TRANSPORT and ROAD
RESEARCH LABORATORY

Department of the Environment
Department of Transport

SUPPLEMENTARY REPORT 595

# A GUIDE TO TRANSYT/7

by

P B Hunt and J V Kennedy



Urban Networks Division
Traffic Engineering Department
Transport and Road Research Laboratory
Crowthorne, Berkshire
1980
ISSN 0305–1315

CONTENTS

# A GUIDE TO TRANSYT/7

## ABSTRACT

TRANSYT/7 is a new version of a computer program which calculates optimum fixed time settings for traffic signals in a network. It is much faster than previous versions for large networks. The computer time required is now proportional to the number of links in the network rather than the square of the number of links. Evidence for this was obtained from ten networks of differing size and character.

Data input to the program has been simplified and the changes are detailed in an appendix. Links can be listed in any order and closed loops of links are identified by the program and modelled by an iterative method to ensure that accurate values of the delay and queue length are obtained for all links in the loop. The modelling of bottlenecks has been included in TRANSYT/7.

## 1. INTRODUCTION

A number of computer programs are available which derive the settings that best coordinate the traffic signals in an urban network[1,2]. TRANSYT[3,4] is one of the more successful of these programs and is used throughout the world. In large networks, however, the computer resources required to produce the signal timings are high because the running time of the computer is approximately quadratic in relation to the number of links. The aim of the work reported here was to restructure the optimisation procedure of TRANSYT to reduce the running time considerably whilst retaining the same high standard of signal coordination.

One of the more onerous tasks that are necessary when setting up the data for TRANSYT/6 is the ordering of links within a network. Although there are rules for this process it is a common source of difficulty and error. This process is handled automatically in TRANSYT/7.

## 2. THE TRANSYT PROGRAM

The TRANSYT program comprises two main sections, a traffic model and an optimisation procedure, as shown in Figure 1. The optimisation procedure uses the traffic model to evaluate the effect of changes in the signal settings. A Performance Index (PI) is defined as the weighted sum of the delay and number of stops made by all vehicles in the network and is a measure of the total disbenefit that is suffered by vehicles at the traffic signals. Optimising the signal settings is achieved by minimising the PI.

### 2.1 *The traffic model*

Traffic networks are considered to consist of the signalised junctions, called nodes, and unidirectional links which represent traffic streams between pairs of nodes. Nodes operate on a common cycle which is divided into a number of equal intervals or steps. Traffic that arrives at each signal stopline is represented by a histogram of the average flow during successive intervals throughout one cycle time. This histogram is called a cyclic flow profile. The length of the queue, and hence the average delay, the number of stops and the journey time through the link is calculated from this cyclic flow profile and from the red and green periods of the signals. The profile of traffic leaving the signal is also calculated and is then available to predict the shapes of the cyclic flow profiles of traffic on downstream links. Allowance is made for the time spent travelling, for platoon dispersion caused by the different speeds of individual vehicles and for traffic that turns into, and out from, side streets. The extra delay caused by random variations in the flow is included in the PI.

## 2.2 *The optimisation procedure in TRANSYT/6*

A simple 'hill-climb' method is used to find a minimum value for the performance index. The performance index is a discrete non-linear function of the signal timings and is defined only for timings which are an integral number of steps. There are other constraints on the signal timings; for example, the minimum green times for each stage must not be violated, and stage orders are fixed.

In TRANSYT the hill-climb takes the form of a rectangular search which varies the signal timings at each node in turn. The signal timings at the current node are shifted by an increment and if the PI decreases a further shift is made. For offset optimisation, all settings at a node are shifted simultaneously, whereas for optimisation of green durations (splits), each stage change time is shifted independently. The optimiser repeats the process for each increment in a preset sequence and can alternate between split and offset optimisation as desired. It is usual to start the optimisation with a large increment and then to reduce the increment when an approximate minimum has been found. Recommended sequences for optimisation of offsets only or for offsets and green durations are quoted in the Appendix (Sequences 1 and 3).

## 2.3 *Limitations in the TRANSYT/6 optimisation procedure*

The PI is first calculated for a set of initial signal settings. When a change is made in the settings at a node A, the traffic model is used to calculate the new value of the PI. Because each link in the model is processed only when all upstream links have been processed the PI will be unchanged on all links up to the first link L in the link list, which is connected to the node A. In TRANSYT/6, the PI is recalculated on all links subsequent to L in the list. Not all of these links will have changed.

For example, consider the arterial in Figure 2, where the links are processed in ascending numerical order. Where nodes A, B, C and D lie close together and the traffic flows are less than capacity, the platoons of traffic are likely to be well defined and the case for signal coordination is usually good. Changing the settings at node A necessitates recalculating the queueing model and PI on links 1, 2, 3, 4, 7, 10 and 16. However, TRANSYT/6 will also recalculate the PI on the remaining nine links, although these cannot possibly be affected by the change at A.

If the settings of node B are changed, then TRANSYT/6 does not recalculate links 1, 2 and 3, but does recalculate all the others.

Suppose now that link 4 is oversaturated. The platoon reaching link 7 will be independent of the settings chosen for signal A, because the flow leaving link 4 will remain at saturation level throughout the green. Thus, link 4 effectively decouples node A from the signal coordination in the rest of the network. It is then possible to find the optimum offset for node A without recalculating the PI on downstream links 7 and 10.

Optimisation will be faster if there is a general criterion to decide when it is necessary to reprocess a link and when traffic patterns can be regarded as unchanged. This is the intention behind TRANSYT/7.

## 3. OPTIMISATION IN TRANSYT/7

### 3.1 *The criterion for processing a link*

In TRANSYT/7 a link must be reprocessed if its cyclic flow profile has changed significantly. The criterion chosen is the percentage change in the cyclic flow profile leaving the stopline of the link. If the changes are large the queues and therefore the performance index will have changed and the link must be reprocessed. The reprocessing will continue downstream until the changes in the profile are insignificant or the edge of the network is reached.

