aside for simplicity merit further consideration.

For the very small examples that we considered, it was easy to find the maximal cliques of the graphs involved, decide whether a consecutive ordering is possible, and thereby determine whether a given graph is an interval graph. This trial-and-error method is unacceptably unwieldy and time-consuming for larger graphs.

There are systematic approaches to organize and simplify this task considerably. We refer you to Golubic [1980] for a detailed discussion of these topics, which we now summarize briefly. Every simple cycle of an interval graph has a chord [Golubic 1980, Theorem 8.4.] (To clarify the statement, we remind you that a cycle in a graph is given by a sequence of vertices v_0, \dots, v_k with $k \geq 2$, such that v_i is adjacent to v_{i+1} for $0 \leq i \leq k$, where we agree for convenience that $v_{k+1} = v_0$. *Simple* means that the vertices are distinct, while a (triangular) *chord* of the cycle is an edge of the form $\{v_i, v_{i+2}\}$, for some i , $0 \leq i \leq k$, where again for simplicity we agree that, additionally, $v_{k+2} = v_1$.)

Recognizing this property for a given graph takes time linear in the number of vertices plus edges of the graph [Corollary 4.6.] Further, if a graph has the property that every simple cycle has a chord, its maximal cliques can be listed in linear time [Theorem 4.17.] Finally, checking whether the maximal cliques can be ordered consecutively can also be done in linear time, a result due to Booth and Leuker [Theorem 8.5.]

The results presented here are only a selection of much more detailed investigations due to Roberts [1979] and Opsut and Roberts [1981, 1983a, 1983b]. They consider assignments not only of intervals but also of various types of

sets, as well as multiple applications of these models. These papers contain also discussions of the use of linear programs to solve the computational problems that arise.

The characterization of interval graphs due to Fulkerson and Gross is not easy to apply. For a different approach, expressed in terms of cycles, see Gilmore and Hoffman [1964].

Index

- associated LP, 19
- consecutive ordering, 12
- cycle length, 3
- duration of a clique, 6
- full intersection assignment, 16
- incompatible streams, 1
- intersection assignment, 9
- intersection assignment of a traffic graph, 15
- intersection number of a traffic graph, 15
- interval graph, 13
- interval traffic graph, 21
- L (associated LP), 19
- linear program (LP), 7–8
- maximal clique, 5
- maximal interval spanning subgraph, 21
- measure of a phasing, intersection assignment, 15
- measure score of a phasing, intersection assignment, 15
- phase, 2
- phasing, 9
- phasing of a traffic graph, 15
- phasing number of a traffic graph, 15
- traffic graph, 15

13. Solutions to the Exercises

1. Of all the cliques containing the given one, consider one with the maximum number of vertices.
2. Reversing the order of the cliques results in reversing the variables d_1, d_2 in the linear program. An optimal solution is therefore $d_1 = 15, d_2 = 25$. We obtain then $S_y = [0, 15), S_x = [0, 40), S_z = S_w = [15, 40)$. The solution is essentially the same, the only difference being that the origin for the measurement of time has been shifted by 15 s.
3. Possible factors: Traffic density, street width, number of streets converging at the intersection.
4. Possible answers: Maximize the minimum amount of green, minimize the maximum amount of red, minimize the total number of light changes per cycle.
5. You can find alternative optimal solutions, for example, by moving and stretching the green and red periods in **Figure 13**. This way, we obtained $d_1 = 35, d_2 = 15, d_3 = 5, d_4 = 15$, which induces $S_z = [0, 35), S_p = [55, 70), S_y = [50, 70), S_x = [0, 55),$ and $S_w = [35, 50)$. This assignment, being an intersection assignment, is also a phasing (different from the one given in the text.)
6. As before, we denote the cliques by $K_1 = \{x, z\}, K_2 = \{x, w\}, K_3 = \{x, y\},$ and $K_4 = \{p, y\}$. Since x belongs to $K_1, K_2,$ and K_3 , these cliques must be consecutive in the ordering. Further, y belongs to K_3 and K_4 , so these two cliques must be next to each other. We conclude that the consecutive orderings of the maximal cliques are $(K_1, K_2, K_3, K_4), (K_2, K_1, K_3, K_4), (K_4, K_3, K_1, K_2),$ and (K_4, K_3, K_2, K_1) .
7. The maximal cliques are $K_1 = \{x, z\}, K_2 = \{x, w\}, K_3 = \{x, y\}, K_4 = \{p, y\}$. The consecutive orderings are $(K_1, K_2, K_3, K_4), (K_2, K_1, K_3, K_4), (K_4, K_3, K_1, K_2),$ and (K_4, K_3, K_2, K_1) .

