HOW TO MAKE CITY TRAFFIC ADAPTIVE

ON AN ALGORITHM FOR THE ADAPTIVE AUTOMATIC CONTROL OF THE MAIN VEHICULAR TRAFFIC STREAMS IN AN URBAN AREA

A. Butrimenko and H. Strobel

June 1974 WP-74-14

How to make City Traffic Adaptive

On An Algorithm for the Adaptive Automatic Control of the Main Vehicular Traffic Streams in an Urban Area

by

A. Butrimenko and H. Strobel

1. Motivation

We will consider a situation as shown in Fig. 1*: a car arrives in an urban district, e.g. at the Swain Square intersection. The driver of the car wishes to go, for example, to the Koopmans Place intersection (his destination point). The driver wants to know which route will require the minimum of driving time, taking into account the different traffic densities in the different streets. We are going to give him advice by means of a special kind of road sign which will be placed at a certain distance before each of the possible intersections. These road signs, which show the driver the recommended route, would be the same as those that are used today to point out the recommended direction and the only difference would be that these proposed road signs would have to be changeable and controlled by an electronic system. The control device (e.g. a decentralized installed mini-computer or even a central control computer) has to advise the driver of any car arriving at the Swain Square intersection on the best route to take, i.e. the driver chooses his route through the intersections (see Fig. 1) from the road signs shown in Raiffa Street.

---

* The map in Fig. 1 is completely fictional and any resemblance to a town or names in existence is purely coincidental.

It may be of interest to point out that this problem is very similar to a traffic control problem arising in data networks [1]. Therefore, the aim of this paper is to point out that the same basic ideas, which have appeared useful for data network systems, could also be useful in solving the described street traffic control problem.

2. Basic Idea of the Algorithm

We now describe, without going into detail, the basic idea of the proposed algorithm. We illustrate it by means of Fig. 1.

The advice to the driver will be given, as mentioned above, by means of changing road signs. The main control information for the updating of these signs will be stored in certain matrices. The number of matrices corresponding to one particular intersection is equal to the number of streets leading to this intersection and to the number of road sign tables. For example, the number of matrices for the Swain Square intersection is equal to four (assuming that each of these streets has traffic in the direction of the intersection). Each of these matrices has the number of rows equal to the number of selected destination points. In our example there are eight of them: IIASA Place, Koopmans Place, Bell Terminal, Rosanov Circus, Mayerhofer Airport, Thompson Place, Manne Square and Swain Square*/. The number of columns is equal to the number of possible driving directions from this street.

---

*/ The choice of the destination points depends on many factors and should not correspond to the number of destination nodes put on the corresponding road signs. The destination points put on the road signs could be different for different intersections--for example if the direction is not changed--but all these problems will have to be investigated in more detail. There is also another way to avoid the bewildering forest of traffic signs. "Before beginning his journey every driver keys a destination identification code in his vehicle. On the basis of these destination requests" [2] the control device at each intersection sends him, by means of a special wireless system, in the recommended direction which is displayed on the driver panel of his car (Fig. 4).

For example, from Raiffa Street leading to the Swain Square intersection, we have three possible driving directions and three columns of the matrix. The structure of the matrix, with respect to Raiffa Street, is shown in Fig. 2.

The elements of the matrix have to represent the estimated minimum driving time from the next intersection to the destination point, assuming that the following streets are chosen in an optimal manner. The separate row, shown under the matrix contains the travelling times needed to reach the corresponding intersection. For example, in the row for Koopmans Place the travelling time needed to reach this destination point from the road sign in Letov Street equals 21; from the road sign in Haefele Avenue - 14; from Dantzig Way - 12.

To reach the corresponding road signs, from the road signs in Raiffa Street, we need 10, 21 and 6 minutes respectively. These last figures include the time for crossing the Swain Square intersection (including waiting time) plus driving time through the following streets, i.e. Letov Street, Haefele Avenue and Dantzig Way. On the basis of the information stored in the matrix, we can fix the road signs. For example, to reach Koopmans Place through Letov Street we need 21 + 10 = 31 minutes; through Haefele Avenue 14 + 21 = 35 minutes; and through Dantzig Way 12 + 6 = 18 minutes. Thus, we have to recommend turning right into Dantzig Way from Raiffa Street.