Consider any link L. Suppose the traffic pattern leaving the stopline of L is stored in the vector $\underline{X}$. Each entry $X_n$ in $\underline{X}$ holds the average flow leaving the stopline at the $n^{th}$ step in the cycle. Let $\underline{X}^1$ be the corresponding departure pattern after the change in settings. Figure 3 shows an example in which the settings change at an upstream node but the green period on link L remains the same. The shaded area in Figure 3 is the alteration in profile caused by the change in signal settings. A measure of the percentage change in profile from the vector $\underline{X}$ to the vector $\underline{X}^1$ is given by

$$\Delta X = \frac{\left|\underline{X} - \underline{X}^1\right|}{\left|\underline{X}\right|} \times 100 \qquad (1)$$

where the modulus $\left|\underline{X}\right|$ of a vector $\underline{X}$ with N elements is defined by

$$\left|\underline{X}\right| = \sum_{i=1}^{N} \left|X_i\right| \qquad (2)$$

The percentage change in pattern $\Delta X$ is compared with a fixed value $\epsilon$, called the accuracy parameter. When $\Delta X$ is greater than $\epsilon$, the new profiles for downstream links should be calculated. When $\Delta X$ is less than $\epsilon$, the change in profile is considered negligible.

The value chosen for the accuracy parameter defines the local area affected by the signal change. In small networks, this local area will often be most of the network and hence, little computational time will be saved, because of the time involved in calculating and testing the percentage change in flow profile. In large networks, however, there can be substantial reduction in computer time.

A small value for the accuracy parameter implies that great accuracy is required and an extensive area is affected by each signal change. In contrast, a large value for the accuracy parameter leads to a much faster, but less accurate optimisation.

### 3.2 *Choice of accuracy parameter sequence*

The choice of accuracy parameter involves a compromise between the desired accuracy and speed of optimisation. The hill-climbing procedure adopts a large increment initially to find an approximate minimum. Whilst the increment is large, the PI need not be calculated accurately to test the effect of a shift in signal timings, and thus the accuracy parameter can be large.

Once an approximate minimum has been found, the increment and correspondingly the accuracy parameter must be decreased to find a better approximation to the true minimum. The final PI should be very close to that obtained by TRANSYT/6.

Recommended sequences of accuracy parameters corresponding to the sequences of hill-climb step-sizes for optimising offsets and green durations are quoted in the Appendix (Sequences 2 and 4). A value of 0.01 per cent is used to obtain accurate flow profiles. These are required for the initial signal timings and in the final step-size calculations in the hill-climb. To avoid the accumulation of errors, accurate profiles are calculated for all links at the completion of each step-size in the hill-climb.

### 3.3 *The link list*

In TRANSYT/6, a link list must be prepared by the user to define the sequence in which links are to be solved by the program. The user must also decide where to locate the dummy links that are needed to solve closed loops of links. Neither of these tasks is necessary with TRANSYT/7 and hence data preparation is easier and faster.

In TRANSYT/7, the link list is generated automatically by an algorithm within the program. Closed loops are broken by giving one of the links in each loop a uniform departure pattern and iterating until the change in pattern is negligible. The link list algorithm makes no attempt to minimise the number of links that are initially given uniform flow. The algorithm takes little time to run and is executed just once, after which the link ordering is unchanged.

### 3.4 *Bottlenecks*

A facility to model bottlenecks has been included in TRANSYT/7. A bottleneck is a point at which the maximum rate of flow is limited by a physical feature of the road layout such as a narrow bridge or roadworks. This gives a more accurate representation of traffic arriving at the downstream stopline. Bottlenecks are treated as junctions with 100 per cent green and a fixed saturation flow. Delays and stops are accumulated as at a normal stopline.

## 4. A COMPARISON BETWEEN TRANSYT/6 AND TRANSYT/7

### 4.1 *Examples*

Ten networks were selected for comparing TRANSYT/7 with TRANSYT/6. Together, they represent a wide variety of characteristics which may be found in different cities and countries. The examples chosen are taken from

1. Glasgow (6 networks)
2. TRANSYT/6 User Guide Example
3. San Jose, California, USA
4. Belgrade, Yugoslavia
5. The Bitterne Road, Southampton.

Table 1 shows the number of links and nodes in each network, and compares the central processor (CPU) time taken by an ICL System 4–70 computer to optimise signal timings using TRANSYT/6 and TRANSYT/7. The time saved varies considerably up to a maximum of 79 per cent. The difference in PI between TRANSYT/6 and TRANSYT/7 averages 0.1 per cent and does not exceed 0.3 per cent.

TABLE 1

A comparison of results obtained using TRANSYT/6 and TRANSYT/7

| Network | Nodes | Links | CPU time (seconds) | | Percentage time saved by TRANSYT/7 |
|---|---|---|---|---|---|
| | | | TRANSYT/6 | TRANSYT/7 | |
| 1. Glasgow Sub-area 1 | 46 | 178 | 3930 | 830 | 79 |
| 2. Glasgow Sub-area 2 | 29 | 138 | 1820 | 620 | 66 |
| 3. Glasgow Sub-area 3 | 10 | 30 | 110 | 90 | 18 |
| 4. Glasgow Sub-area 4 | 6 | 34 | 130 | 80 | 38 |
| TOTAL | 91 | 380 | 5990 | 1620 | 73 |
| 5. Glasgow Sub-area 5 | 39 | 150 | 2540 | 570 | 78 |
| 6. Glasgow bus area | 17 | 124 | 1310 | 820 | 37 |
| 7. User Guide Example | 5 | 18 | 62 | 60 | 3 |
| 8. San Jose | 48 | 131 | 2340 | 670 | 71 |
| 9. Belgrade | 23 | 129 | 1720 | 480 | 72 |
| 10. Bitterne Road | 17 | 131 | 710 | 530 | 25 |

The accuracy parameter sequence (A) adopted in TRANSYT/7 is the one recommended (Sequence 4) and is intended to maintain sufficient accuracy whilst saving as much computer time as possible. Thus, it is tailored to become more accurate as the performance index approaches the minimum.