Changing the ordering does not change the solution fundamentally, only the order of the variables. For example, for the ordering (K_2, K_1, K_3, K_4) we obtain the optimal solution $(15, 40, 0, 15)$. The corresponding schedule is $S_z = [15, 55), S_p = S_y = [55, 70), S_x = [0, 55), S_w = [0, 15)$.

8. For the ordering $K_1 = \{w, z\}, K_2 = \{z, v\}, K_3 = \{v, x, y\}, K_4 = \{y, t, v\},$ the corresponding LP is

$$\begin{aligned} &\text{maximize } 2d_1 + 2d_2 + 3d_3 + 3d_4 \quad \text{subject to} \\ &\quad d_1 \geq 18 \\ &\quad d_1 + d_2 \geq 18 \\ &\quad d_2 + d_3 + d_4 \geq 10 \end{aligned}$$

$$\begin{aligned}
d_3 &\geq 15 \\
d_3 + d_4 &\geq 20 \\
d_4 &\geq 20 \\
d_1 + d_2 + d_3 + d_4 &\leq 60 \\
d_1, d_2, d_3, d_4 &\geq 0.
\end{aligned}$$

Lindo gives $d_1 = 18, d_2 = 0, d_3 = 22, d_4 = 20$ units of time. The corresponding schedule is a phasing.

Note. Don't worry if you get a different optimal solution. In fact, all optimal solutions are given by $(18, 0, d_3, 42 - d_3)$, where $15 \leq d_3 \leq 22$. Indeed, it is easy to verify that these are feasible solutions all with the same value of the objective function, 162. Conversely, if (d_1, d_2, d_3, d_4) is an optimal solution, then $(e_1, e_2, e_3, e_4) = (18, 0, d_3 + d_1 - 18 + d_2, d_4)$ is also a feasible solution for which the value of the objective function is

$$2e_1 + 2e_2 + 3e_3 + 3e_4 = 2d_1 + 2d_2 + 3d_3 + 3d_4 + d_1 - 18 + d_2.$$

Since $d_1 - 18 + d_2 \geq 0$, we have

$$2d_1 + 2d_2 + 3d_3 + 3d_4 + d_1 - 18 + d_2 \geq 2d_1 + 2d_2 + 3d_3 + 3d_4,$$

but (d_1, d_2, d_3, d_4) is optimal, so $d_1 - 18 + d_2 = 0$. Given that $d_1 \geq 18, d_2 \geq 0$, we conclude that $d_1 = 18, d_2 = 0$, and $d_3 + d_4 = 42$. Finally, $d_4 = 42 - d_3 \geq 20$ implies $d_3 \leq 22$.

9. It suffices to show that $\sum |S_v|$ is bounded above over all phasings (or intersection assignments) of the traffic graph. But this is obvious, because Nn is a bound, where n is the number of vertices in G .
10. For the leftmost graph, the phasing $S_x = [0, 1), S_y = [0, 2), S_z = [2, 4)$, and $S_t = [3, 4)$ satisfies all constraints. For the other graph, given that x, y , and z are incompatible, we need at least $r_x + r_y + r_z = 5$ units of time to satisfy the constraints.
11. We show that $U = \bigcup_v S_v$ is an interval by showing that this set contains every element between two of its points. Suppose that $p < r < q$ with $p \in S_v, q \in S_w$. Since G is connected, there is a path z_1, \dots, z_n in G with $z_1 = v$ and $z_n = w$. Since each interval S_{z_i} intersects the next one, $S_{z_{i+1}}$, it is obvious by induction that $U' = \bigcup S_{z_i}$ is an interval. But $p, q \in U'$, so $r \in U'$, hence $r \in U$.
A counterexample is provided by the graph consisting of two vertices x, y and no edges, with the intersection assignment $S_x = [0, 1), S_y = [1, 2)$.
12. If the given intersection assignment S is not full, then there are vertices w such that there exist intervals I satisfying $S_w \subset I \subseteq \bigcup S_v$ and for every vertex v different from w and not adjacent to it, $S_v \cap I = \emptyset$. For brevity, vertices such as w are called in this proof *defective* for S . It is sufficient to

show that the number of defective vertices can be reduced by at least 1. To prove this, fix a vertex w defective for S and note first that $S_w \neq \emptyset$. For otherwise, since I is nonempty and $I \subseteq \bigcup S_v$, I would intersect some S_v , while v would be different from w and not adjacent to it.

