Optimising cycle times of controlled intersections with VRIGen

Dr. Ir. A.M. Salomons
Technische Universiteit Delft, Civiele Techniek en Geowetenschappen,
afd. Transport en Planning
a.m.salomons@tudelft.nl

Bijdrage aan het Colloquium Vervoersplanologisch Speurwerk
20 en 21 november 2008, Santpoort

Optimalisering van cyclustijden van kruispunten met VRIGen

"Vroeger was de toekomst beter", maar voor vroeger en nu geldt dat om kruispunten te ontwerpen die veilig zijn en geen wachtrijen genereren, de wachttijd van het verkeer op alle verkeersstromen zo minimaal mogelijk moet zijn. Toen F.V. Webster zijn formule voor het optimaliseren van de minimum cyclustijd van een door verkeerslichten geregelde kruising opstelde (in 1966), stond het gebruik van computers nog in de kinderschoenen. Natuurlijk was de verwachting dat met gebruik van computertechnologie het optimaliseren van wachttijden eenvoudiger, beter en sneller zou worden. Hoe is de toekomstverwachting van Webster ingevuld?
Om de optimale structuur te bepalen en de minimum cyclustijd en groentijden voor alle verkeersstromen is het programma VRIGen (VRI Generator) ontwikkeld aan de Technische Universiteit Delft.
VRIGen genereert alle mogelijke structuren aan de hand van de onderlinge conflicten van de verkeersstromen en de koppelingen tussen stromen op het kruispunt. Bij deze structuren hoort een minimum cyclustijd, die samenhangt met de intensiteit, de capaciteit en de onderlinge ontruimingstijden van de verkeersstromen. VRIGen bepaalde de minimum cyclustijd met de cyclustijd van de kritieke conflictgroep T_{Crit}. Als de structuur niet binnen de kritieke conflictgroep cyclustijd T_{Crit} afgehandeld kon worden, bepaalde VRIGen iteratief de juiste minimum cyclustijd. Dit was een tijdrovende benadering, speciaal in situaties waarin veel structuren zijn gegenereerd. Het blijkt dat het bepalen van de minimum cyclustijd sneller kan met behulp van het kritieke pad. Dit kritieke pad wordt bepaald door de groentijden behorend bij T_{Crit} in de structuur in te vullen en dan te bepalen welke stromen ervoor zorgen dat de conflictgroepen een grotere cyclustijd nodig hebben dan die van de kritieke conflictgroep. Ook blijkt dat de kritieke pad benadering gebruikt kan worden bij het bepalen van de maximum verlenggroen voor de verkeersstromen.
Voor verdere ontwikkeling van VRIGen moet nog onderzoek gedaan worden aan het zo snel mogelijk genereren van de meest optimale structuur: dit bepaalt nu de rekentijd. Verkeerslichtinstallaties uitgerust met software dat in seconden de meest optimale structuur gegenereert, die direct samenhangt met het verkeersaanbod van het moment zullen wachtrijen bij kruisingen nog verder kunnen terugdringen. De toekomst kan nog steeds beter.

1 Introduction

“The future was brighter in the past”, but for the past and the present applies that controlled intersections that are safe and will generate no queues, the wait time for road users should be as low as possible, for all traffic streams involved. When F.V. Webster designed the formula for the minimum cycle time for a controlled intersection [1], the use of computers was imminent. Of course there were high expectations of computer technology for the optimisation of the wait time. This article will show how these expectations of Webster are fulfilled nowadays.

To find the optimum structure, minimum total cycle time and green times for all traffic streams, the program VRIGen (Traffic Control Generator) has been developed at the University of Technology Delft [2]. VRIGen generates all possible control structures, using the mutual conflicts between traffic streams and the coupling between streams on the intersection. For all those structures the cycle time is computed, which depends on the flow and the capacity of the traffic streams and the clearance time between streams. Then VRIGen presents the user all structures that have less than the prescribed maximum cycle time, ordered by increasing cycle time. The structure description and tactics that are chosen by the user are saved in a control program such as CCol [3] or TrafCod [4].