Table 2 compares the performance indices obtained and computer time taken by TRANSYT/7 with two different accuracy parameter sequences. One of these is the sequence A. The other, B, gives equal accuracy for all step-sizes, following profile changes until these are less than 0.01 per cent, thus taking longer than A in the early part of the optimisation. Sequence A saves an average of about 20 per cent of the computer time and, in some cases, up to about 40 per cent over sequence B.

TABLE 2

TRANSYT/7: A comparison of two accuracy parameter sequences

| Network | Performance Index | | CPU time (seconds) | | Percentage time saved by sequence A |
|---|---|---|---|---|---|
| | A | B | A | B | |
| 1. Glasgow Sub-area 1 | 1544.0 | 1548.0 | 830 | 870 | 5 |
| 2. Glasgow Sub-area 2 | 635.1 | 635.5 | 620 | 840 | 16 |
| 3. Glasgow Sub-area 3 | 474.8 | 475.8 | 90 | 120 | 25 |
| 4. Glasgow Sub-area 4 | 211.4 | 211.7 | 80 | 90 | 11 |
| TOTAL | 2865 | 2872 | 1620 | 1920 | 16 |
| 5. Glasgow Sub-area 5 | 901.8 | 901.8 | 570 | 670 | 15 |
| 6. Glasgow bus area | 311.6 | 311.6 | 820 | 1270 | 35 |
| 7. User Guide Example | 117.8 | 117.8 | 60 | 64 | 6 |
| 8. San Jose | 148.0 | 148.1 | 670 | 1170 | 43 |
| 9. Belgrade | 945.8 | 942.8 | 460 | 580 | 18 |
| 10. Bitterne Road | 383.3 | 383.3 | 530 | 590 | 10 |

The central processor time saved by using TRANSYT/7 is greatest for large networks, particularly where there are no closed loops. Figure 4 shows how the CPU time taken increases with the number of links in the network. For TRANSYT/6, the increase is approximately quadratic, whereas for TRANSYT/7 it is almost linear.

4.1.1 Glasgow: The present TRANSYT area in Glasgow consists of 91 nodes and 380 links. Since this exceeds the maximum of 50 nodes and 250 links permitted by the TRRL version of the program, the area is divided into sub-areas. Different parts of the network require different cycle times and this type of consideration led to the four sub-areas shown in Figure 5.

The sub-areas vary in size and characteristics, for example, SA1 comprises most of the shopping area and SA3 is a network of one-way streets in a commercial and industrial area. All sub-areas except one, SA4, contain closed loops of links. The computer time required by TRANSYT/7 to optimise signal settings for all four sub-areas has been reduced by 73 per cent compared with TRANSYT/6. Most of this saving comes from the two larger networks Sub-areas 1 and 2.

Two other networks have also been optimised using TRANSYT/7. The areas were set up for research purposes. One of these, SA5, is a large network with no closed loops. The reduction in running time is from 2500 seconds to 600 seconds (78 per cent). The other network is a small one, chosen originally to study in detail the modelling of buses. In TRANSYT, different categories of vehicles can be modelled as if they travel along separate links, having a single shared stopline[6]. This network therefore has a high proportion of shared stoplines and contains seven closed loops. The computer time saved by using TRANSYT/7 (sequence A) is 37 per cent, which is less than that expected for the number of links. This is because many of the links share their stoplines. When sequence B is used, only 3 per cent of the CPU time is saved, because of the time spent in modelling closed loops.

4.1.2 The TRANSYT/6 User Guide Example: This example is a very small network with several shared stoplines and other special features. The computer time saved is 3 per cent using sequence A. However, sequence B takes 3 per cent longer than TRANSYT/6.

4.1.3 San Jose: This network was used to compare TRANSYT predictions with those of SIGOP[5] in 1972. San Jose is a typical American city with a grid of mostly one-way streets, running at a very short cycle time (45 seconds). The network contains 21 closed loops, but the PI obtained by TRANSYT/7 is almost identical with TRANSYT/6. The reduction in central processor time is 71 per cent.

4.1.4 Belgrade: Although this network is approximately the same size as San Jose it contains only two closed loops. Using sequence A produces a time saving of 72 per cent, similar to San Jose. However, sequence B is comparatively much slower for San Jose than it is for Belgrade. This is a consequence of the extra iteration round closed loops in San Jose.

4.1.5 The Bitterne Road: The Bitterne Road network in Southampton is essentially an arterial and is part of a metering system designed to relocate queues and to give priority to buses. TRANSYT/6 was used to determine suitable offsets on the main road. Green durations were regarded as fixed, so the accuracy parameter sequence A used with TRANSYT/7 was that given by Sequence 2. The behaviour of buses was modelled in detail, with four separate categories of buses at some stoplines.

The network has no closed loops, but because of the large number of shared stoplines, the time saved by using TRANSYT/7 is 25 per cent. The signal settings obtained are identical with TRANSYT/6.

4.2 *Differences in performance index and signal settings*

It was stated in Section 4.1 that the PIs obtained by TRANSYT/7 may not be identical with those of TRANSYT/6, although the maximum difference is 0.3 per cent. Correspondingly, different signal settings may also be obtained.

There are two main reasons for these differences. The first is that the TRANSYT optimiser sometimes fails to reach a global optimum after the sequence of hill-climb steps and settles in a local minimum. Because TRANSYT/7 does not necessarily maintain the same level of accuracy as TRANSYT/6, it may settle in a different local minimum, with a slightly different PI. A similar effect can occur in TRANSYT/6 by changing the initial settings. Differences in the final PI are likely to increase with network size.

The second reason for the difference in minimum is the fact that TRANSYT/7 models closed loops more accurately. Given a set of signal timings for a network with closed loops, the initial PI will in general not be identical with that of TRANSYT/6.

Of the networks cited in Table 1, Glasgow SA4, the Glasgow research area and the Bitterne Road have no closed loops. The optimised settings obtained for SA4 and the Bitterne Road are the same for TRANSYT/6 and 7. For the Glasgow research area SA5, signal settings differ by 4 seconds at one junction and by up to 2 seconds at a further 9 of the 45 junctions. The corresponding performance index from TRANSYT/7 is 0.1 per cent larger.