The problem is that we cannot keep the road signs fixed because of the stochastic nature of the traffic. We have to update the matrices taking into consideration the real traffic situation. The way to this is as follows. We calculate continuously the time* needed from the intersection at Swain Square which consists of waiting time in the corresponding queues and driving time through the intersection and following streets. For example, to drive from the intersection in the direction of Haefele Avenue and through

---

* This should be done by means of a special algorithm for real time estimation of the travelling times.

this avenue to the next road sign before IIASA Place, we need 21 minutes. This time depends on traffic density in Haefele Avenue and waiting time before Swain Square. Both of these times have to be measured.

We carry out this calculation for each intersection of each street (matrix). We will store this information in a special row for which the number of elements is equal to the number of possible driving directions. For example, for Raiffa Street we have 10, 21, 6. Given the information on estimated travelling time from IIASA Place to Koopmans Place, we can calculate the estimated time to reach Koopmans Place from Swain Square through Haefele Avenue as a sum of the measured travelling time from the road sign in Raiffa Street to the road sign in Haefele Avenue before IIASA Place and the estimated time from this point to Koopmans Place.

We repeat this calculation for each destination point and for each possible direction. Then we will take the minimum for each destination point out of the possible directions and store it as a special vector, each element of it giving the minimal time to reach each destination point from the intersection for which this calculation has been carried out. For example, for the matrix in Raiffa Street we have the following (Fig. 3) vector of minimum destination times. This vector will be sent to all neighbouring intersections with streets leading to this particular road sign table in Raiffa Street; in this case to the devices connected to the road signs in Bykov and Holling Streets. This vector will be stored as a column corresponding to the direction of Swain Square in the matrices in Holling and Bykov Streets. In turn this would also mean that information stored in the matrix in Raiffa Street, as shown in Fig. 2, has been sent from devices in Letov Street, Haefele Avenue and Dantzig Way, respectively.

The important point is that each matrix corresponding to the intersection, which is itself a destination point, contains a row consisting of constant zeroes, and the name of this row is the destination point. For example, in the matrix in Raiffa Street we have the row corresponding to the destination point Swain Square which contains constant elements equal to zero. By updating continuously the matrices and rearranging the road signs, we will supply the drivers with information on the shortest time routes.

We have already formulated this algorithm and are going to carry out the following investigations in the near future.

3. Some Proposals for Detailed Investigations

- Development of an algorithm for real time estimation of travelling times;
- Simulation of the street network system on a digital computer with different traffic densities;
- Comparison of the sub-optimal (decentralized) algorithm mentioned above with other algorithms;
- Answering the question of whether it is possible to determine the real optimal solution, e.g. by a central computer in real time;
- Different optimizing criteria;
- Controlled changes of the lane directions in the street, as additional control parameters to the proposed system.
- Consideration of synergistic operation of lights (green waves);
- Take into account prediction of congestions by the changing of road signs.

References

1 Butrimenko, A., "Adaptive Routing Technique and Simulation of Communication Networks." IV ITC, Munich, 1970.

2 Von Tomkewitsch, Romuald, "Traffic Automation - its Possibilities and Limits. Observations by a Communications Engineer." 1st IFAC/IFIP Symposium on "Traffic Control and Transportation Systems", Versailles, 1-5 June 1970.

FIG. 1
A Network consisting of the main streets and intersections of an assumed city ("IIASA TOWN")

