AN OPTIMISATION METHOD FOR SIGNAL TIMING IN AREA TRAFFIC CONTROL

Kiseok Sung
Associate professor
Industrial Engineering Department
Kangnung National University,
Kangnung 210-702, Korea.
Phone : +82 391 640 2371
Facsimile : +82 391 640 2244
e-mail : SUNG@knusun.kangnung.ac.kr

Michael G.H. Bell
Professor
Transport Operations Research Group,
University of Newcastle Upon Tyne,
Newcastle Upon Tyne, NE1 7RU, UK
Phone : +44 191 222 7939
Facsimile : +44 191 222 8352
e-mail: M.G.H.Bell@ncl.ac.uk

abstract : In this paper an optimisation method to find a good assignment of signal control stages to links in the network (the link - stage assignment) for TRANSYT is described. In the method, a special genetic algorithm is involved. In the genetic process, the partially matched crossover operator and exchange mutation operator are used to maintain the feasibility of the sequence of phases. A special conversion algorithm is devised to convert the phase sequences into the link-stage assignment. Results from tests using data from a SCOOT region in Leicester are given.

1. INTRODUCTION

Allsop (1992) reviewed the terms signal group, phase, stage, the compatibility of phases and the constraints on signal timings. For a full understanding of this paper it is first necessary to define some terms used in traffic signal control. Traffic (vehicular or pedestrian) at a junction is considered in terms of *streams*. For vehicles, a stream is defined as a flow of traffic which can be regarded as forming a single queue. It may cover one or more adjacent lanes. A *phase* is a sequence of signal conditions given to one or more streams so that each stream receives identical signal indications. A *stage* is part of a signal cycle during which a particular set of phases receives green. One complete sequence of stages is called a *cycle*.

The calculation of signal timings for an isolated intersection is easier than for a network of closely-spaced junctions because in the latter case co-ordination of the signals becomes advantageous to allow for interaction between traffic at adjacent junctions. The distance between neighbouring intersections is frequently too small for platoons of traffic released by one intersection to disperse completely before arrival at the next intersection.

The calculation of optimal signal co-ordination plans is not an easy task, especially for closed-loop network topologies such as grids, so many computer methods have been developed. TRANSYT (**TRA**ffic **N**etwork **S**tud**Y** **T**ool) is an off-line computer program for determining and studying optimum fixed-time, co-ordinated, traffic signal timings in any network of roads for which the average traffic flows are known (Vincent et al, 1980). It is the most widely-used program of its type in the world.

The internal traffic model of TRANSYT calculates a Performance Index (PI) for the network which is a weighted sum of all vehicle delays and stops. TRANSYT searches for its optimum settings by a 'hill climbing' process. The optimiser systematically alters signal offsets and/or allocation of green times to search for the timings which reduce the PI to a minimum value, subject to minimum green and other constraints. There is a common cycle time, calculated to ensure that maximum degree of saturation is less than 90%. TRANSYT is stage-based and only optimises for one particular stage sequence for each junction. The later versions of the TRANSYT package include a support program, STAGOR, to investigate potential benefits of revised stage order (Crabtree, 1988).

Allsop (1992) outlined progress on mathematical optimisation methods for stage-based and phase-based design. As the design progressed from stage-based to phase-based control, it was shown that more efficient signal settings became available with more flexibility, but at the cost of increasing complexity. SIGSIGN (Silcock and Sang, 1990) and LINSIG II (Simmonite, 1994) are examples of phase-based optimisation programs for individual signal-controlled junctions.

W suggest a method of finding a good Link-Stage assignment for TRANSYT using Genetic Algorithm (GA) (Goldberg, 1989). GA imitates the mechanism of natural selection in genetics. It represents every structure with strings of digits similar to the genetic structure of a chromosome and it searches for a superior group of structures in a manor similar to evolution in nature. The differences of GA from other deterministic optimisation techniques are that it searches a population of strings, not a single string, that gives a near optimum of the objective value and that it uses probabilistic transition rules. The elementary operators of GA are reproduction, crossover and mutation.

We represent a sequence of signal phases with a string of ordered number and apply a special crossover operation to the string so that they can maintain the feasibility of phase sequence. For each sequence of signal phase, we use an algorithm to convert the phase sequences into link-stage assignments

2. GENETIC ALGORITHM IMPLEMENTATION

The phase sequences for each junction, represented by a string of ordered numbers, were concatenated together to form the chromosome. TRANSYT was used to determine the fitness of each chromosome.