In VRIGen, the cycle time was originaly determined as the minimum cycle time according to the critical conflict group (see Chapter 2). For those structures that cannot complete all traffic streams within this critical conflict group cycle time, the minimum cycle time was iteratively adjusted until the minimum cycle time was found in which all streams can be handled. To find the minimum cycle time via iteration is a very time consuming method, however, especially when VRIGen generates many structures and for every structure the cycle time needs to be determined in an iterative way. In this paper a new method is described, based on optimising the mutual start-off between the conflict groups and the critical path instead of the critical conflict group.

2 Computing the minimum cycle time of a structure

2.1 Critcal conflict group and critical path

VRIGen uses streams and their conflicts as input to generate structures. First the conflict groups are determined and then all possible structures are generated. VRIGen determines the structures by finding all possible rankings of the full conflict groups, the other conflicts group which have fewer streams than the full conflict groups are fit in the structure. Then VRIGen tries to fill in the empty stages in the structure.

If the structure is found, the cycle time for conflict group j is determined with Eq. 2.1

$$T_{Ccrit,j} = \frac{\sum_{i=1}^{M} T_L}{1 - \sum_{i=1}^{M} \frac{q_i}{s_i}} \qquad 2.1$$

where M is the number of streams in a conflict group, $\sum_i T_L$ is the summation over the "lost time" between successive streams in the conflict group (yellow time plus clearance times) and $\sum_i \frac{q_i}{s_i}$ the proportion flow q_i and capacity s_i.

The critical conflict group minimum cycle time is:

$$T_{CCrit} = \max_j \left(T_{Ccrit,j}\right) \qquad 2.2$$

The green time of stream i is given by:

$$T_{g,i} = \frac{q_i}{s_i} T_{C\min} \qquad 2.3$$

provided that the minimum green time $T_{g\min,i}$ is reached.

If the green time is smaller than the minimum green time, the minimum cycle time has to be recomputed. If the calculated green time for stream m does not reach the minimum, $T_{g\min,m}$ is added to the lost time and its flow q_m is removed from the summation:

$$T_{Ccrit,j} = \frac{\sum_{i=1}^{M} T_L + T_{g\min,m}}{1 - \sum_{i=1, i \neq m}^{M} \frac{q_i}{s_i}} \qquad 2.4$$

VRIGen also computes the Webster cycle time for fixed time control (Eq.2.5) [1]. This cycle time is much larger than the minimum cycle time.

$$T_{CWeb} = \frac{1.5\sum_{i=1}^{M} T_L + 5}{1 - \sum_{i=1}^{M} \frac{q_i}{s_i}} \qquad 2.5$$

The Webster formula was developed for intersections that cannot handle the traffic streams within the minimum cycle time, but is in most cases over dimensioned, but sometimes also under dimensioned, as will be shown in next examples.

As an example, the cycle time is given of an intersection that comprises of the streams 02, 03, 05, 06, 08, 09, 11 and 12, as shown in Fig. 2.1.

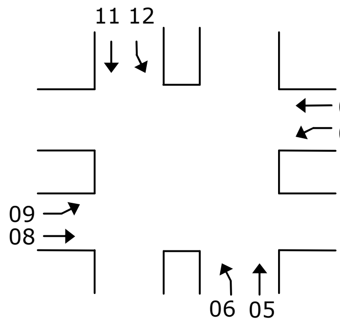


Fig. 2.1 Example intersection

The capacity of all streams is 1800 veh/h, the yellow time is 3 s. For several examples, the flow of the streams is given in Tab. 2.1, the clearance times in Tab. 2.2.

Examples 2 and 3 are a variation of example 1, the flows 02-06 are increased with a difference "diff", flows 08-12 are increased with the same difference. As will be shown, this has a great impact on the minimum cycle time.