Those networks in Table 1 which have closed loops give slightly different PIs for the same signal settings and are difficult to compare directly. Consider the case of SA1. The signal settings obtained as optimum by TRANSYT/7 were used with the TRANSYT/6 traffic model, whilst those from TRANSYT/6 were used in TRANSYT/7. The PIs are shown in Table 3. In this example the TRANSYT/7 optimised settings are seen as better by both versions of TRANSYT.

TABLE 3

A comparison of signal settings obtained by TRANSYT/6 and 7

| | TRANSYT/6 model | TRANSYT/7 model |
|---|---|---|
| TRANSYT/6 optimum settings | 1546.7 | 1549.0 |
| TRANSYT/7 optimum settings | 1546.5 | 1543.8 |

The small differences in PI show that the optimum settings obtained by TRANSYT/6 and 7 are equally valid. Such differences are unlikely to be of any practical significance.

### 4.3 *General comments*

Users of TRANSYT/7 should note the following points:

The signal settings and PI may not be identical with those of TRANSYT/6 but the differences in PI should be of no practical significance, and the settings equally valid.

Changing the accuracy parameter sequence may also lead to a slightly different PI.

Large reductions in computer time can be obtained for large networks, but there will be less time saved for small networks.

Less time is saved in networks which contain closed loops, but model accuracy is improved.

Less time is saved in networks which contain a large number of shared stoplines.

## 5. CONCLUSIONS

The use of version 7 of TRANSYT rather than earlier versions, leads to considerable reductions in computer time for most types of signal networks. For example, in a signal network of 29 junctions and 138 links, version 7 takes only 34 per cent of the computer time required by earlier versions. The savings in computer time are greater in larger networks. This improvement is gained without loss of accuracy although the program user can achieve further reductions in computer time at the expense of some increase in traffic delays and stops.

The most difficult part of preparing data for earlier versions of the TRANSYT program was the ordering of the link list. This task is accomplished automatically in version 7 of TRANSYT.

## 6. ACKNOWLEDGEMENTS

The work described in this report was carried out in the Urban Networks Division of the Traffic Engineering Department of TRRL. Particular thanks are due to Mr D I Robertson and Mr P Gower.

## 7. REFERENCES

1. HOLROYD, J and J A HILLIER. The Glasgow Experiment: PLIDENT and after. *Department of the Environment, RRL Report* LR 384. Crowthorne, 1971 (Road Research Laboratory).

2. ANON. SIGOP traffic signal optimisation program users manual. New York, 1968 (Peat, Marwick, Livingston & Co).

3. ROBERTSON, D I. TRANSYT: a traffic network study tool. *Ministry of Transport, RRL Report* LR 253. Crowthorne, 1969 (Road Research Laboratory).

4. ROBERTSON, D I and P GOWER. User Guide to TRANSYT Version 6. *Department of the Environment, TRRL Report* SR 255. Crowthorne, 1976 (Transport and Road Research Laboratory).

5. WAGNER, F A. SIGOP–TRANSYT evaluation: San Jose, California. Alan M Voorhees and Associates. Los Angeles DOT–FH–11–7822, June 1972.

6. PIERCE, J R and K WOOD. BUS TRANSYT – a user's guide. *Department of the Environment Department of Transport, TRRL Report* SR 266. Crowthorne, 1977 (Transport and Road Research Laboratory).

