Assessing the quality of complex fixed-time traffic signals

Peter Wagner*, Thorsten Neumann, and Robert Oertel
Institute of Transportation Systems, German Aerospace Center, Rutherfordstraße 2,
12489 Berlin, phone +49 30 67055 237, eMail: peter.wagner@dlr.de

Abstract
Usually, the parameters of fixed-time traffic signals – which are still the most common type of signals in the world – will be determined on the basis of a characteristic traffic demand pattern. So, their quality assessment is also based on this single traffic demand pattern. This work describes a more complete assessment method which is better suited to find the average performance of the signal. The method is especially well adopted to complicate intersections with a large number of different traffic streams and traffic signal phases. In principle, this approach can also be used not only for the analysis, but also for the generation of a good set of traffic signal parameters.
Keywords: quality assessment, fixed-time traffic signals, quasi-random numbers

Introduction

When setting up a fixed-cycle traffic signal, traffic engineers try to determine the demand for traffic at this intersection and pick e.g. either the peak demand hour or something like the 99-percentile of the overall demand (the HCM defines it for the 30-th highest hour of the 8760 hours of the year ordered with decreasing demand [3] – this is the 99.65-percentile) to base the computation of the fixed cycle parameters on. In addition, those data are then used to do an assessment of the intersection control under consideration. Obviously, this type of assessment is quite limited, since it looks only at a small number of special cases and ignores a large number of possible scenarios. This work introduces a different scheme. It addresses especially complex intersection where one has to consider a large number of different scenarios in order to arrive at a more exhaustive assessment of the quality of the signal. Furthermore, this method is also applicable to tests with micro-simulations of traffic flow, since it tends to minimize the number of simulation scenarios to be run to determine the quality of the traffic signal settings as function of a set of general demand patterns.

The challenge

Traffic signals operate in phases, where the numbers of phases P range from two up to more than ten. Each phase may provide green to several traffic streams that are non-interacting or only weakly interacting. A normal four arm intersection can have 12 vehicle streams, at least 4 pedestrian streams (one for each arm), and a small number of public transit streams, altogether N different streams. If controlled by a fixed cycle signal, such a control has the following parameters: the cycle time C and the durations of the phases $\{D_i\}_{i=1,\dots,P}$, together with the green times $\{G\}_{i=1,\dots,N}$ for each of the streams, which are determined by the demand for traffic $\{q_i\}_{i=1,\dots,N}$ of the streams. Here, it will not be considered how to best assign the streams to the phases, it will be assumed that this has been done already. Therefore, the intersection has to handle the demand for the different streams by assigning the correct green and cycle times. For an isolated intersection, the cycle time is determined from the demand by the following equation [1,3]:

$$C = \frac{1.5L+5}{1-\sum_{i=1}^{N}\frac{q_i}{Q}} \qquad (1)$$

In this equation, L is the total loss time of the signal, i.e. the sum of all the intermediate times required when changing phase, and Q is the saturation flow of the traffic stream. Note, that Q may depend on the stream i itself, e.g. right-turners having to wait for a pedestrian stream may have a different saturation flow than a stream of straight-ahead vehicles, while different approaches may have a different number of lanes. Note, that this equation is not applicable in any case, a counter-example is provided by a signal embedded into a co-ordination scheme when the cycle time is determined externally.

After the cycle time is fixed, the green times are determined from the demands as follows:

$$G_i = \frac{q_i}{\sum_{i=1}^{N} q_i}(C-L) \qquad (2)$$

In the following, it will be dealt with the question of how to best do an assessment of the intersection that does take into account a much broader range of demand scenarios. Given a measurement of the daily course of the demand functions $\{q_i(t)\}_{i=1,\dots,N}$, which of the $\{q_i(t)\}_{i=1,\dots,N}$ should be used to set-up the signal?