Tab. 2.1 Flows of the streams in example 1-3

Stream	02	03	05	06	08	09	11	12
q_i ex. 1	500	200	500	200	500	200	500	200
q_i ex. 2, diff=20	480	180	480	180	520	220	520	220
q_i ex. 3, diff=90	410	110	410	110	590	290	590	290

Tab. 2.2 Clearance times of successive streams for example 1and 2

from	02	09	09	05	12	12	08	03	03	11	06	06
to	09	05	11	12	02	08	03	05	11	06	02	08
ex.1, ex. 2, ex. 3	1.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	1.0

The structure of example 1 is given in Fig. 2.2a. The conflict group cycle time is maximum for the 1st and 4th conflict group: T_{CCrit} = 63.0 s, whereas T_{CWeb}=117 s.

In example 2, the difference of the flows with respect to example 1 is set to 20 (Tab. 2.1). In this situation there is no single critical conflict group that determines the minimum cycle time. Conflict group 1 and 4 give still a maximum $T_{Ccrit,j}$ =63.0 s and $T_{CWebster}$ is the same, 117 s, but now T_{Cmin}= 73.1 s. In this example, stream 08 determines the start of 12, and 12 determines both the start of 03 and 09. Stream 09 determines the start of stream 11, which determines both stream 08 and 02. The critical path is 08-12-09-11-08 and it extends over all four conflict groups.

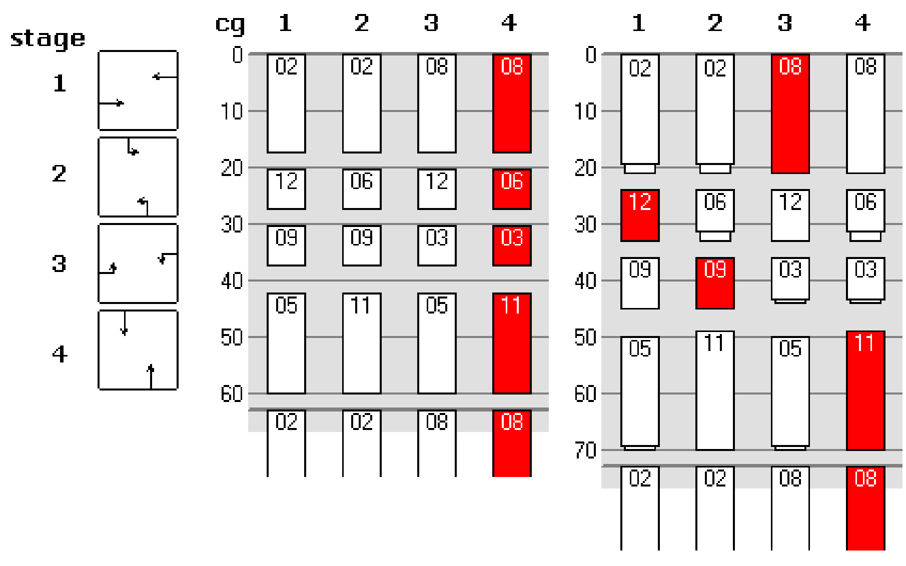


Fig. 2.2 Screenshots of optimized structures generated with VRIGen, Fig2.2a-b correspond with example 1-2 Horizontal the conflict group number "cg" is given, vertical the stage number. The vertical size of a stream box is a measure of its green time. "cg" stands for "conflict group"

The streams that do not determine the next stream can receive more green time than their flow demands. If the tactic option of VRIGen "Continue Parallel Green" is enabled, the stream is continued. In Fig. 2.2b this is depicted as a smaller box underneath the stream box. The equation Eq. 2.1 is still suitable to determine the minimum cycle time T_{Cmin}, but instead of the critical conflict group, the lost time and flow of the critical path must be used. For example 1 and 2 T_{Crit} is the same, but the resulting T_{Cmin} is 10.1 s larger than the critical conflict group cycle time.
In contrast to conflict groups, critical paths may contain streams that have no mutual conflict: stream 08 and 09 are not conflicting, but both play a role in determining the minimum cycle time T_{Cmin}.

Example 3 generates a cycle time T_{CMin}=585 s, this exceeds the standard prescribed maximum cycle time of 120 s by far. The critical conflict group cycle time $T_{Ccrit,j}$ =75.5 s and T_{CWeb}=117 s are too small.