Consider now the union J of all such intervals I . Since J is a union of intervals with nonempty intersection, it is also an interval. Define $T_v = S_v$ for $v \neq w$ and $T_w = J$. We claim that T is an intersection assignment for G . Note that $J \supset S_w$. Assume that v is adjacent to w . Then the inequality $S_v \cap S_w \neq \emptyset$ implies that $S_v \cap J \neq \emptyset$, i.e., $T_v \cap T_w \neq \emptyset$. Suppose now that $v \neq w$ and $S_v \cap J \neq \emptyset$. By the definition of J , there exists an interval I as above such that $S_v \cap I \neq \emptyset$. Now, by the assumption on I , v is adjacent to w .

Note also that for all v we have $T_v \supseteq S_v$, so $|T_v| \geq |S_v|$, and $\bigcup T_v = \bigcup S_v$.

To prove that the number of defective vertices has decreased by at least 1, we establish two facts:

- w is not defective for T .
- If a vertex is defective for T , then it is also defective for S .

Regarding the first claim, recall that $\bigcup T_v = \bigcup S_v$, and suppose that there exists an interval L such that $J \subset L \subseteq \bigcup S_v$ and $L \cap T_v = \emptyset$ for every vertex $v \neq w$ not adjacent to w . Then $L \supset S_w$ and, since $T_v \supseteq S_v$, we have $L \cap S_v = \emptyset$ for every $v \neq w$ not adjacent to w . This shows that L is one of the intervals whose union is J , which contradicts $J \subset L$.

As for the second claim, let u be defective for T , so that in particular $u \neq w$. There exists an interval L such that $T_u \subset L \subseteq \bigcup S_v$ and $L \cap T_v = \emptyset$ for every vertex $v \neq u$ not adjacent to u . Since $S_v \subseteq T_v$, we also have that $L \cap S_v = \emptyset$ for every vertex $v \neq u$ not adjacent to u . But $S_u = T_u$, so we can conclude that u is defective for S .

13. Let K be the set of vertices v such that $\alpha \in S_v$. If v and w are distinct elements of K , then $\alpha \in S_v \cap S_w$. By definition of phasing, v, w are adjacent.
14. Consider the interval graph and intersection assignment of **Figures 19** and **20**. You can take $\alpha = 100$. This intersection assignment is not full.
15. We may additionally assume that α is larger than all elements in S_w (the other case was treated in the text). For each vertex v , let $T_v = \{\beta: -\beta \in S_v\}$. It is routine to prove that T is a full intersection assignment and $-\alpha \in \bigcup T_v$. Further, the set of vertices v such that $-\alpha \in T_v$ is also K , and $-\alpha$ is less than all elements in T_w . The result follows then from the case proved in the text.
16. Let G be an interval graph, S a full intersection assignment, and $\alpha \in \bigcup S_v$. We wish to prove that the set K of vertices such that $\alpha \in S_v$ is a maximal clique. We know that it is a clique, so we only need to prove that it is maximal.

Since K is connected, it is included in a connected component H of G . Now H is a connected graph, and $(S_v)_{v \in H}$ is an intersection assignment for

it. It is also full, for if I is an interval such that $S_w \subset I \subseteq \bigcup_{v \in H} S_v \subseteq \bigcup_{v \in G} S_v$, then we know that there exists a vertex z in G such that $S_z \cap S_w = \emptyset$ but $S_z \cap I \neq \emptyset$. Since $I \subseteq \bigcup_{v \in H} S_v$, $S_z \cap S_u \neq \emptyset$ for some $u \in H$, hence z is adjacent to u . But H is a connected component, so $z \in H$. Since K is the set of vertices v of H such that $\alpha \in S_v$, by **Exercise 15**, K is a maximal clique of H . It is also a maximal clique of G , because H is a connected component.

17. Suppose by contradiction that $\sum d_i < N$. Then we could replace each d_i with $d'_i = d_i + \epsilon$ for some $\epsilon > 0$ and still have a feasible solution of L . Indeed, we could take $\epsilon = (N - \sum d_i)/s$, for then

$$\sum_{i:v \in K_i} d'_i \geq \sum_{i:v \in K_i} d_i \geq r_v$$

for all vertices v , and

$$\sum d'_i = \sum (d_i + \epsilon) = \sum d_i + s\epsilon = N,$$

but

$$\sum p_i d'_i = \sum p_i d_i + \epsilon \sum p_i > \sum p_i d_i,$$

which contradicts the fact that d_1, \dots, d_s is optimal.

