Green Time Split Optimization by Dynamic Programming

Tom Bagehorn (presenting author)
Siemens Mobility GmbH, MO MM ITS SE, Werner-von-Siemens-Str. 6, 86159 Augsburg, Germany; mailto:tom.bagehorn@siemens.com

Andreas Poschinger
Munich University of Applied Sciences, Faculty 05, Lothstraße 34, 80335 München, Germany; mailto:andreas.poschinger@hm.edu

Problem Statement

The quality of urban road networks mainly depends on the capacity of its intersections and their generated delay. Therefore, many urban intersections are controlled by traffic signals to improve traffic quality and safety while decreasing pollutant air emissions.
Modern model-based adaptive traffic control systems like SCOOT [HUNT, et al., 1982], OPAC [GARTNER, 1982, 1983; GARTNER, et al., 1983], Sitraffic Motion [BUSCH and KRUSE, 2002] and others adapt to changing traffic demand more flexible than other control strategies and can be regarded as state of the art. However, with the ongoing digitalization, new possibilities and innovative ideas arise. Autonomous driving is a hot issue and raises the question for new traffic light control systems. In 2013, ERDMANN [2013] introduced a new green time split optimization method taking into account individual vehicle arrivals. In the past GRAFTON and NEWELL [1967], ROBERTSON and BRETHERTON [1974] and GARTNER [1983] utilized dynamic programming for traffic signal control. All latter three approaches use dynamic programming as it was introduced by BELLMAN [1957]. The solution is computed recursively from the future back to the current point in time. This approach ensures that the real optimal solution for a time series of control actions is found rather than a heuristic approximation. The disadvantage of dynamic programming is ist high computational effort. In contrast, Erdmann utilizes a kind of forward dynamic programming method where the solution is calculated from the current point in time to the future.

Research Objectives

The research focus of this contribution is the question whether dynamic programming as proposed by Bellman and used by Robertson and Bretherton is nowadays applicable to real world problems that also comprise individual arrivals so that it is forward compatible to upcoming control systems. The hope is, that the nowadays computer power is capable. If it rules out that computer power is still limited for dynamic programming possible solutions or heuristics shall be developed. A first major step is to make a dynamic programming-based optimization method applicable under real world conditions. This major step forms the research scope of this paper.

Methodological approach

The first step is to review ROBERTSON and BRETHERTON's [1974] DYPIC algorithm; for that purpose, it is necessary to program it. Then the second step of this paper is to overcome any drawbacks of the DYPIC method. This step is mainly subdivided into two parts: first to address basic algorithmic problems, which arise from the nature of traffic and traffic control theory, such as to allow realistic saturation flow rates. The solution in this step leads to an offline optimization algorithm. The second part deals with the principle problem of current traffic control systems that the arriving vehicles are not known in advance being a major problem for backward dynamic programming. The solution for that issue leads to an online optimization algorithm.

Fig. 1 shows Bellman's principle of dynamic programming.