Crossover: The representation chosen caused some difficulties when applying the general crossover operator during the GA process. To avoid infeasible offspring phase sequences being generated, a special crossover operator, known as Partially Matched Crossover (PMX), was used on the phase sequence representations so that the feasibility of phase sequence was maintained. Figure 1 shows an example of this operator working on a phase sequence for one junction. The position of the crossover points are marked by the solid vertical line (|). The phases following the crossover have to maintain the rule that each phase should appear once and only once in each feasible sequence.

Mutation: The general mutation operator might also have led to similar problems of infeasible phase sequences, so exchange mutation was used instead. Figure 2 shows the example of exchange mutation.

A | D B | E C F → A | E F | D C B
C | E F | D B A → C | D B | E F A

Figure 1. An example of Partially Matched Crossover

A D B E C F → A C B E D F
  ^   ^

Figure 2. An example of Exchange mutation

Reproduction: The fitness of the phase sequences was obtained by inverting the PI obtained from TRANSYT because the GA searched for the maximum fitness and the optimum PI is the minimum one found. The values of PI obtained from TRANSYT for the test network were in the order 400 to 4000. Inverting these will lead to very similar (small) numbers. In order to increase the efficiency of the reproduction process it has been suggested that linear scaling should be used on the raw fitness values (Goldberg, 1989). Figure 3 shows linear scaling of the fitness. The recommended value of the scaling factor, m, is between 1.5 and 2.0.

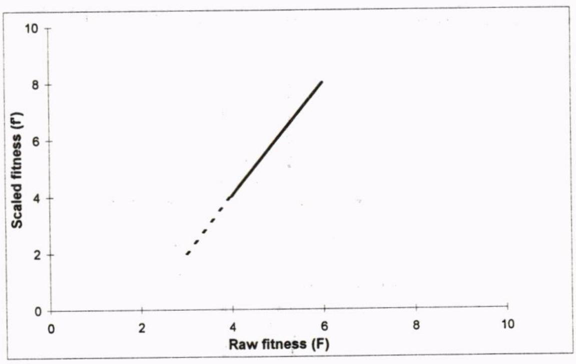

Figure 3. Linear fitness scaling

The average of the scaled fitnesses is set equal to the average of the raw fitnesses and the maximum of the scaled fitnesses is set equal to m times the average of the raw fitnesses (in the example above, m = 2). A straight line is drawn between the two points and the value of scaled fitness corresponding to each raw fitness then determined by interpolation or extrapolation of this line as appropriate.

3. CONVERSION OF A PHASE SEQUENCE INTO LINK-STAGE ASSIGNMENT

As discussed in the Introduction, TRANSYT is a stage-based, not a phase-based procedure, and furthermore requires link-stage assignment as input data. The green time for each link is determined by specifying one stage to be the start stage and another stage to be the end stage. The algorithm by which a phase sequence was converted into the necessary link-stage assignment for TRANSYT is shown in Algorithm 1.

Algorithm 1: To convert a given phase sequence into a link-stage assignment

Step 1: For each phase in turn (the reference phase), find the first incompatible succeeding phase and set the end-stage of the reference phase and the start-stage of the found phase to the order of the found phase.

Step 2: For each phase not yet assigned to any start-stage (the reference phase), find the first incompatible preceding phase and set the start-stage of the reference phase to the end-stage of the found phase.

Step 3: Concentrate stages by deleting the stages not assigned to any phase.

To illustrate the algorithm, consider the junction shown in Figure 4. Table 1 shows the associated phase incompatibility matrix. In the matrix a "1" signifies incompatibility between the phases whereas a space indicates compatibility. For example, phases A and D cannot be switched together, but phases A and B can. Given a phase sequence A-D-B-E-C-F, algorithm 1 will produce the stage sequence shown in Table 2. After combining stages 2 and 3, the link-stage assignment that results is shown in Table 3.