18. We follow the construction described in the algorithm. Let G be a graph admitting a consecutive ordering of its maximal cliques K_1, \dots, K_s . Choose s arbitrary positive numbers d_1, \dots, d_s and consider the s intervals

$$J_i = \left[\sum_{h=1}^{i-1} d_h, \sum_{h=1}^i d_h \right), \quad 1 \leq i \leq s.$$

For each vertex v , let $S_v = \bigcup_{i: v \in K_i} J_i$. We prove that $S = (S_v)_v$ is an intersection assignment for G .

Consider an arbitrary vertex v . Since by hypothesis the maximal cliques are consecutively ordered, v belongs to K_h, K_{h+1}, \dots, K_l but to no other maximal cliques; so we have $S_v = J_h \cup J_{h+1} \cup \dots \cup J_l$, which is an interval.

We prove next that if v, w are distinct vertices such that $S_v \cap S_w \neq \emptyset$, then v, w are adjacent. Let $\alpha \in S_v \cap S_w$. Then $\alpha \in J_h$ with $v \in K_h$, and $\alpha \in J_r$ with $w \in K_r$, so $h = r$ and v, w are adjacent.

To finish, we prove that if v, w are adjacent, then $S_v \cap S_w \neq \emptyset$. In fact, $\{v, w\}$ is a clique, so by **Exercise 1**, v and w belong to some maximal clique K_i . Then $\emptyset \neq J_i \subseteq S_v \cap S_w$, so $S_v \cap S_w \neq \emptyset$.

19. Suppose that the LP is feasible. To prove that it has an optimal solution, we need to show only that the objective function is bounded above over all feasible solutions. But this is obvious, for $N \max_i p_i$ is an upper bound.

20. a) Note that for every feasible solution d_1, \dots, d_s of L' , we have $\sum d_i \geq 0$.
- b) If L has feasible solutions, then it also has an optimal solution d_1, \dots, d_s , which will also be a feasible solution of L' . By **Exercise 17**, $\sum d_i = N \geq N_0$. A similar argument proves that if $l = \sum d_i$ for a feasible solution of L , then $N_0 \leq l \leq N$. Finally, let $N_0 \leq l \leq N$. We consider an optimal solution d_1, \dots, d_s of L' . Then d'_1, d_2, \dots, d_s , with $d'_1 = d_1 + l - N_0$, is a feasible solution of L with $d'_1 + \sum_2 d_i = l$.
21. For the left graph, we take $N = 3$. Then L is

$$\begin{aligned} &\text{maximize } 2d_1 + 2d_2 + 2d_3 \text{ subject to} \\ &d_1 \geq 1 \\ &d_1 + d_2 \geq 2 \\ &d_2 + d_3 \geq 2 \\ &d_3 \geq 1 \\ &d_1 + d_2 + d_3 \leq 3 \\ &d_1, d_2, d_3 \geq 0. \end{aligned}$$

An optimal solution of this LP is $d_1 = d_2 = d_3 = 1$, so i) holds. The value of the LP is 6. On the other hand, by **Exercise 20**, condition ii) is equivalent to the value of the following LP being less than 3:

$$\begin{aligned} &\text{minimize } d_1 + d_2 + d_3 \text{ subject to} \\ &d_1 \geq 1 \\ &d_1 + d_2 \geq 2 \\ &d_2 + d_3 \geq 2 \\ &d_3 \geq 1 \\ &d_1, d_2, d_3 \geq 0. \end{aligned}$$

But adding the first four inequalities, we obtain $2(d_1 + d_2 + d_3) \geq 6$, so for any feasible (in particular, any optimal) solution, we have $d_1 + d_2 + d_3 \geq 3$. Alternatively, you can run Lindo to verify that the value of this LP is 3.

For the right graph, we also take $N = 3$, and choose ordering $K_1 = \{x, y\}$, $K_2 = \{y, z\}$, $K_3 = \{z, t, u\}$ of the maximal cliques. Then L is

$$\begin{aligned} &\text{maximize } 2d_1 + 2d_2 + 3d_3 \text{ subject to} \\ &d_1 \geq 1 \\ &d_1 + d_2 \geq 1 \\ &d_2 + d_3 \geq 1 \\ &d_3 \geq 1 \\ &d_3 \geq 1 \\ &d_1 + d_2 + d_3 \leq 3 \end{aligned}$$

$$d_1, d_2, d_3 \geq 0.$$

This LP has value 8. To determine N_0 , we solve

minimize $d_1 + d_2 + d_3$ subject to

$$d_1 \geq 1$$

$$d_1 + d_2 \geq 1$$

$$d_2 + d_3 \geq 1$$

$$d_3 \geq 1$$

$$d_3 \geq 1$$

$$d_1, d_2, d_3 \geq 0.$$

Since this LP has value 2, condition ii) is satisfied. On the other hand, if d_1, d_2, d_3 is an optimal solution of L , from the last inequality we have $d_3 \leq 3 - (d_1 + d_2)$, and consequently

$$8 = 2d_1 + 2d_2 + 3d_3 \leq 2(d_1 + d_2) + 3(3 - (d_1 + d_2)) = 9 - (d_1 + d_2).$$

Therefore, $d_1 + d_2 = 1$, which together with $d_1 \geq 1$ implies $d_2 = 0$, hence i) does not hold.