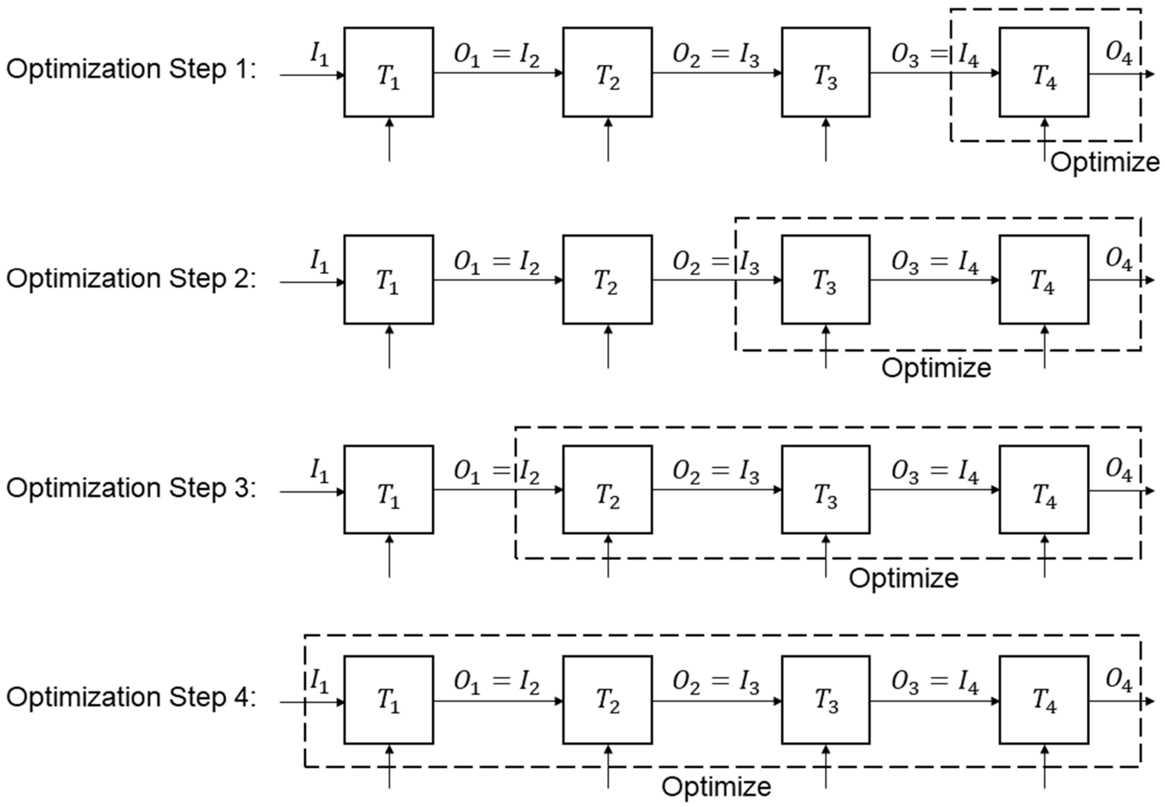

Fig1: Bellman's principle

Bellman's principle says that if the decision step n in a series of decision steps is optimal then also the series of the optimal decision step n-1 and decision step n is optimal if decision step n-1 leads to decision step n. In more practical terms: the last decision step N is optimized first, then the decision step N-1 is optimized in a way, that its output O_{N-1} fits to input I_N of decision step N and so on as seen in Fig 1. for N=4.

The decisions in ROBERTSON and BRETHERTON [1974] are modeled as two unidirectional approaches A and B which one of them in green and the decision to change direction in green or not which is calculated on a five seconds basis assuming a departure rate of 2veh /5sec interval (Details shown in full paper). The drawback of this method is that the inter-green time is always assumed to be 5sec as well as is the minimum green time. Further more than one stage cannot be easily extended.

This contribution overcomes those limits by additional states which model the boundary conditions (see Fig. 2); the original approach of ROBERTSON and BRETHERTON use one state per approach only. Changes of right of way can only be done in states 0. The remaining states (in example states 1-4) assure the inter-green and minimum green times.

Another extension allows for usage of an arbitrary number of stages (not shown in extended Abstract but in full paper).