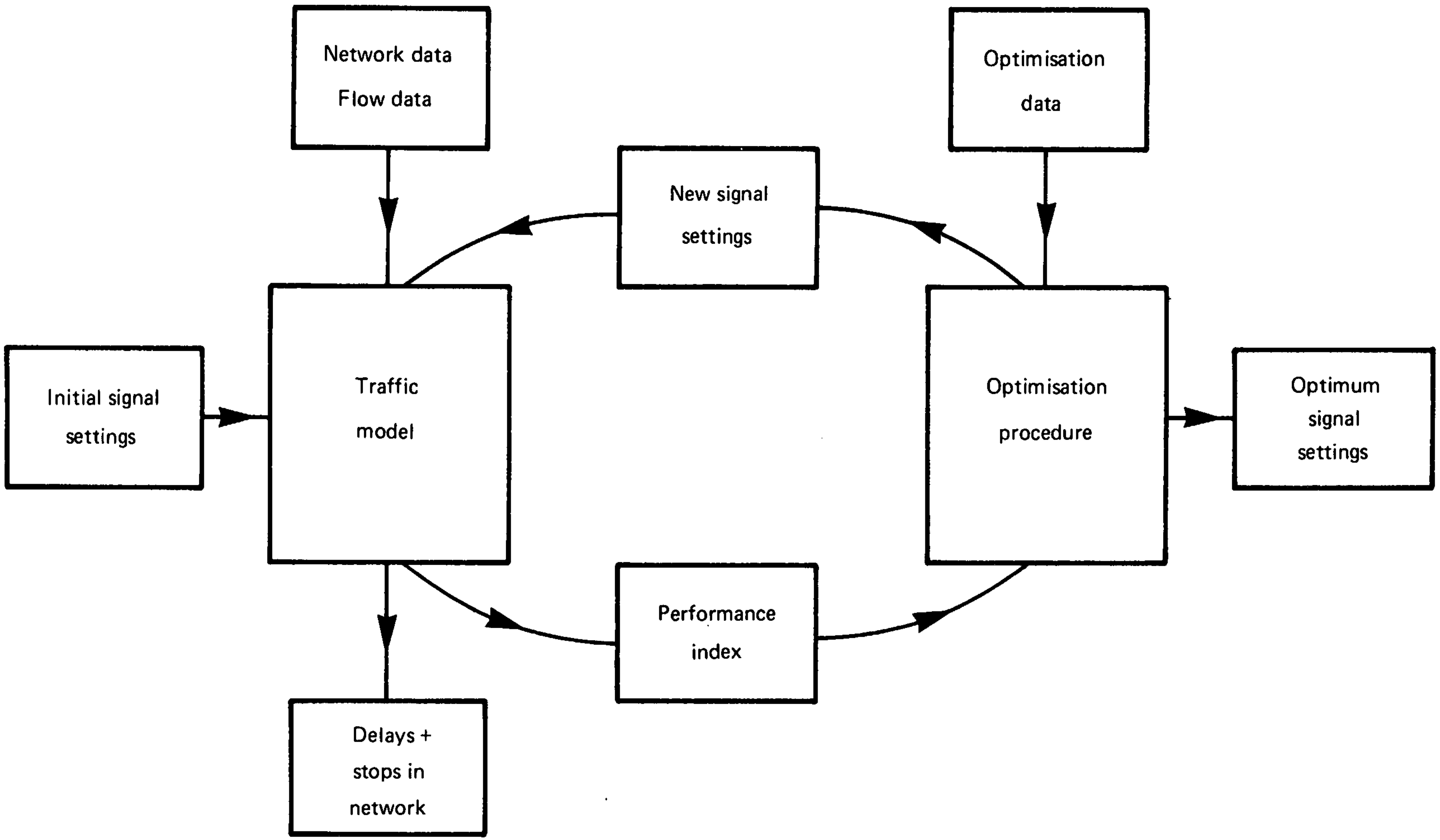


Fig. 1 THE STRUCTURE OF THE TRANSYT PROGRAM

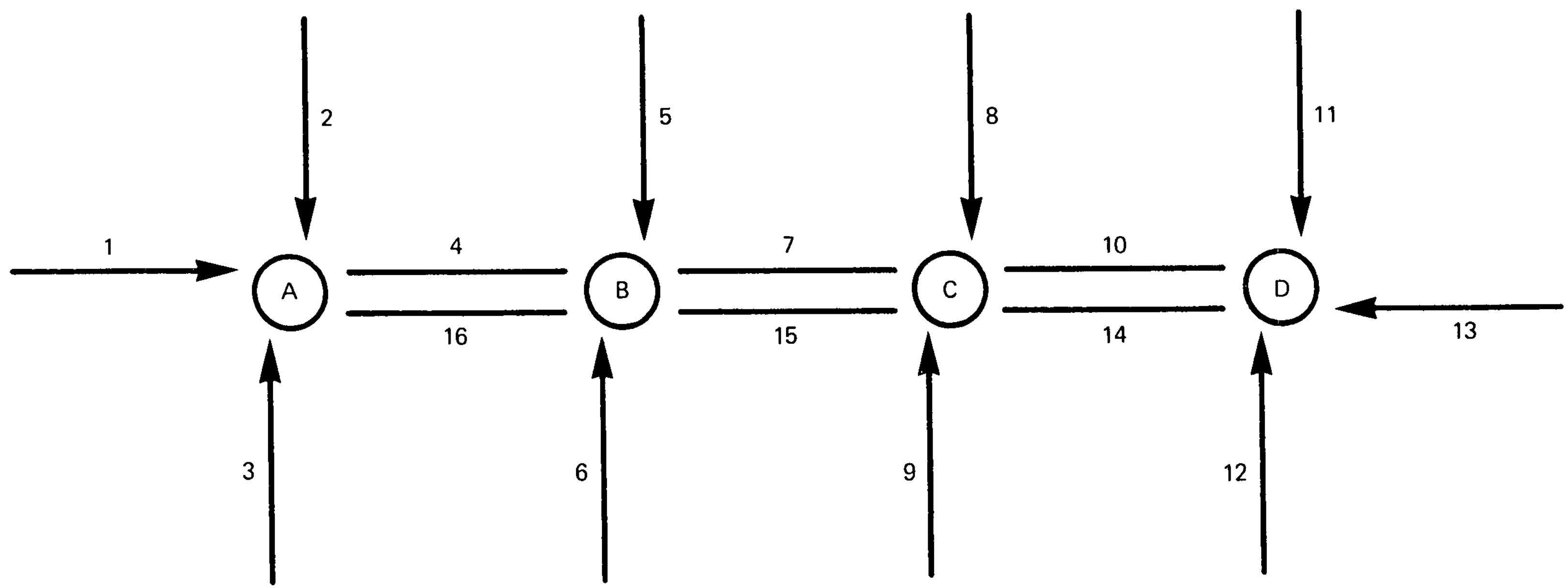


Fig. 2 AN ARTERIAL ROAD