22. By **Theorem 1 (Phasing Existence)**, L is feasible, therefore (**Exercise 19**) it has an optimum, which generates an optimal phasing as in 3) of the algorithm.
23. The graph of the intersection is given in **S1**.

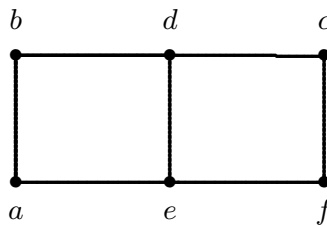


Figure S1. Solution for **Exercise 23**.

An argument similar to the one we used for the graph in **Figure 14** shows that this is not an interval graph.

Following the algorithm to the letter for noninterval graphs can be lengthy and tedious, so we take some shortcuts to simplify the task. We use **Theorem 3 (Phasing in Terms of Spanning Subgraphs)** to calculate the phasing number.

To obtain a maximal spanning subgraph that is an interval graph, we have to remove edges so that the two squares are eliminated. Further, it is easily seen that the hexagon obtained by removing edge de is not an interval graph either. We conclude that we need to remove two edges, so that no square remains. We obtain in this way 15 traffic subgraphs, 9 of which we

depict in **Figure S2**. The remaining ones, II b, III b, V b, VII b, VIII b, and IX b, are symmetric to the corresponding ones in the figure with respect to a vertical line through the middle of the graph (obviously, the labels and durations remain fixed.)