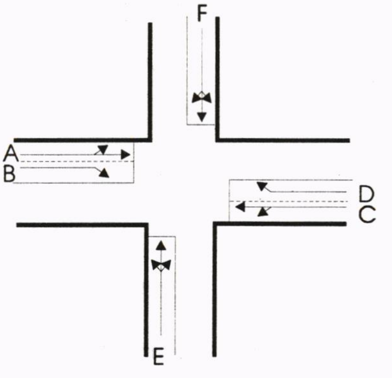

Figure 4. Example junction

Table 1. Phase incompatibility matrix

	A	B	C	D	E	F
A				1	1	1
B			1		1	1
C		1			1	1
D	1				1	1
E	1	1	1	1		
F	1	1	1	1		

Table 2: Stage sequence

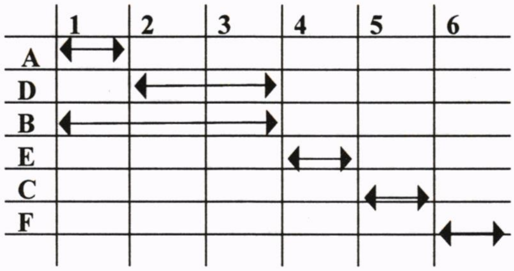

Table 3: Stage assignment converted from phase sequence of A-D-B-E-C-F

Phase	Start-stage	End-stage
A	1	2
B	1	3
C	4	5
D	2	3
E	3	4
F	5	1

4. OVERALL PROCEDURE

Algorithm 2 finds a near-optimal solution to the phase-based area traffic control problem by combining the GA and TRANSYT.

```
Algorithm 2: To find solution to phase-based area traffic control

/* Initializing */
Set parameters for GA;
Generate initial random population of phase sequences;
Repeat
    /* Generate new population of phase sequences */
    Repeat
        Select two sequences as parents;
        Crossover and generate two offspring sequences;
        /* Evaluate offspring sequences */
        Convert the sequences into link-stage assignment;
        Get the PI of the link-stage assignment by TRANSYT;
    Until the new population is complete;
    Reproduce population with PI of each sequence;
    Accumulate statistics of new population;
Until statistics satisfy certain criterion;
Report the link-stage assignment that gives minimum PI ever generated;
End;
```