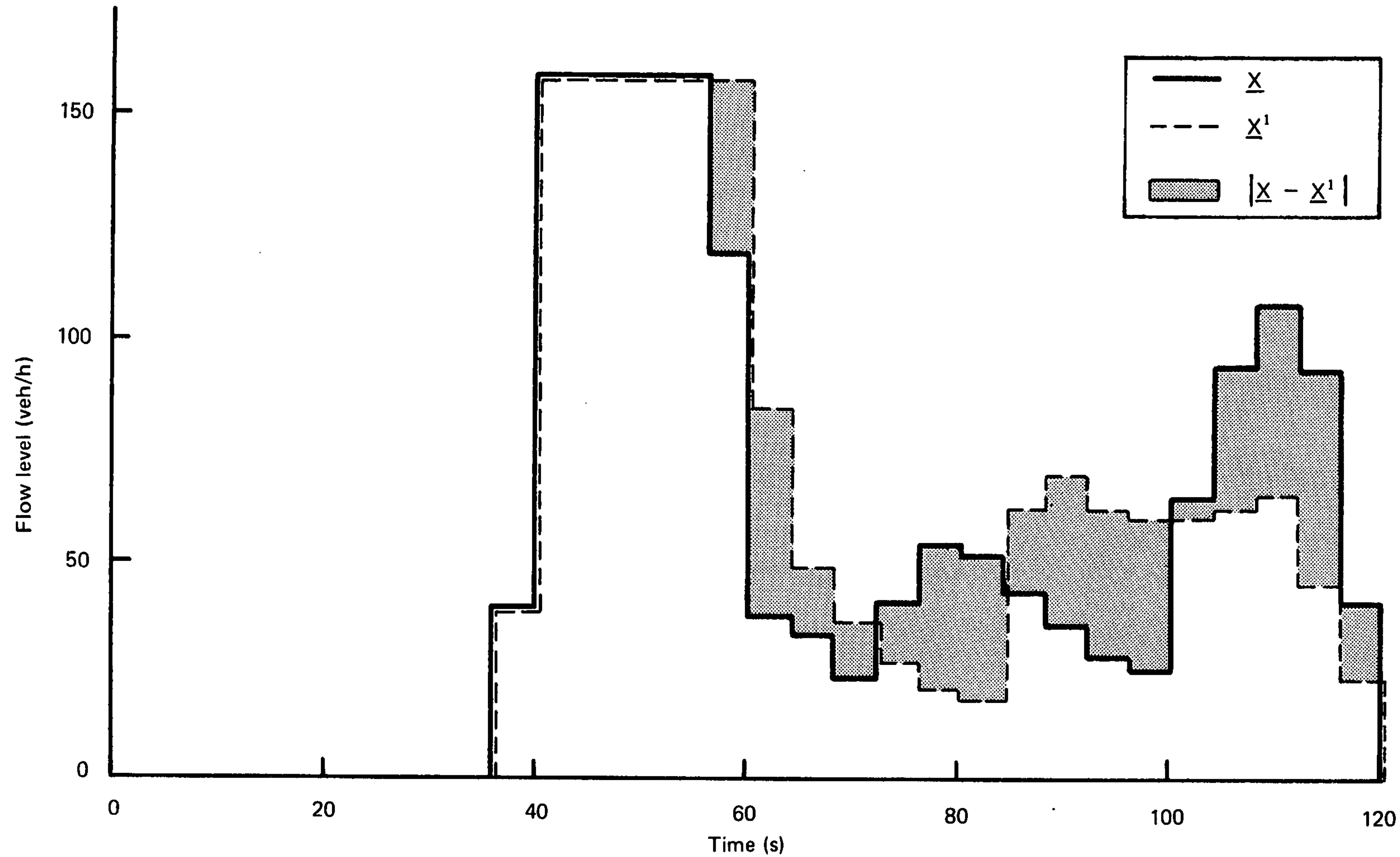


Fig. 3 FLOW PROFILES $\underline{X}$ AND $\underline{X}^1$ LEAVING A STOP LINE BEFORE AND AFTER A CHANGE IN SIGNAL SETTINGS AT AN UPSTREAM JUNCTION