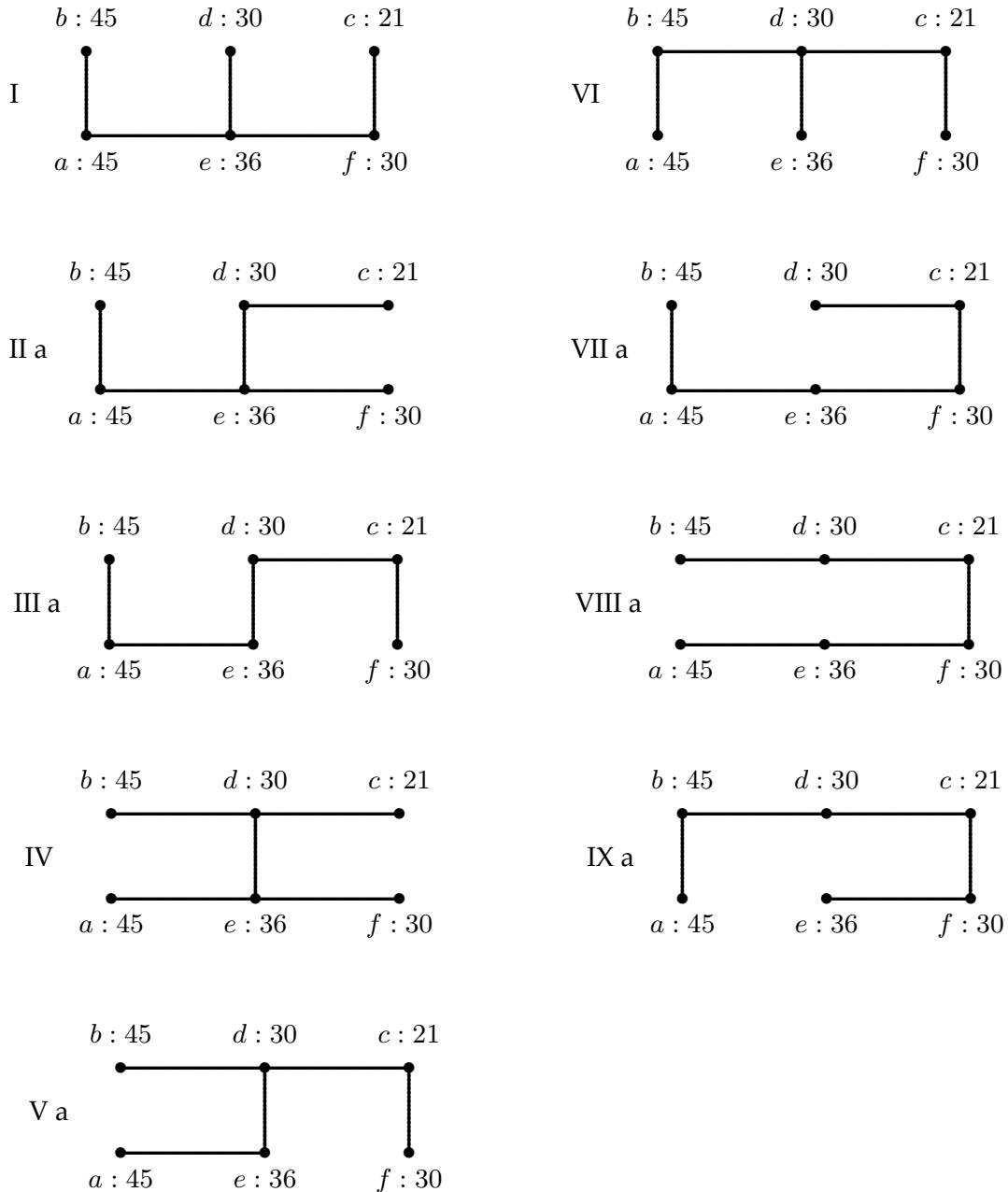


Figure S2. Traffic graphs for Exercise 23.

To use **Theorem 3 (Phasing in Terms of Spanning Subgraphs)** to find the phasing number of the given graph, we have to show that these 15 subgraphs are interval graphs. Note that as graphs (i.e., disregarding the r_i s and N), there are just three of them, those in **Figure S3**. The labeling

indicates consecutive orderings of the maximal cliques, which shows that these are indeed interval graphs.

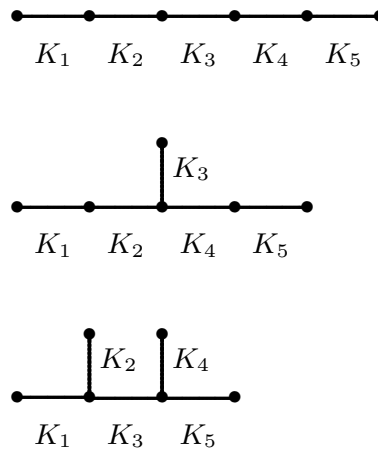


Figure S3. Interval graphs corresponding to the traffic graphs of **Figure S2**.

Our next task is to determine which of the 15 LPs associated with the traffic subgraphs are feasible, and for those who are, give a feasible solution such that the sum of the variables is strictly less than N , as required by **Theorem 3 (Phasing in Terms of Spanning Subgraphs)**.

It is convenient to label the edges uniformly in all subgraphs, as in **Figure S4**. Note that the edges are also the maximal cliques of the subgraphs; we use the same labels for the LP variables.

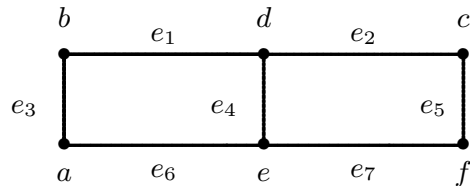


Figure S4. Labeling of edges of **Figure S1**.

In principle, we would have to inspect 15 LPs. We could solve them using Lindo, but we can proceed more efficiently by noting that these LPs are closely related. Indeed, if we consider all the restrictions associated with the original graph of **Figure S1**, we can obtain the LP associated with one of the 15 subgraphs by adding the condition that two of the variables be equal to zero (actually, we disregard the objective function, since at this point we only need feasible solutions.) It follows from this that our task will be simplified considerably if we can find solutions of the system associated with the original graph where as many as the variables as possible are zero, for then one such solution will generally satisfy several of the LPs.

Consider the following system:

$$\begin{aligned}
 e_3 + e_6 &\geq 45 \\
 e_1 + e_3 &\geq 45 \\
 e_2 + e_5 &\geq 21 \\
 e_1 + e_2 + e_4 &\geq 30 \\
 e_4 + e_6 + e_7 &\geq 36 \\
 e_5 + e_7 &\geq 30 \\
 e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7 &\leq 135 \\
 e_1, e_2, e_3, e_4, e_5, e_6, e_7 &\geq 0.
 \end{aligned}$$

This system is simple enough that we can find the required solutions by hand (if you prefer a more systematic approach, you can solve the system using a software package, taking as objective the minimization of the sum of the variables, which will probably force several of them to be zero). We see by inspection that $e_2 = 30$, $e_3 = 45$, $e_7 = 36$, and the remaining variables equal to zero, is a solution of this system, and therefore it is also a solution of the systems associated with the subgraphs that contain the edges e_2 , e_3 , and e_7 , namely IIa, Vb, VIIa, VIIIb, and IXa. Note also that this solution satisfies $e_1 + \dots + e_7 < N$.