To assess the quality of the intersection under this set of scenarios, at least two methods can be used. The first is analytically and is based on a delay model, here the model of Webster [1] will be used:

$$d_i(q_i) = \frac{\alpha}{2}\left(\frac{\left(1-\frac{G_i}{C}\right)^2}{1-\frac{q_i}{Q}}C + \frac{\left(\frac{q_iC}{G_iQ}\right)^2}{q_i\left(1-\frac{q_iC}{G_iQ}\right)}\right) \qquad (3)$$

$$d(q_1,\dots,q_N) = \sum_{i=1}^{N}\frac{q_i}{\sum_{i=1}^{N} q_i}d_i \qquad (4)$$

The delay in these equations is defined as delay per vehicle. The second approach is to set-up a microscopic simulation that is run for all the different scenarios selected (a scenario is defined here as a particular demand pattern $\{q_i\}_{i=1,\dots,N}$ applied for a certain time interval, e.g. one hour) to assess the performance of the signal settings. In principle, it would be desirable to estimate the performance for as many demand patterns as possible. However, running a microscopic simulation is a time-consuming task. Even worse, the space of all possible demand patterns is fairly large. If one tries to conquer e.g. a ten-dimensional pattern space by having in each dimension just two values for the demand $\left\{q_i^{(\min)}, q_i^{(\max)}\right\}$ then 1024 different simulation runs result. This is sometimes called the "curse of dimensionality", a term introduced by Richard Bellmann. What is even worse is that it can be shown that this approach does not make good use of the compute resources: by using a so called sequence of quasi-random numbers (for example a Sobol sequence [4], see next section) a much better coverage of this large pattern space can be achieved. So this is the scientific question we would like to answer: how can we characterize a fixed cycle signal with a given set of parameters and a set of corresponding demands with a minimum effort?

Discrepancy

Either analytically as given above, or by simulation, it is possible to compute for a certain demand $\{q_i\}_{i=1,\dots,N}$ the corresponding delay $d(q_1, \dots, q_N)$. The delay depends on the traffic signal parameters as described above. A complete characterization of the traffic signal at hand would try to test as much input values $\{q_i\}_{i=1,\dots,N}$ as possible in order to compute the average demand as a performance indicator. Of course, to be well-defined, at least a minimum and maximum value for each of the demands should be defined, which defines an *N*-dimensional cuboid. So, the question arises: how many (*n*) different input scenarios $\{q_i\}_{i=1,\dots,N}$ are needed to arrive at a good estimation of the average delay?

This is a well-defined mathematical question, and it has a very interesting answer which, to the best of our knowledge, has not been used in the traffic management (see [2] for a review), but there are applications related to the modelling of the demand for traffic, see [8] for an example. The currently best known method that does this job is a sequence of quasi-random numbers[1].

[1] Quasi-random numbers should not be confused with pseudo-random numbers. Pseudo-random numbers are, like quasi-random numbers, produced by a deterministic algorithm and "look random". True random numbers are difficult to obtain, for most questions the difference between pseudo-random and random is not important.

Computing the mean values of all the delays of all the *n* points in this *N*-dimensional space is essentially an integration of the delay function $d(q_1, \ldots, q_N)$ over the cuboid. Therefore, it is very interesting to know the integration error that is produced by a certain integration scheme, which is basically the choice of the points in the cuboid, and of course to use a scheme that makes this integration error as small as possible.

Albeit there is (of course) a much longer story behind this, the following might be sufficient to know. In one dimension, a set of equidistantly distributed points (i.e. the famous trapezoidal rule) yields a $1/N^2$ convergence, while just throwing points randomly (produced by a pseudo-random number generator) in the interval yields a very weak $1/\sqrt{N}$ convergence. In higher dimensions, this changes: distributing points randomly still has the $1/\sqrt{N}$ convergence, while the equidistant point distribution becomes worse than even this simple random distribution. It turns out that the so called quasi-random numbers produce an even better $1/N$ convergence in any dimension. An example of such quasi-random numbers in 2D can be seen in the Figure 1.