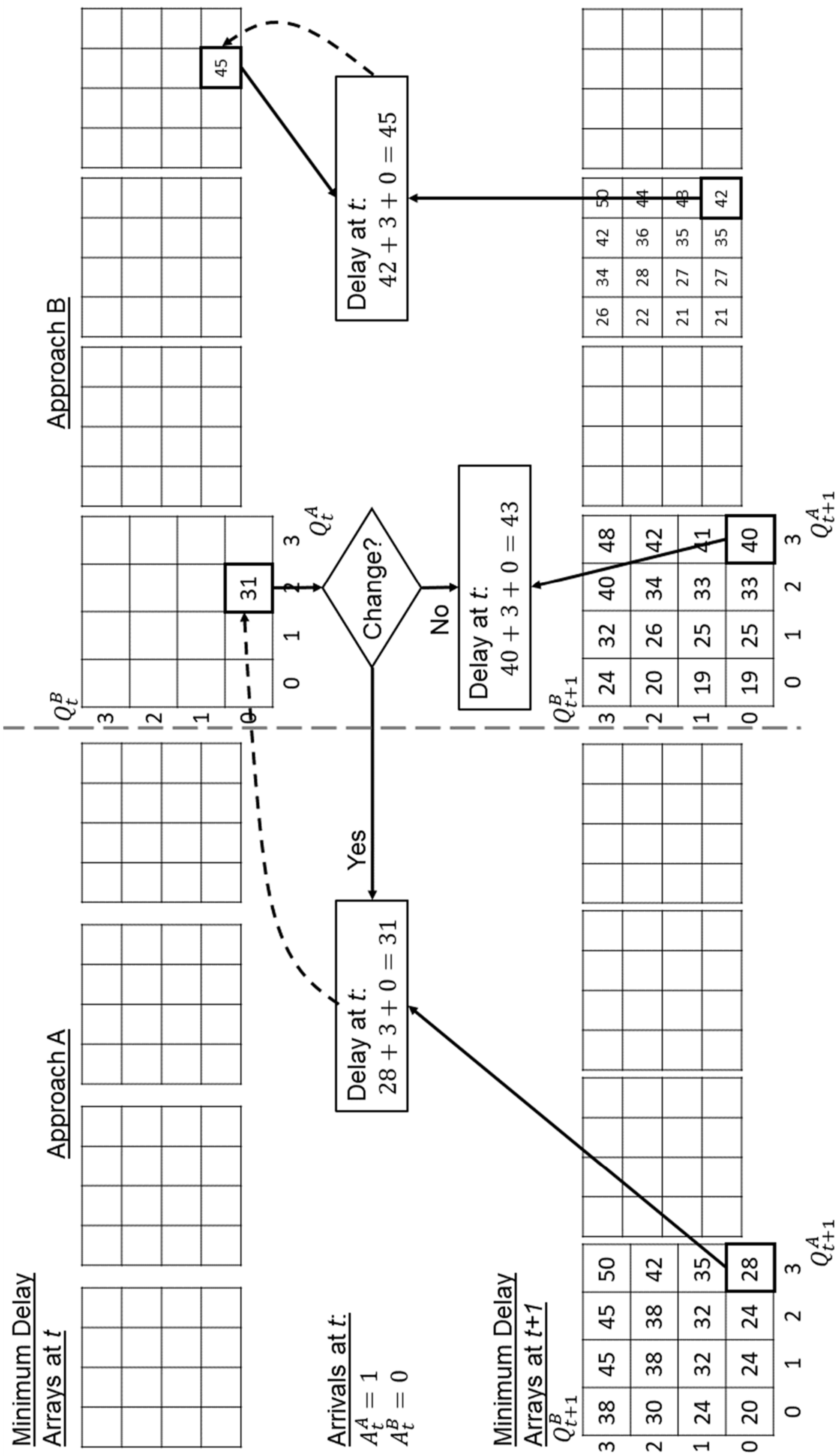

Fig 2. Extended states for boundary conditions.

By microscopic simulations, it is possible to generate the vehicle arrivals in advance and to store them so that they are known to the optimization method in advance. The implementation allows for optimization horizons of up two 3600 sec, which means that real world peak hours can be optimized according to their absolute optimum.

Nevertheless, in real systems, even in cooperative systems, vehicle arrivals will not be known in advance by every desire-able time. Hence, an approach is needed, that allows for restricted data availability of future arrivals. For that purpose, a rolling horizon approach is used according to CAI, 2009 and is shown in Fig 3.

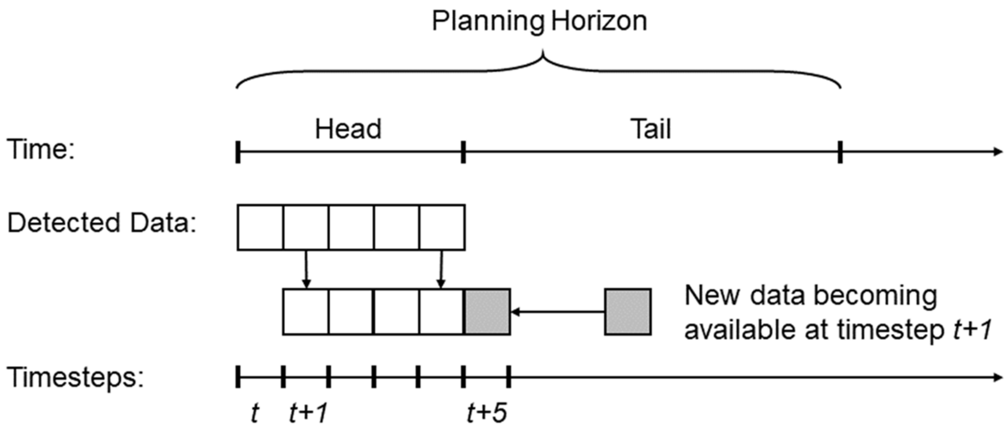

Fig 3. Rolling Horizon Approach [CAI, 2009]

This rolling horizon approach distinguishes the optimization horizon into a head and a tail part. For the head part, the data of the future is available through the driving time between detection and stop-line respectively end of queue. The vehicle arrivals in the tail of the optimization horizon are predicted based on detections.