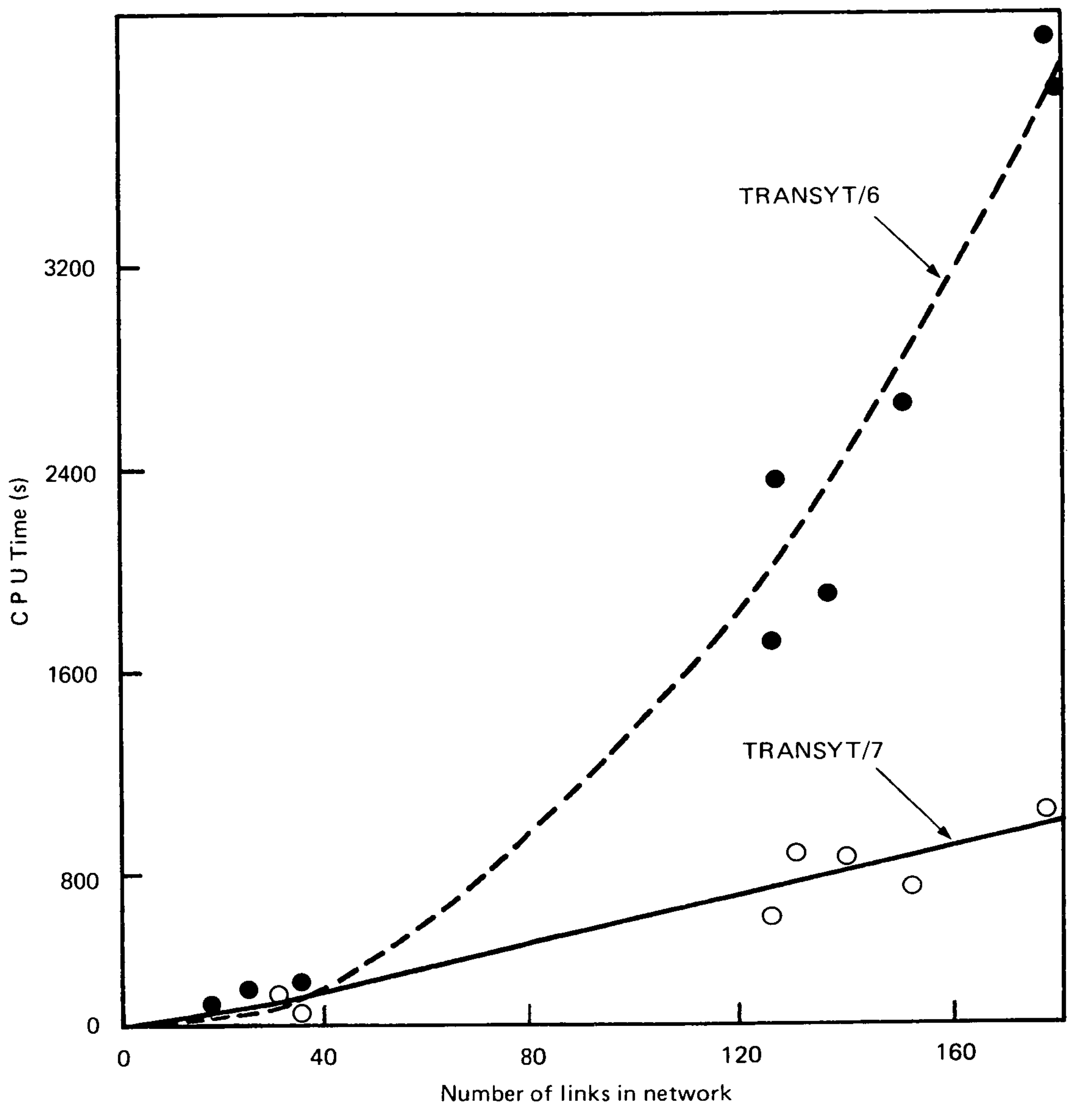


Fig. 4 VARIATION IN COMPUTER TIME WITH NETWORK SIZE

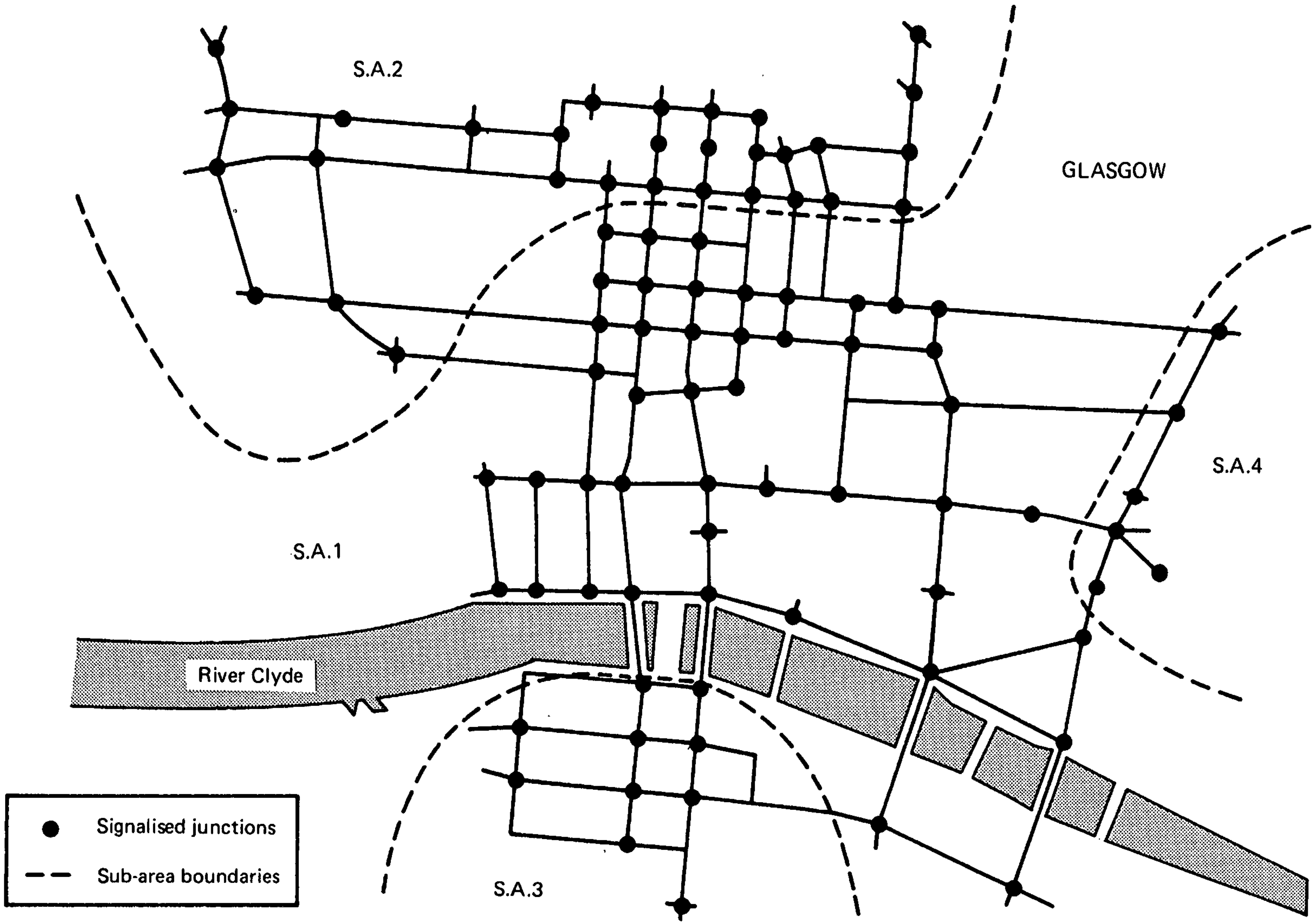


Fig. 5 THE GLASGOW NETWORKS

# 8. APPENDIX

## How to convert TRANSYT/6 data to TRANSYT/7

These notes should be read in conjunction with the TRANSYT/6 User Guide[4].

I Card type 1

The entry 1 in column 40 on card type 1 will now calculate 'STAR 1' timings for all nodes listed on card type 2. These are the nodes which are to be optimised; the timings of other nodes will be unchanged. This feature enables the user to calculate signal timings for one sub-area and then transfer these timings to an overlapping sub-area[6], provided both areas have a common cycle time.

II Card type 3 (obsolete)

There is no longer any need for a link list to be submitted on card type 3. If cards type 3 are present, they will be ignored.

III Card type 4 (obligatory for optimisation)

A type 4 card can be omitted when no optimisation is required.

IV Card type 6 (optional)

A card type 6 which controls the degree of accuracy in the optimisation procedure can be added if desired. The default values used when this card is omitted lead to accurate signal timings but do not give the maximum time saving. Section 8.1 below describes card type 6 in detail.

V Card type 7 (obligatory where there are shared stoplines)

When shared stoplines occur in a network, they must be specified using cards type 7. Section 8.2 describes card type 7 in detail.

VI Card type 36 (replaces card type 30 in TRANSYT/6)

If a card type 30 is to be used, it must be replaced by a card type 36. Where a card type 30 occurs in the input data, it will be ignored and a warning message given. The format is identical to card type 30.

VII Dummy links (optional)

There is no longer any need to break closed loops by introducing dummy links. However, data sets which have already been used with TRANSYT/6 may retain their dummy links for simplicity. If dummy links are removed, the traffic model will be slightly more accurate. Any negative links will be omitted from the calculation of the performance index.