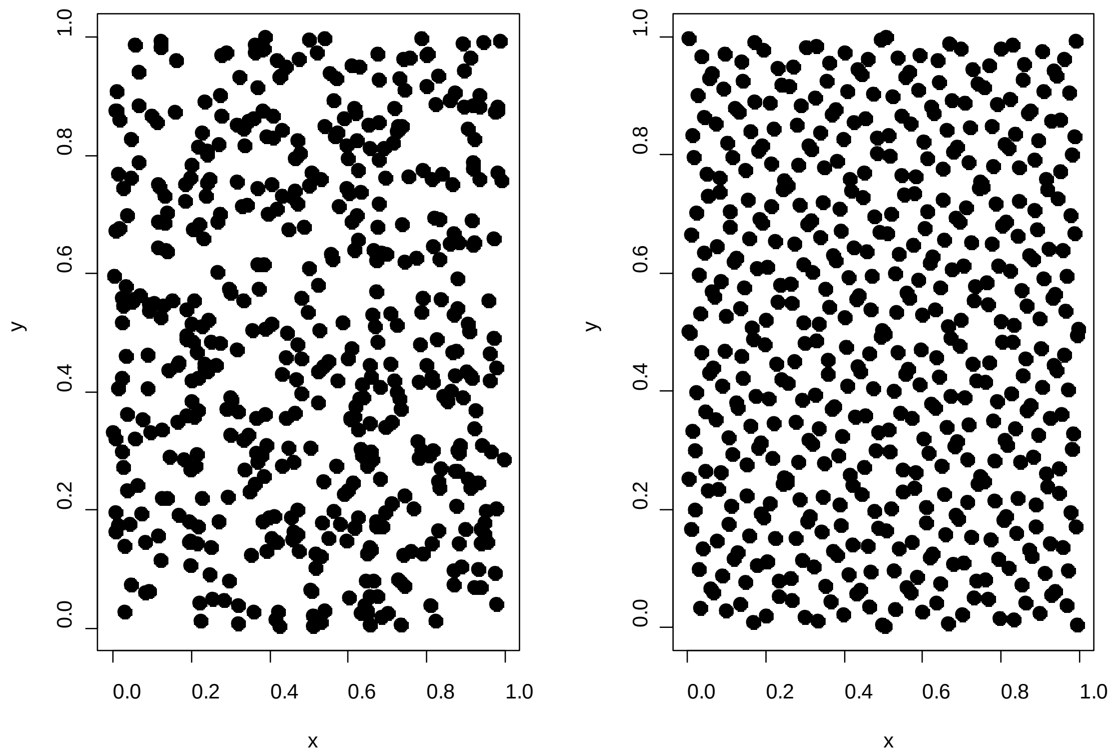

Figure 1 - The first 512 points of pseudo-random sequence (left) and of a Sobol quasi-random number sequence (right). These points have been generated using the pseudo- and quasi-random number algorithms [6] implemented in R [5]

The measure that describes how fast such a sequence converges towards the true mean value is the so called discrepancy. The precise definition (including a piece of Fortran code) can be found in [2], here the interesting point is that such a sequence of quasi-random numbers has in fact a much better approximation quality, an example is shown in the Figure 2.