Fig. 2.3 shows how the cycle time changes if the difference in flow for stream 02, 03, 05, 06 is decreased and the flow for stream 08, 09, 11, 12 is increased with the same amount with respect to example 1.

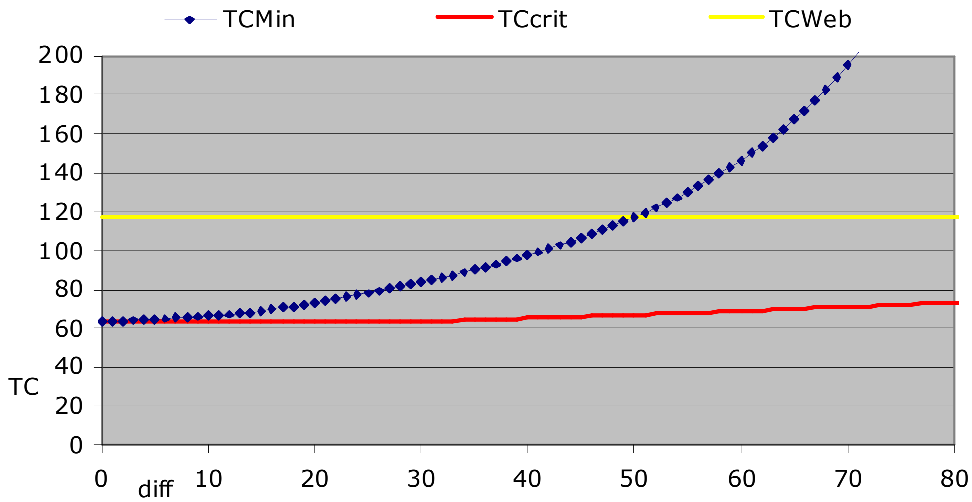


Fig. 2.3 The minimum cycle time, the critical conflict group cycle time and the Webster cycle time as function of the difference between the flows of stream 02, 03, 05, 06 and 08, 09, 11, 12 with respect to example 1.

If "diff" is larger than 30, T_{CCrit} increases a bit, since stream 03 and 06 do not reach minimum green time and Eq. 2.4 is used. As can be seen, T_{CCrit} is under dimensioned for these examples, and only if "diff" is near 50, T_{CMin} agrees with T_{CWeb}.

2.2 Critical conflict path extending over more than one cycle.

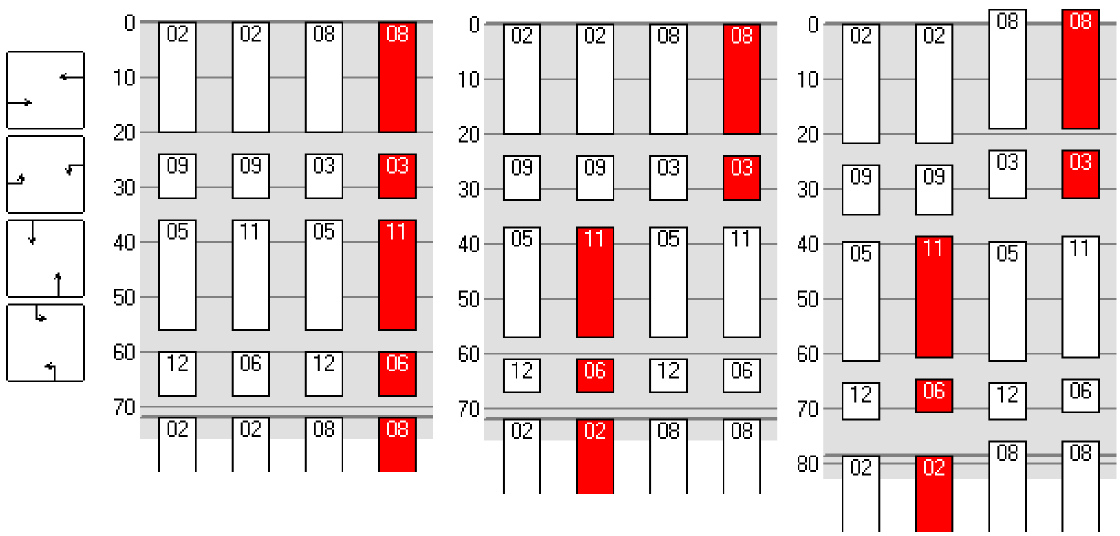
For example 2, the critical path ends with the same stream as it begins. It is also possible that a critical path ends with another stream than it begins with. As long as both streams have an equal start at the next cycle, the T_{CMin} can be calculated with the critical path and Eq. 2.1. If the critical path is not circular and begin and end stream do not start at the same time, the critical path extends over more than one cycle. To explain this in more detail, three examples are given, for the same intersection, with another structure (Tab. 2.3, Tab. 2.4, Fig. 2.4). Example 4 has a critical conflict group, which gives a T_{CMin}=72.0 s, T_{CWeb} = 130.5 s. Example 5 has a non-circular critical path: 08-03-11-06-02, but the begin stream 08 starts at the same moment as the end stream 02. Eq. 2.4 is used with the critical path to find the minimum cycle time, because with given flow stream 06 does not reach the minimum green time. The critical conflict group (2 or 4) gives T_{Crit}= 69.0 s, T_{CMin}= 72.0 s, T_{CWeb}=122.0 s.