Applying the resulting algorithm to the T-Shaped intersection used in RiLSA, state $6{,}6 * 10^{12}$ state variables would be needed based on a horizon of 30 time steps, three stages and twelve states and 25 lanes which is computational not feasible.

A reduction of complexity is gained by following simplification:

- Reduction of the horizon to only 2 time steps
- Usage of only one decisive lane per stage

Please note that there is no reduction in the number of states, the main contribution of this work, as they take care about modeling the control consequences of safety critical timings.

Results

Finally, the developed online and offline algorithms are evaluated by testing them against optimized fixed-time control, DYPIC which represents the algorithm according to ROBERTSON and BRETHERTON and the adaptive control method smartCarpet which is an advanced version of Sitraffic Motion, intended for the use in co-operative systems (see Fig. 4).

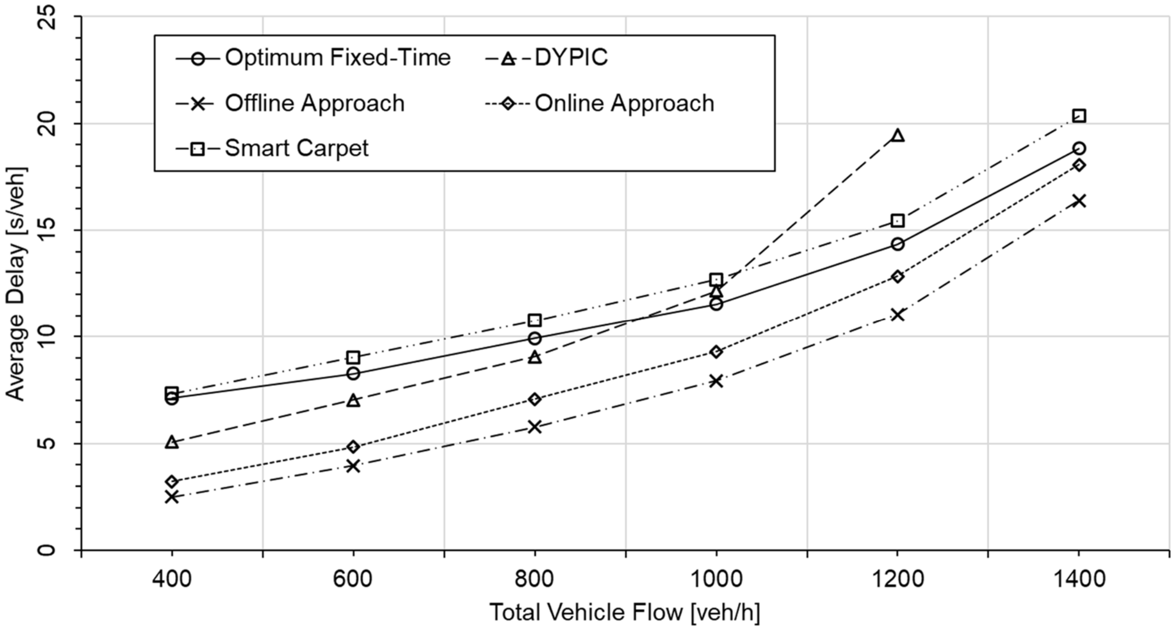

Fig. 4. Results in terms of delay

In these simulations six different traffic pattern are evaluated with the total traffic flow between 400 and 1400veh/hour. The mean arrival rates are kept constant with only stochastic variations in arrivals.

In all cases, the offline optimization approach yields the best results, since it represents the global optimum considering inter-green times. DYPIC, which also optimizes for the total time horizon, yields a higher delay. The growth in delay is caused by the fact that DYPIC does not consider inter-green times properly. Interestingly the online method which uses and horizon of two time intervals respectively four seconds only is very close to the offline method. An explanation may be that the arrivals only vary in terms of stochastics, and hence control variations within four seconds may be sufficient. This would also explain, that smartCarpet is always worst, since it tries to follow the stochastic variations in a macroscopic way, which misleads its optimization result. Therefor another simulation based on a three-leg T shaped intersection will be presented in the full paper.