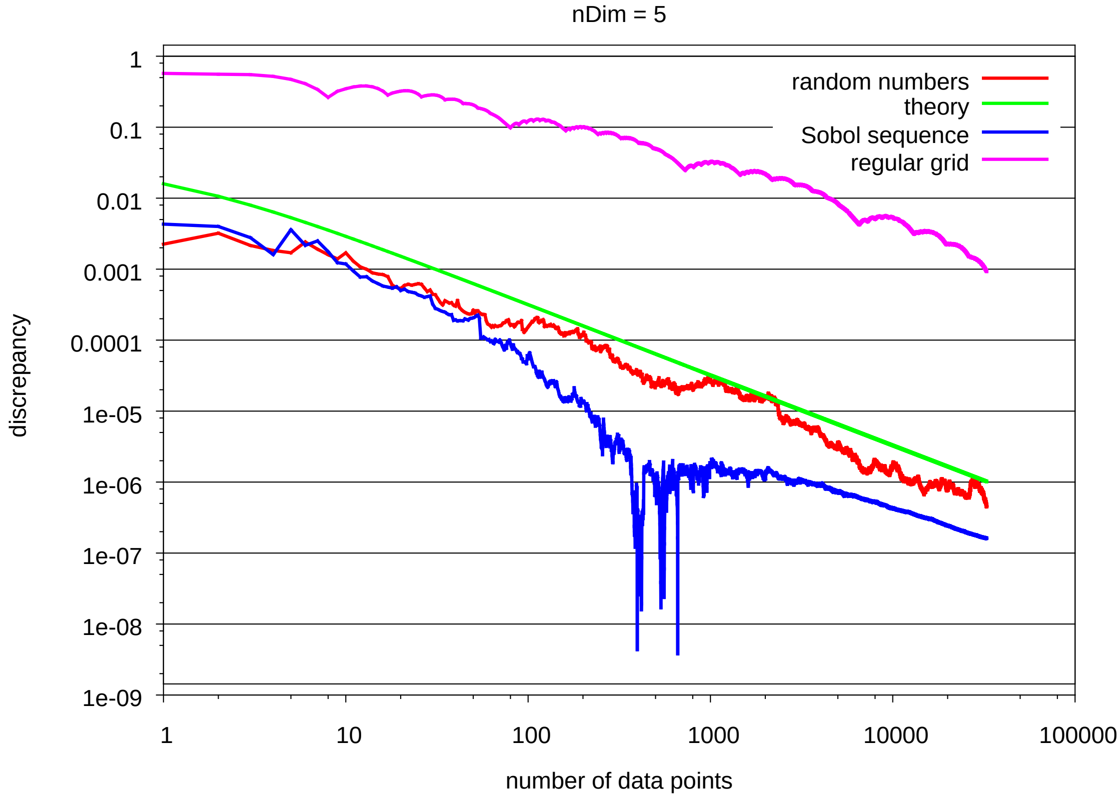

Figure 2 - Discrepancy for three 5-D sequences: a regular grid (pink), a normal sequence of random numbers (red), for which an analytical result exists (green) and a Sobol sequence (blue). The Sobol sequence's discrepancy is ten times smaller than a normal random sequence, and more than 1,000 times better as a regular 5-D grid. To get the same accuracy as for 10,000 random five-dimensional vectors, just 1,000 simulations with a Sobol sequence are needed. Pseudo-random numbers have been generated with the C standard library, Sobol sequences with the GSL [7].

A practical example

Since 10-dimensional spaces are difficult to visualize, the following intersection in Figure 3 will be considered which has just three phases:

- Phase 1: Traffic flows in west/east (q_1) and in east/west (q'_1) direction
- Phase 2: The traffic from west to east is stopped, to let the left turns from east to the south pass (q_2). In addition, there is a bus service on this stream.
- Phase 3: The traffic from the south gets it green time (q_3).

This is only a 3-dimensional example, and by analysing for example only two of the three traffic streams, a simple 2D representation can be displayed.

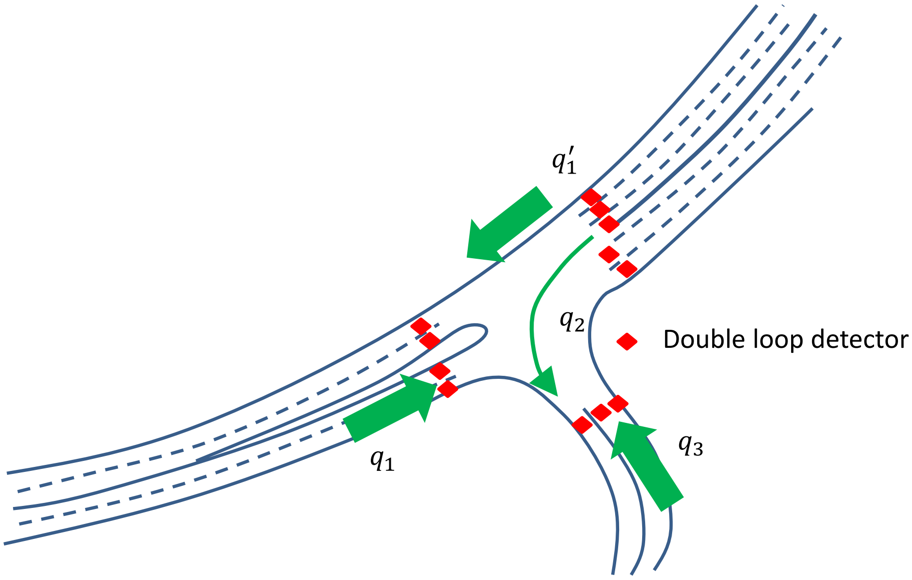

Figure 3: Sketch of the intersection analysed in the text.