Tab. 2.3 Flow of the streams streams for examples 4-6

Stream	02	03	05	06	08	09	11	12
q_i ex. 4	500	200	500	200	500	200	500	200
q_i ex. 5, ex. 6	500	200	500	100	500	200	500	150

Tab. 2.4 Clearance times of successive streams for examples 4-6

from	02	09	09	05	12	12	08	03	03	11	06	06
To	09	05	11	12	02	08	03	05	11	06	02	08
ex. 4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ex. 5	1.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	1.0
ex. 6	1.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	4.0	1.0	5.0	1.0



a. *b.* *c.*

Fig. 2.4 Screenshots of structures generated with VRIGen, Fig2.1a-c: example 4-6

In example 6 the lost time between 03 and 11 and 06 and 02 are increased by an extra amount and now stream 08 and 02 do not have an equal start. The critical conflict group gives a T_{Crit} = 78.0 s, T_{CWeb} = 128.0 s, the minimum cycle time is T_{Cmin}= 78.9 s. The path 08-03-11-06-02 gives a cycle time of 94.5 s, the path 02-09-05-12-08 gives a cycle time of 63.0 s. It appears that the paths 08-03-11-06-02 and 02-09-05-12-08 are not two separate paths, but this structure needs two cycles to handle all streams. The path is 08-03-11-06-02-09-05-12-08 and the minimum cycle time is:

$$T_C = \frac{\sum_{i=1}^{2M} T_L}{2 - \sum_{i=1}^{2M} \frac{q_i}{s_i}} \qquad 2.6$$

The difference in cycle time of path 08-03-11-06-02 and 02-09-05-12-08 results in a difference in start offset S_1 of 02 and 08:

$$S_1 = \left(1 - \sum_{i=1}^{M} \frac{q_i}{s_i}\right) T_C - \sum_{i=1}^{M} T_L \qquad 2.7$$

2.3 *Determination of the minimum cycle time with VRIGen explained with a flow diagram*

In Fig. 2.5 the flow diagram for the minimum cycle time determination as used in VRIGen is given. In this flow diagram T_{Ci} is the cycle time of the conflict group: the difference between the offsets of the first stream of a conflict group in this and the next cycle. This should be smaller or equal to the minimum cycle time.