We pick now one of the subgraphs not covered by the preceding calculation, e.g. subgraph I. The system to consider now is

$$\begin{aligned}
 e_3 + e_6 &\geq 45 \\
 e_3 &\geq 45 \\
 e_5 &\geq 21 \\
 e_4 &\geq 30 \\
 e_4 + e_6 + e_7 &\geq 36 \\
 e_5 + e_7 &\geq 30 \\
 e_3 + e_4 + e_5 + e_6 + e_7 &\leq 135 \\
 e_3, e_4, e_5, e_6, e_7 &\geq 0.
 \end{aligned}$$

We take $e_3 = 45$, $e_4 = 36$, $e_5 = 30$, and the remaining variables equal to zero. This shows that the LPs associated with the subgraphs I, IIIa, IIIb, and VI are feasible and have a solution satisfying the condition of the theorem.

The LP associated with IV is not feasible, since some of the constraints are $e_6 \geq 45$, $e_1 \geq 45$, $e_2 \geq 21$, $e_7 \geq 30$, and this is not compatible with $e_1 + e_2 + e_4 + e_6 + e_7 \leq 135$.

Next we consider IIb. For this, we choose $e_1 = 45$, $e_5 = 30$, $e_6 = 45$, and the remaining variables equal to zero. This shows that the LPs corresponding to IIb, Va, VIIb, VIIIa, and IXb satisfy the conditions we need.

By **Theorem 2 (Phasing Number)**, each one of the 14 subgraphs with feasible associated LP has intersection number equal to the value of the LP.

On the other hand, by **Exercise 17**, if d_1, \dots, d_s is an optimal solution of one of these LPs, we have $\sum d_i = N$, and, given that all maximal cliques of the subgraph consist of two vertices, the value of the LP is $2 \sum d_i = 2N = 270$. Invoking now **Theorem 3 (Phasing in Terms of Spanning Subgraphs)**, we conclude that this is also the phasing number of the original traffic graph.

24. It suffices to find a phasing of one of the maximal spanning subgraphs that realizes its intersection number. We choose for example the subgraph labeled I in **Figure S2** and solve the associated LP, obtaining $e_3 = 75, e_6 = 0, e_4 = 30, e_7 = 9, e_5 = 21$ (this order indicates a consecutive ordering of the maximal cliques.) We determine a phasing in the usual manner, obtaining $S_a = S_b = [0, 75), S_c = [114, 135), S_d = [75, 105), S_e = [75, 114), S_f = [105, 135)$.
25. a) There is no schedule satisfying the requirements, since $r_p + r_y = 115 > 110$.
- b) The LP to solve is

$$\begin{aligned} &\text{minimize } d_1 + d_2 \quad \text{subject to} \\ &\quad d_1 \geq 65 \\ &\quad d_1 + d_2 \geq 45 \\ &\quad d_2 \geq 50 \\ &\quad d_1, d_2 \geq 0. \end{aligned}$$

The value is $N_0 = 115$.

- c) Solving

$$\begin{aligned} &\text{maximize } 2d_1 + 2d_2 \quad \text{subject to} \\ &\quad d_1 \geq 65 \\ &\quad d_1 + d_2 \geq 45 \\ &\quad d_2 \geq 50 \\ &\quad d_1 + d_2 \leq 115 \\ &\quad d_1, d_2 \geq 0 \end{aligned}$$

results in $d_1 = 65, d_2 = 50$ units of time. The corresponding schedule is given by $S_p = [0, 65), S_x = [0, 115), S_y = [65, 115)$. It is an intersection assignment, and therefore a phasing.

- d) Possible schedules are given by $d_1 = 70, d_2 = 50$ and $d_1 = 65, d_2 = 55$, the total amount of green is 240. These schedules are intersection assignments, and therefore phasings.
- e) These schedules cannot be improved (see 4) of the algorithm).

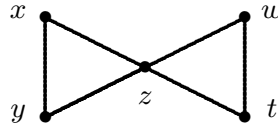


Figure S5. Graph for solution to Exercise 26a.