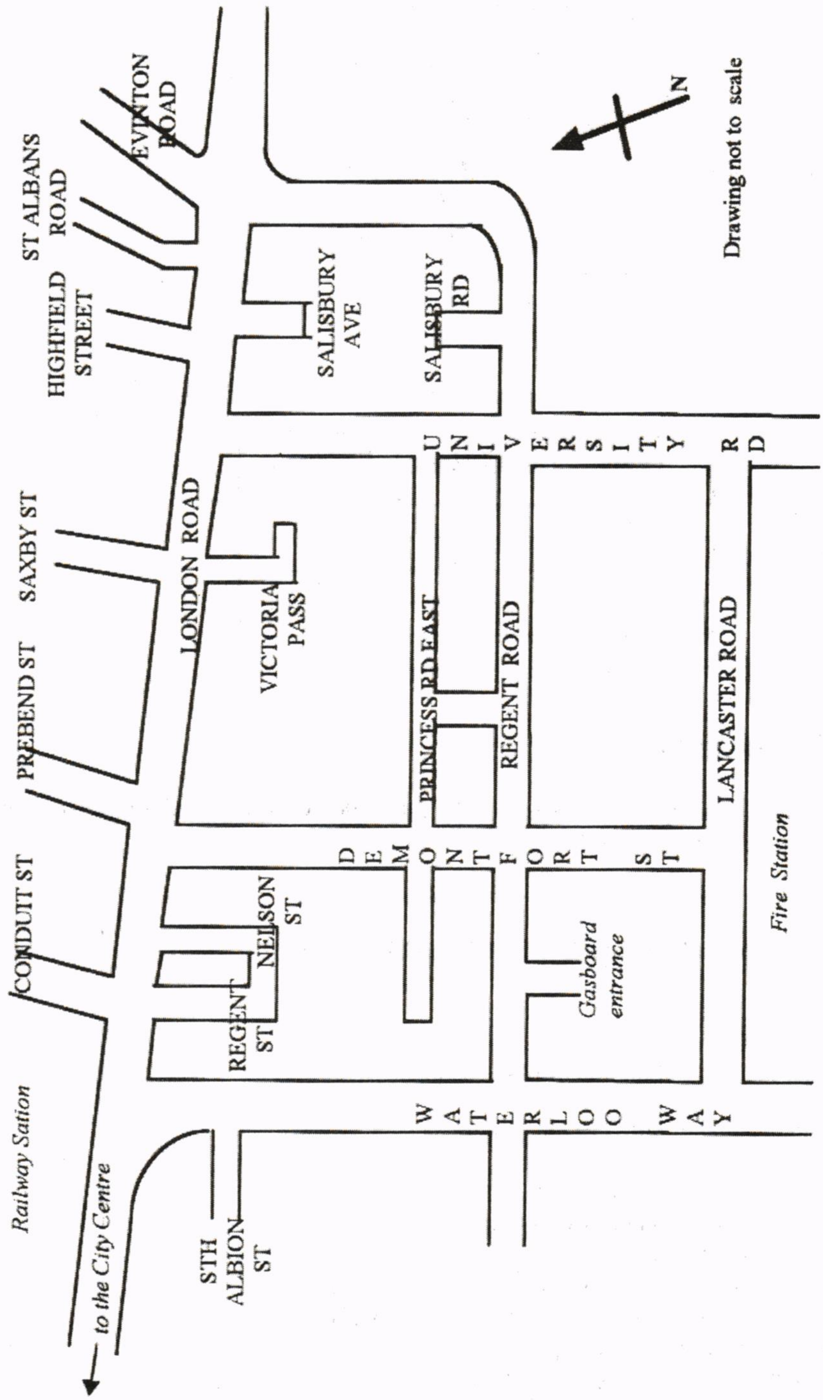

Figure 5 Road Layout – Leicester Region R

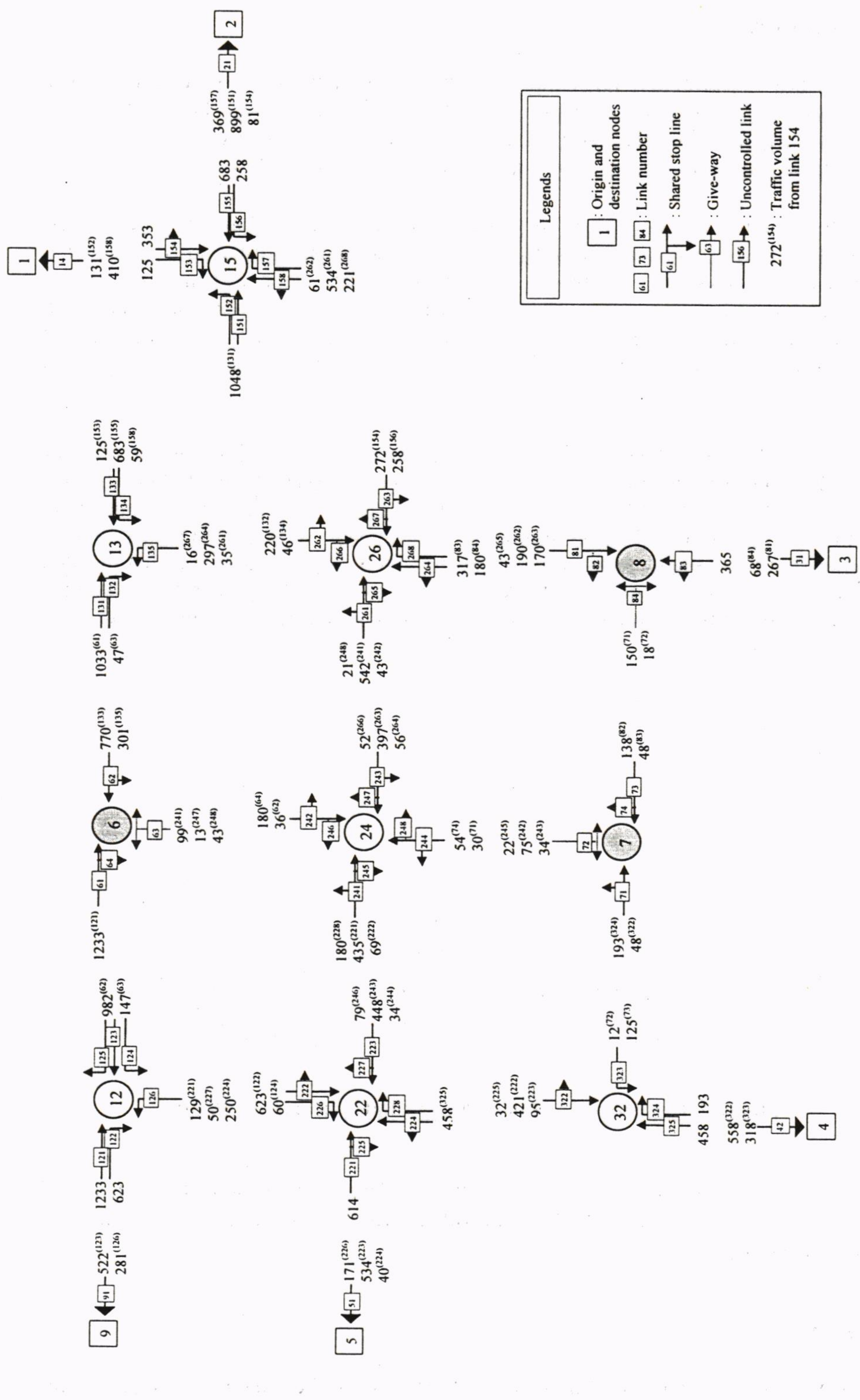

Figure 6. Test network - Leicester Region R

5. SAMPLE RUN

In order to test the proposed procedure, data from region R of the Leicester SCOOT network were used. This was because all the data required to run TRANSYT were available from registration plate surveys carried out by the University of Nottingham's Transport Research Group (Evans, 1992) for the Instrumented City (IC).