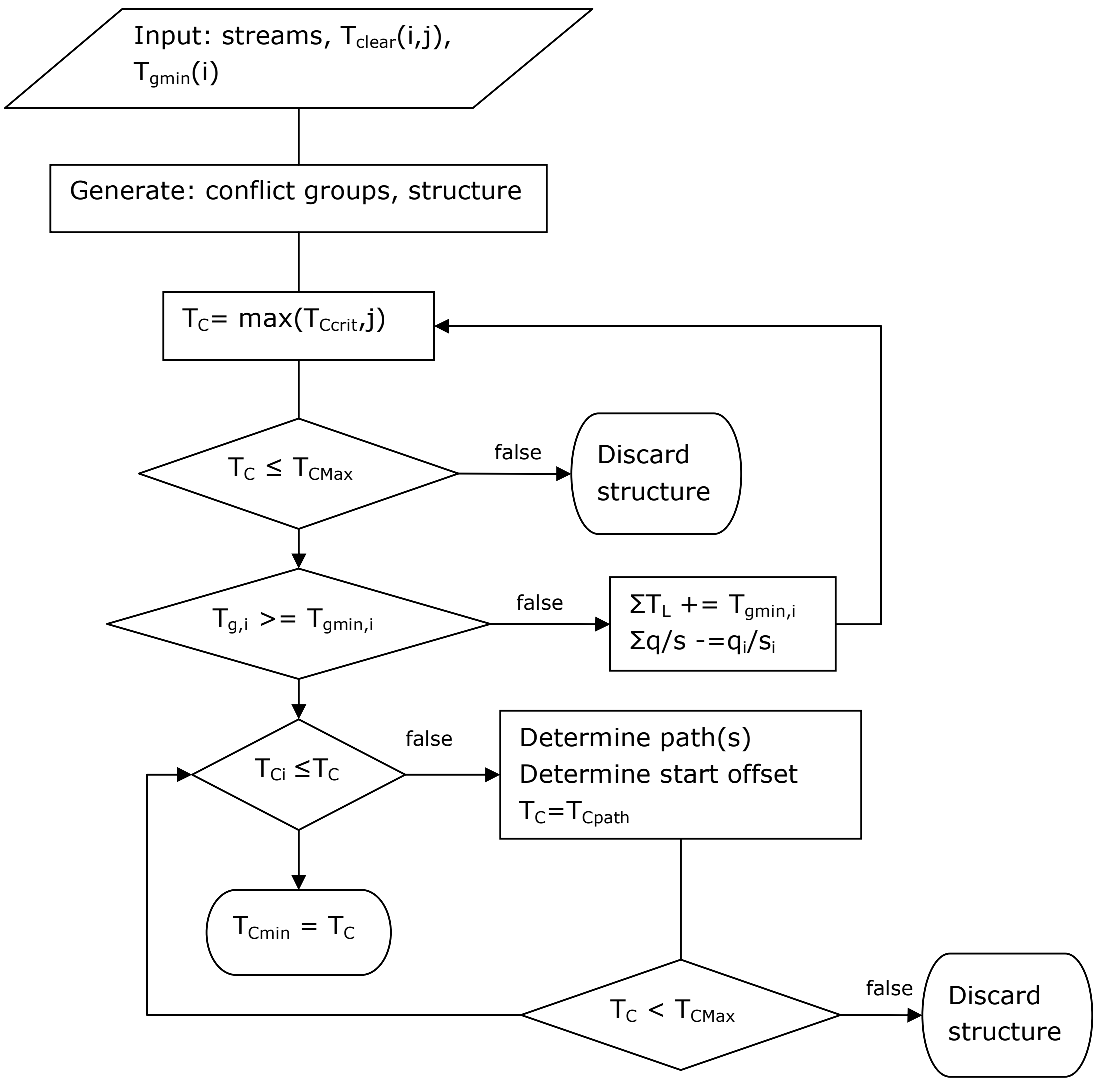


Fig. 2.5 Flow diagram for determination T_{Cmin} as used by VRIGen

If $T_C > T_{CMax}$ the structure is discarded, but the user is warned that the T_{CMax} must be increased to find all structures.

VRIGen finds the path by generating the structure with given T_C. With this value the green times are calculated and the offsets of the streams are deterimined. The resulting conflict group cycle time T_{Ci} should be smaller or equal T_C. If $T_{Ci} > T_C$ applies, a critical path determines the cycle time. The stream in the last stage with the largest T_{Ci} is the last stream in the critical path. If all conflict groups have the same conflict group cycle time, the stream of the last conflict group is chosen. The stream in the stage before the last stage which determines the synchronization loss of this stream is also part of the critical path, and so on until the first stage is reached. More than one critical path is possible.