26. a) See Figure S5. We choose the ordering $K_1 = \{x, y, z\}$, $K_2 = \{z, t, w\}$ of the maximal cliques. Let's agree that $r_x = 65$, $r_y = 50$, $r_z = 110$, $r_t = 80$, $r_w = 70$, and $N = 150$ units of time. The corresponding LP is

$$\begin{aligned} &\text{maximize } 3d_1 + 3d_2 && \text{subject to} \\ &d_1 \geq 65 \\ &d_1 \geq 50 \\ &d_1 + d_2 \geq 110 \\ &d_2 \geq 80 \\ &d_2 \geq 70 \\ &d_1 + d_2 \leq 150 \\ &d_1, d_2 \geq 0. \end{aligned}$$

The value is 435 units of time, achieved for $d_1 = 65$, $d_2 = 80$. The corresponding schedule is $S_x = S_y = [0, 65)$, $S_z = [0, 145)$, $S_w = S_t = [65, 145)$. It is an intersection assignment.

- b) This time the graph is that of Figure S6. We choose the same values for the r s but $N = 180$.

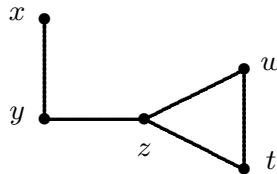


Figure S6. Graph for solution to Exercise 26b.

We choose the consecutive ordering $K_1 = \{x, y\}$, $K_2 = \{y, z\}$, $K_3 = \{z, t, w\}$ of the maximal cliques. The corresponding LP is

$$\begin{aligned} &\text{maximize } 2d_1 + 2d_2 + 3d_3 && \text{subject to} \\ &d_1 \geq 65 \\ &d_1 + d_2 \geq 50 \\ &d_2 + d_3 \geq 110 \\ &d_3 \geq 80 \\ &d_3 \geq 70 \\ &d_1 + d_2 + d_3 \leq 180 \end{aligned}$$

$$d_1, d_2, d_3 \geq 0.$$

An optimal solution is $d_1 = 65$, $d_2 = 0$, $d_3 = 115$. The associated schedule is $S_x = S_y = [0, 65)$, $S_z = S_w = S_t = [65, 180)$. It is a phasing but not an intersection assignment, because y and z are compatible but $S_y \cap S_z = \emptyset$.

- c) These assignments cannot be improved, by the **Proposition on Phasing as an Intersection Assignment** and by **Theorem 1 (Phasing Existence)**.

References

- Fulkerson, D.R., and O.A. Gross. 1965. Incidence matrices and interval graphs. *Pacific Journal of Mathematics* 15: 835–855.
- Gilmore, P.C., and A.J. Hoffman. 1964. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics* 16: 539–548.
- Golumbic, Martin. 1980. *Algorithmic Graph Theory and Perfect Graphs*. New York: Academic Press.
- Hurd, Frederick W., Theodore M. Matson, and Wilbur S. Smith. 1955. *Traffic Engineering*. New York: McGraw-Hill.
- Opsut, Robert J., and Fred S. Roberts. 1981. On the fleet maintenance, mobile radio frequency, task assignment, and traffic phasing problems. In *Theory and Applications of Graphs*, edited by Gary Chartrand, 479–492. New York: Wiley.
- _____. 1983a. Optimal I -intersection assignments for graphs: A linear programming approach. *Networks* 13: 317–326.
- _____. 1983b. I -colorings, I -phasings, and I -intersection assignments for graphs, and their applications. *Networks* 13: 327–345.
- Roberts, Fred S. 1979. On the mobile radio frequency assignment problem and the traffic light phasing problem. *Annals of New York Academy of Sciences* 319: 466–483.
- Stoffers, Karl E. 1968. Scheduling of traffic lights—A new approach. *Transportation Research* 2: 199–234.
- Van Vuren, Tom, and Dirk Van Vliet. 1992. *Route Choice and Signal Control: The Potential for Integrated Route Guidance*. Aldershot, Hants, UK: Avebury Press.

About the Authors

Sara Kuplinsky received her B.A. in Computer Science from the University of Buenos Aires, Argentina, and her M.Sc. in Applied Mathematics from Rutgers University. She has been at Ramapo College of New Jersey since 1991, teaching a range of mathematics courses and supervising support services for remedial and general education offerings in the field. Her primary interest is undergraduate mathematics education.

Julio Kuplinsky received his B.A. from the University of Buenos Aires, his M.Sc. from Universidad Simón Bolívar, Venezuela, and his Ph.D. from Rutgers University, all in mathematics. He has worked for several years in software development. He also teaches mathematics courses at Yeshiva University and Montclair State University on a part-time basis.