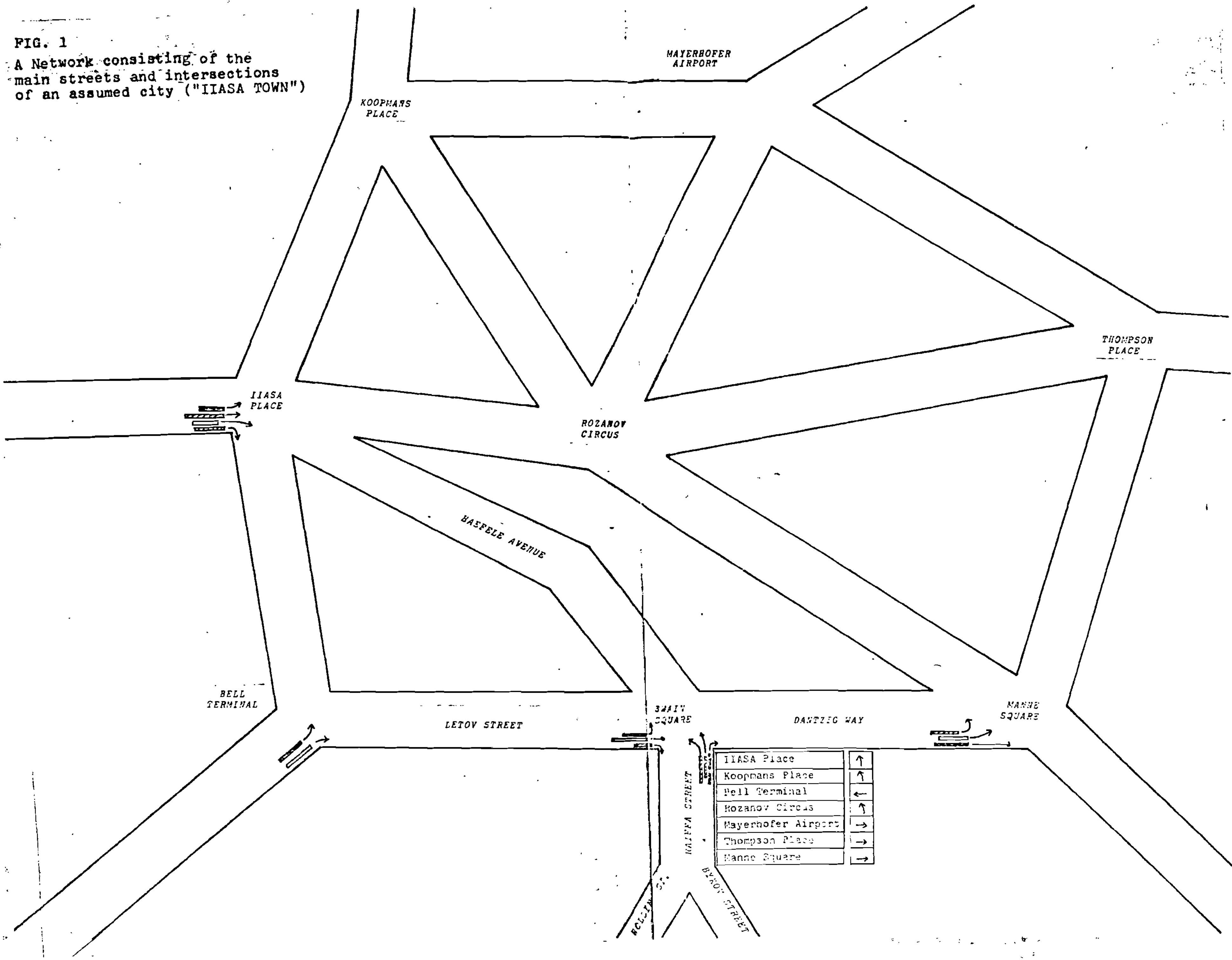

FIG. 2

Matrix of Travelling Times
for RAIFFA STREET at SWAIN SQUARE

	Letov Street ↰	Haefele Avenue ↑	Dantzig Way ↱
IIASA Place	18	0	30
Koopmans Place	21	14	12
Swain Square	0	0	0
Bell Terminal	0	7	34
Rosanov Circus	21	3	21
Mayerhofer Airport	52	30	20
Thompson Place	50	44	17
Manne Square	30	10	0
RAIFFA STREET	10	21	

FIG. 3

Vector of Minimum
Travelling Times

Minimum Travelling Time	Optimal Direction
21	↰
18	↱
0	
10	↰
24	↑
26	↱
23	↱
6	↱

FIG. 4