In Tab. 2.5 an example for finding the critical path is given for the structure of example 2 (Fig. 2.2b), the path is marked with yellow. If the critical path has to swap to another conflict group, this is indicated with orange.

Tab. 2.5 The yellow indicated critical path 08-12-09-11-08 (or 08-12-09-11-02) is determined for example 2, input T_C=T_{Crit}=63.0, T_{Ci} > T_{Crit}

conflict group cg	1	2	3	4
stage 1, stream	02	02	08	08
S_1	0.0	0.0	0.0	0.0
T_{g1}	16.8	16.8	18.2	18.2
T_{L1}	3.0	3.0	3.0	3.0
S_{min1}	19.8	19.8	21.2	21.2
stage 2, stream	12	06	12	06
S_2	21.2	21.2	21.2	21.2
T_{g2}	7.7	6.3	7.7	6.3
T_{L2}	3.0	3.0	3.0	3.0
S_{min2}	31.9	30.5	31.9	30.5
stage 3, stream	09	09	03	03
S_3	31.9	31.9	31.9	31.9
T_{g3}	7.7	7.7	6.3	6.3
T_{L3}	5.0	4.0	4.0	5.0
S_{min3}	44.6	43.6	42.2	43.2
stage 4, stream	05	11	05	11
S_4	44.6	43.6	44.6	43.6
T_{g4}	16.8	18.2	16.8	18.2
T_{L4}	3.0	3.0	3.0	3.0
S_{min4}	64.4	64.8	64.4	64.8
Next cycle				
stage 1, stream	02	02	08	08
S_{1next}	64.8	64.8	64.8	64.8
T_{Ci}= S_{1next}-S_1	64.4	64.8	64.4	64.8
				start

De minimum offset for a stream at (cg, stage) is calculated with Eq. 2.8, the offset of a stream in the next stage is the maximum value of all S_{min}(cg,stage) at those positions where this stream is present (Eq. 2.9)

$$S_{\min stage}(cg) = S_{stage}(cg) + T_g(cg,stage) + T_L(cg,stage) \qquad 2.8$$

$$S_{stage+1}(cg(stream)) = \max_{cg}\left(S_{\min stage}(cg(stream))\right) \qquad 2.9$$

The stream which determines S_{stage}(cg) is in the critical path: 08-12-09-11-08. The path 08-12-09-11-02 is also permitted, both paths give the same T_{CMin}=73.1 s.

2.4 Determination of Extension Green with the aid of the critical path.

Extension green is the sub-phase that extends the green phase as long as the queue before the stop line lasts, but up to the given maximum cycle time T_{Cmax}. VRIGen determines the maximum extension green by adding a percentage to all flows until the cycle time reaches the maximum cycle time. This was previously done iteratively, a rather time consuming task. With the aid of the critical path the extension green can be computed in only a few steps and in only a single step if the critical path is the same for T_{Cmax} and T_{Cmin}.
Take p_{ext} as the factor with which all flows are multiplied to reach the maximum cycle time, q_i is the flow at T_{CMin}:

$$p_{ext} = \frac{1 - \frac{\sum_i T_{L,i}}{T_{CMax}}}{\sum_i \frac{q_i}{s_i}} \qquad 2.10$$

The maximum extension green time is:

$$T_{gext,i} = p_{ext} \frac{q_i}{s_i} T_{C\max} - T_{fix,i} \qquad 2.11$$

With $T_{gfix,i}$ is the fixed time green for stream i.

If the flows are multiplied with p_{ext}, but the corresponding T_{Cmin} does not equal T_{CMax}, the critical path is not the same for T_{Cmax} and T_{Cmin}. The critical path at T_C=T_{CMax} is found by filling T_{Cmax} in the structure (as in Tab. 2.5) and then Eq. 2.10 is used again.