The layout of the test region is given in Figure 5. The network consisted of seven signal-controlled junctions and three uncontrolled junctions. A schematic of the network, showing node (junction) numbers, link (or phase) numbers and traffic volume of links is given in Figure 6. The uncontrolled junctions are distinguished by shading of the node. Since each signal-controlled junction has between four and eight independent links the number of possible phase sequence combinations is very large ($\sim 10^{16}$), hence the need for a search technique.

Figure 7 shows the convergence of PI in Genetic progress. The solid line and the fine line in the graph represent the minimum and the average PI respectively. In the run, the parameters for genetic algorithm are given as 20, 0.2 and 1.5 for the size of population, probability of mutation and scale multiplier respectively. When TRANSYT run as subprogram, the type (single cycle or double cycle) and duration of cycle at each junction may be given by input or adjusted by TRANSYT. Figure 7-a and Figure 7-b represent the result of each case respectively. Single 180 seconds cycle time is given for the former case.

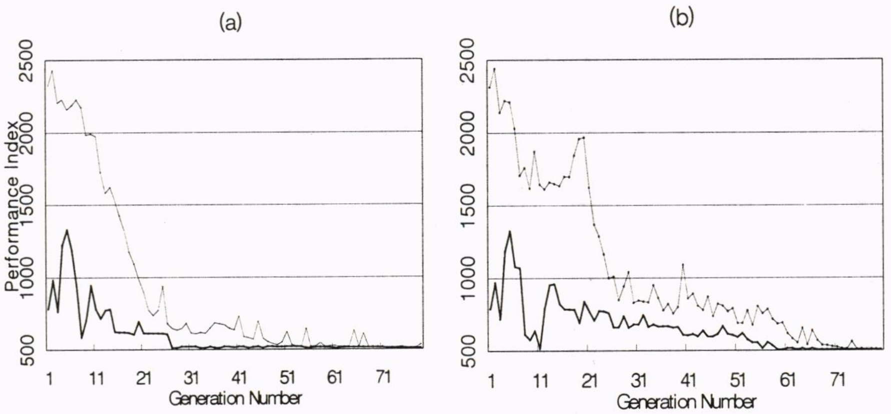

Figure 7. Convergence of Performance Index in Genetic Progress.

The result of sample run shows that the PI of TRANSYT are 509.0 and 841.4 with a good link-stage assignment determined by GA and with an arbitrary link-stage assignment given by input respectively.

6. SUMMARY

In this paper a genetic algorithm finding a good solution for the link-stage assignment for

TRANSYT is described. Results of initial tests on a network in Leicester are promising. Further researches are to be made to determine a good type and duration of cycle at each junction by GA.

ACKNOWLEDGEMENTS

This work carried out by the first author during his British Council funded visit to the Transport Operations Research Group in University of Newcastle upon Tyne.

REFERENCES

Allsop, R. E. (1992) Evolving application of mathematical optimisation in design and operation of individual signal-controlled road junctions, *Mathematics in Transport Planning and Control*, J.D.Griffiths (Ed.), pp 1~24, Clarendon Press, Oxford.

Crabtree, M. R. (1988) TRANSYT/9 Users Manual, *TRRL Application Guide 8*, Transport and Road Research Laboratory, Crowthorne, Berkshire.

Evans, R. G. (1992) A description of the data obtained from three traffic surveys in Leicester, *NUTRG Technical Report 92045*, Nottingham University Transport Research Group.

Goldberg, D. E.(Ed.) (1989) *Genetic Algorithms in search, optimisation and machine learning*, Addison-Wesley, Massachusetts.

Silcock, J. P. and A. Sang (1990) SIGSIGN : A phase-based optimisation program for individual signal-controlled junctions, *Traffic Engineering and Control*, **31(5)**, May, pp 291~298.

Simmonite, B. F. (1994) LINSIGII : The next generation. A computer program to aid traffic signal design and assessment, *Traffic Engineering and Control*, **35(4)**, April, pp 236~242.

Vincent, R. A., A. I .Mitchell and D. I. Robertson (1980) User guide to TRANSYT version 8, *TRRL Laboratory Report 888*, Transport and Road Research Laboratory, Crowthorne, Berkshire.