To see how real data look like, the data collected from one week in June 2011, aggregated to five minutes intervals will be analysed in the following (see Figure 4).

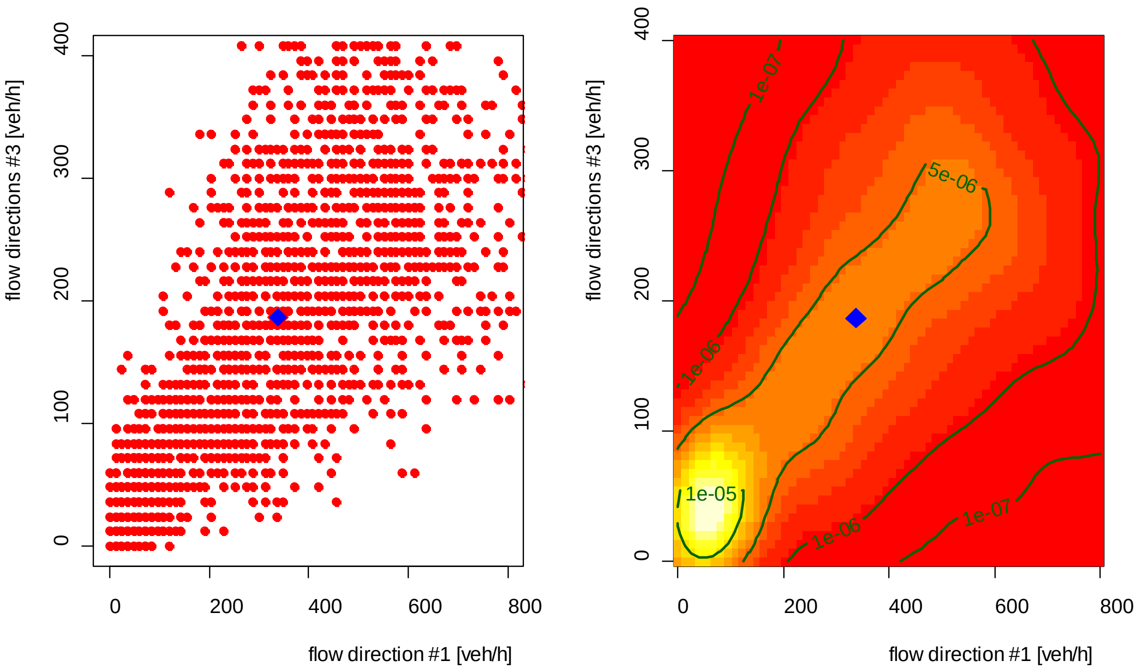

Figure 4: Scatter plot (left) of the five minutes data for the intersection in Figure 3, and a density plot of these data showing a strong concentration on small demand values (right). The blue diamond is the average value.

This shows one important property of the data: they are distributed in a very inhomogeneous manner in the space of the demands, especially the density plot Figure 4 (right) demonstrates this clearly. It is highly likely, that this is not a particular feature of the data at hand, but is common for real data. Of course, the above approach with quasi-random number sequences can be applied to those inhomogeneous demand data as well, simply by weighting each demand pattern according to the probability with which it appears. For all those cases, where such a detailed probability distribution does not exist, one can still use the normal quasi-random number sequences that are bounded by reasonable values of minimum and maximum demand.