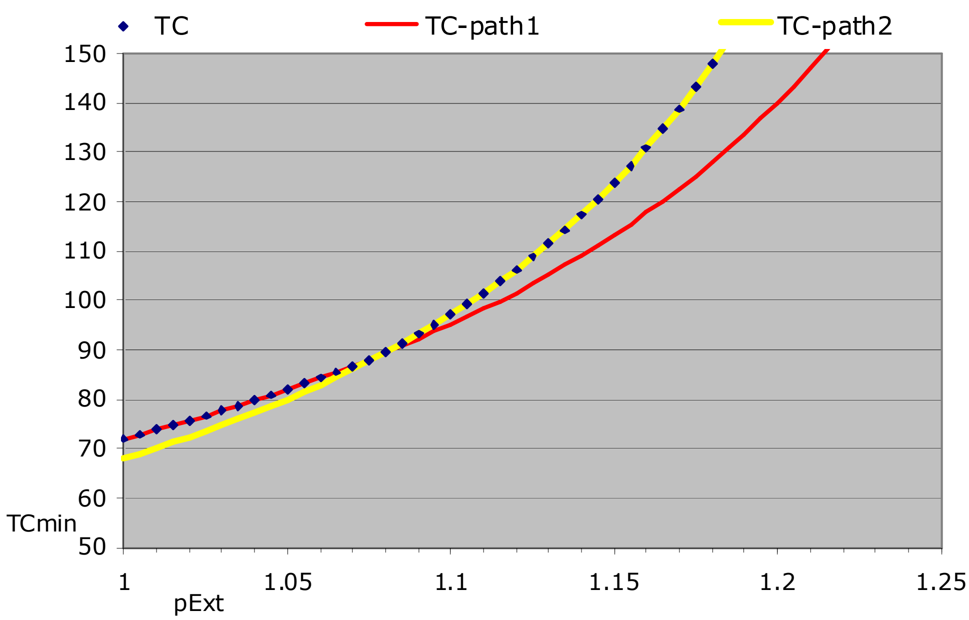


Fig. 2.6 T_{Cmin} as function of the multiply factor of the flow p_{Ext} for example 5 (Fig. 2.4b)

An example of the increase of the cycle time as function of p_{ext} for the structure of example 5 is given in Fig. 2.6. It shows how T_{Cmin} increases with p_{ext} (black dots). For $p_{ext} < 1.08$ T_{Cmin} is determined by the critical path: 08-03-11-06-02-09-05-12-08 (red line), for $p_{ext} >= 1.8$ T_{Cmin} is determined by the 3rd critical conflict group: 08-03-05-12-08 (yellow line).

If some streams do not reach the minimum green time T_{gmin} at T_{Cmin} the minimum green time is added to the lost time and their flow is not taken into account (Eq. 2.4). At T_{CMax} the green time of these streams can be larger than the minimum green, so the lost time is without the minimum green time and its flow counts again in Eq.2.10.

3 Conclusion

The critical path approach is a fast method to determine the minimum cycle time of a structure. The search of the correct critical path can be generally handled in one step, sometimes in several steps with a fast procedure. This saves much time in comparison to the formerly used iterative method. Also the maximum green time can be found with the aid of the critical path and is much faster than the earlier used iterative method.

Further development of VRIGen will be on the matter of structure generation, for this now determines the computing time. Intersections that are equipped with software that generates within seconds the most optimal structure, determined by the traffic volumes that are present will reduce waiting queues even more. The future can still be brighter than today…

Bibliography

[1] Webster, F. V., and B. M. Cobbe. 1966. Traffic Signals, Road Research Tech. Paper No. 56, 55-60. Her Majesty's Stationery Office, Lodon, U.K.

[2] T.H.J. Muller and M. de Leeuw. A new method to design traffic control programs. In *TRB 2006 Annual Meeting CD-ROM*, 2006. http://trb.metapress.com/content/p1h4514q340n4431/

[3] http://www.tpa.nl/CCOL/introductie_CCOL.php

[4] P. Furth and T. Muller. Trafcod: A method for stream based control of actuated traffic signals, 78th Annual meeting of the Transportation Research Board, Washington, USA, 1999