## VIII Bottlenecks (optional)

The facility to model bottlenecks is now available. The bottleneck is treated as a junction with 100 per cent green at which a fixed saturation flow is defined. Links which have a bottleneck for their exit node are treated in the same way as any other link. However, the exit node number on type 31 cards (column 15) is always set to zero and no stage times are entered, that is, columns 20–55 inclusive are left blank. Shared links which are bottlenecks are specified in cards type 7, as for signalised links, but the 'main' link still has a zero exit node on card type 31.

Delays and stops are calculated for bottleneck links exactly as for other links, except that the flow leaving the link is controlled only by the saturation flow.

The total number of nodes must not exceed 49 when bottlenecks are being modelled.

## IX Changes to the output

The heading 'maximum uniform queue' in TRANSYT/6 gives the average value of the maximum number of vehicles queueing which is calculated to occur during the signal cycle. Its value is calculated assuming that the front of the queue is always at the stopline. TRANSYT/7 prints under the heading 'maximum back of queue' a value which takes some account of the position of the back of the queue on the street. The new value will still underestimate the maximum queue because random effects are excluded.

### 8.1 *Card type 6*

This card enables the user to limit the accuracy with which the hill-climb proceeds for each optimisation increment specified on card type 4. The value given on card type 6 lies in the range 1–2000 and represents 100 times the percentage change in the flow profile leaving the stopline of a link below which downstream effects are to be ignored.

The default value for each increment if card type 6 is omitted is 1, corresponding to a change in the flow profile of 0.01 per cent. This is the most accurate value allowed by the program.

However, further computer time can be saved by the following procedure. It is not necessary to calculate the flow profiles very accurately whilst the increment is large. Thus, small changes in the flow profile on a link can be ignored until the signal settings are close to optimum. For the usual card type 4 sequence, quoted in the TRANSYT/6 User Guide[4], which optimises offsets over a 50 step cycle:

7 20 7 20 7 1 1 . . . . . . . . . . Sequence 1

the recommended accuracy parameter list is

1000 1000 100 100 10 1 1 . . . . . . . . . . Sequence 2

Similarly, the list corresponding to the usual card type 4 splits and offsets optimisation sequence:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 20 | −1 | 7 | 20 | 1 | −1 | 1 . . . . . | Sequence 3 |

is

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1000 | 1000 | 100 | 100 | 10 | 10 | 1 | 1 . . . . . | Sequence 4 |

These sequences will not necessarily lead to the same optimum settings as TRANSYT/6 or as other accuracy parameter lists with TRANSYT/7. The differences in performance index should be trivial.

The maximum accuracy parameter value permitted is 2000. Values greater than 100 are not recommended for the calculation of final signal settings, except to find a first approximation. The initial signal settings are always calculated with an accuracy parameter value of one. Thus, omitting cards type 4 and 6 will continue to ensure greatest accuracy for the model when there is no optimisation.

The card format is:

| Column | Entry |
|---|---|
| 5 | 6 |
| 10,15 etc | The chosen accuracy parameter sequence |

There is only one card type 6. The number of entries should correspond to the number on card type 4. Blanks are permitted on both cards type 4 and 6 and will be ignored. Thus, the $n^{th}$ non-zero entry on card type 6 will be treated as the accuracy parameter for the $n^{th}$ non-zero entry on card type 4.

8.2 *Card type 7*

This card is used to specify up to three sets of shared stoplines. There may be from two up to five links within a set, which discharge queues over a single shared stopline. Queues are discharged in the order in which arrivals occur on the links, representing the intermingled queues of different types of vehicles.

Links within a set of shared stoplines may be listed in any order, but the first link is treated as the main one and the saturation flow across the stopline is listed opposite this link number on the final output. For example, when modelling buses separately, the main link could be the one carrying all other traffic.

In earlier versions of TRANSYT, links other than the main one in the set have been given a zero value on card type 31 for the downstream node number. This is no longer necessary, but it will be ignored where it occurs, for example, on existing data.

The format is:

| Column | Entry |
|---|---|
| 5 | 7 |
| 10 | 'main' link in set of shared stoplines (non-zero outnode on card type 31) |
| 15,20,25,30 | Other links sharing this stopline |
| 35–55 | As 10 to 30 |
| 60–80 | As 10 to 30 |

(1314) Dd0536380 1,600 7/80 HPLtd So'ton G1915
PRINTED IN ENGLAND

ABSTRACT

A GUIDE TO TRANSYT/7: *P B Hunt and J V Kennedy:* Department of the Environment Department of Transport, TRRL Supplementary Report 595: Crowthorne, 1980 (Transport and Road Research Laboratory). TRANSYT/7 is a new version of a computer program which calculates optimum fixed time settings for traffic signals in a network. It is much faster than previous versions for large networks. The computer time required is now proportional to the number of links in the network rather than the square of the number of links. Evidence for this was obtained from ten networks of differing size and character.

Data input to the program has been simplified and the changes are detailed in an appendix. Links can be listed in any order and closed loops of links are identified by the program and modelled by an iterative method to ensure that accurate values of the delay and queue length are obtained for all links in the loop. The modelling of bottlenecks has been included in TRANSYT/7.

ISSN 0305–1315

ABSTRACT

A GUIDE TO TRANSYT/7: *P B Hunt and J V Kennedy:* Department of the Environment Department of Transport, TRRL Supplementary Report 595: Crowthorne, 1980 (Transport and Road Research Laboratory). TRANSYT/7 is a new version of a computer program which calculates optimum fixed time settings for traffic signals in a network. It is much faster than previous versions for large networks. The computer time required is now proportional to the number of links in the network rather than the square of the number of links. Evidence for this was obtained from ten networks of differing size and character.

Data input to the program has been simplified and the changes are detailed in an appendix. Links can be listed in any order and closed loops of links are identified by the program and modelled by an iterative method to ensure that accurate values of the delay and queue length are obtained for all links in the loop. The modelling of bottlenecks has been included in TRANSYT/7.

ISSN 0305–1315