One step further

To demonstrate the benefit of this approach even further, the example above has been dealt with slightly different. Given the data at hand, a fixed cycle configuration has been computed. This uses the maximum of the demand data (q_1, q_1', q_2, q_3) as defined in Figure 3, and from this, it follows the standard procedure to compute a fixed cycle control. The fixed cycle equations (1) and (2) have two parameters that have to be set, namely the saturation flow Q and the total loss time L. While the loss time can be set for this example to 10 s, the saturation flows are not independent of the index. For instance, in the main direction there are two lanes, the left turners have one lane, and the vehicles from the south have almost two lanes for the left and right turners. So, the saturation flows were set to (1.2, 1.2, 0.55, 0.75) veh/s. From this, the fixed cycle parameters have been computed, it yields $(G_1, G_2, G_3) = (22, 21, 20)$ s with a cycle time of 75 s. Note, that the stream 1p gets a total green time of $G_{1p} = G_1 + G_2$.
By inserting these parameters and the roughly 2000 different flow values contained in the data-set into Webster's equations (3) and (4), the average delay for this set-up can be computed. For the data-set at hand, it turns out that each vehicle has to wait for 17.2 s on average.
In the next step, 500 Halton-distributed sets of green times had been tested with the data. The green times were distributed between a minimum and a maximum green time, and from each set of green times the corresponding delay belonging to this set had been computed. Not unexpected, the delays now cover a wide range of values. More important however, is the fact that better values had been found by this approach than by the traditional approach. The best value that has been found so far is given by the green times $(G_1, G_2, G_3) = (16, 16, 17)$ and yields an average delay time of 14.8 s, giving an overall improvement of about 13%.

Conclusions

The scheme presented here is not only useful for the quality assessment of a fixed cycle signal, but can be used much more generally in any case where an objective function depends on many parameters, e.g. when determining the best possible strategy and the like. It tries to draw a general error margin on some objective function one wishes to determine. It helps to optimize computational resources, since it clarifies how many simulations are needed to achieve a certain result in terms of precision. Note also, that the research in quasi-random numbers is not yet finished, so it eventually may happen that even stronger quasi-random schemes will be found.

In any case, the method suggested here yields a much more balanced assessment of a fixed cycle signal, especially of the very complex ones. It also demonstrates, that in dimensions larger than 2, the most straightforward approach of a regular grid is completely outperformed by a quasi-random sequence, and to a smaller degree, even by a pseudo-random sequence.

Finally, it has been demonstrated, that the whole approach can be used just the other way round: instead of determining the quality of a given fixed-cycle signal, one might try to directly find the best set of parameters, this time by finding for a given set of demands the best possible set of parameters – now, the search with quasi-random numbers runs over the parameter space of the signal, i.e. it tests different combinations of green times and tries to find the one which yields the minimum average delay time (or any other performance measure). However, this turns out to be not trivial, since with this combination it happens fairly often, that one hits a condition with over-saturation – simply due to the fact, that the green time were chosen too small causing the denominator in the second term of equation (3) to overflow. So far, this has been handled by setting in this case a rather large value for the delay (equal to the five minutes aggregation interval). Smarter solutions are clearly needed and are one topic of future research.

References

1. Webster, F. V. (1958). *Traffic signal settings*. Road Research Laboratory Technical Paper No. 39
2. James, F., J. Hoogland, R. Kleiss (1996). *Multidimensional sampling for simulation and integration: measures, discrepancies, and quasi-random numbers*, see preprint server arxiv.org, http://arxiv.org/abs/hep-ph/9606309v3
3. HCM 2010 - *Highway Capacity Manual*, Transportation Research Board, Washington

(2010).

4. Sobol, I. M. (1967). *Uniformly distributed sequences with additional uniformity properties*. USSR Comput. Math. and Math. Phys. 16: 236-242.
5. R Development Core Team (2011). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.
6. C. Dutang and P. Savicky (2010), *randtoolbox: Generating and Testing Random Numbers*.
7. M. Galassi J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi (2009), *GNU Scientific Library Reference Manual (3rd Ed.)*, ISBN 0954612078, URL: http://www.gnu.org/software/gsl/
8. C. R. Bhat (2001), *Quasi-random maximum simulated likelihood estimation of th emixed multinomial logit model*, Transportation Research Part B, 35, 677 -- 693.