Technical report 06-013

Traffic adaptive control of a single intersection: A taxonomy of approaches

R.T. van Katwijk, B. De Schutter, and J. Hellendoorn

If you want to cite this report, please use the following reference instead:
Traffic Adaptive Control of a Single Intersection: A Taxonomy of Approaches

R.T. van Katwijk* B. De Schutter**
J. Hellendoorn**

* Netherlands Organization for Applied Scientific Research, TNO, ronald.vankatwijk@tno.nl
** Delft University of Technology, Delft Center for Systems and Control {r.vankatwijk, b.deschutter, j.hellendoorn}@dcsc.tudelft.nl

Abstract: The design of a traffic adaptive control system is as well a science as an art. Along the way compromises have to be made in order to end up with a workable system that is not only able to come up with good signal timings, but is also able to deliver them on time. In this paper we propose a taxonomy of the various traffic adaptive control algorithms based both on their underlying principles and the compromises that were made to come up with a workable, albeit less optimal system.

Keywords: Traffic, Adaptive, Dynamic Programming, Branch-and-bound, Optimization, Simulation

1. INTRODUCTION

A common function of a traffic controller is to seek to minimize the delay experienced by vehicles through manipulation of the traffic signal timings. There are various levels of sophistication in traffic signal control system applications.

Basically, the modes of operation can be divided into three primary categories:

**Pre-timed** Under pre-timed operation, the timing plan is based on predetermined rates. These predetermined rates are determined from historical data. Pre-timed control frequently results in inefficient use of intersection capacity because of the inability to adjust to variations in traffic flow and actual traffic demand; this inefficiency is pronounced when flows are substantially below capacity.

**Responsive** In traffic responsive mode, signals receive inputs that reflect current traffic conditions, and use this data to choose an appropriate timing plan from a library of different plans. In traffic responsive mode, the signal timing plan responds to current traffic conditions measured by a detection system. The general traffic responsive strategies in use are either selection of a background signal timing plan based on detector data, or online computation of a background timing plan. The computation time interval may range from one cycle length to several minutes.

**Actuated** Vehicle-actuated controllers operate in real-time by applying a control in response to the current traffic state. An actuated controller operates based on traffic demands as registered by the actuation of vehicle and/or pedestrian detectors. There are several types of actuated
Traffic adaptive systems are currently the most advanced and complex control systems available. Traffic adaptive systems apply an optimization algorithm in real-time to create optimal signal timings. They differ from vehicle-actuated controllers because they incorporate decision making. That is, the system evaluates a set of feasible control actions and chooses an action that is optimal with respect to its current objectives. As traffic adaptive systems incorporate information from further upstream in their decision making (i.e. predicted arrivals) traffic actuated controllers are considered to be myopic (i.e. shortsighted) with respect to their control actions.

With recent advances in communication network, computer, and sensor technologies, there is increasing interest in the development of traffic adaptive signal control systems. Numerous systems have been proposed including PRO-DYN (Henry et al., 1983), (Barriere et al., 1986), (Henry and Farges, 1989), UTOPIA-SPOT (Mauro, 1990), OPAC (Gartner, 1983), (Gartner et al., 1995), (Gartner et al., 1999), RHODES (Sen and Head, 1997), (Mirchandani and Head, 2001), SPPORT (Dion and Hellinga, 2001), (Dion and Hellinga, 2002) and ALLONS-D (Porche et al., 1996), (Porche, 1998), (Porche and Lafortune, 2005). This overview is based on these references and the references contained therein.

The design of a traffic adaptive control system is as well a science as an art. Along the way compromises have to be made in order to end up with a workable system that is not only able to come up with good signal timings, but is also able to deliver them on time. Each adaptive system mentioned above has a different approach in dealing with the computational complexity of determining the best set of signal timings.

All adaptive systems therefore have their own specific strengths and weaknesses that make that system more - or - less suited for particular networks and traffic demand patterns. In this paper we propose a taxonomy of the various traffic adaptive systems based both on their underlying principles and the compromises that were made to come up with a workable, albeit less optimal system.

2. A TAXONOMY OF APPROACHES

Traffic signal control essentially comes down to making the right decisions at the right time. As such the traffic signal control problem solved by all traffic adaptive systems can be formulated in the form of a general decision problem. This general decision problem in turn can be represented as a simple decision tree such as that shown in figure 1.

The root of a decision tree represents the current state \( s_i, s_i \in S \), where \( i \) is the current time index and \( S \) is the set of all states. States in the decision tree of figure 1 are represented by open circles and actions by solid circles. The cost involved in order to transition to the subsequent state, \( s_{i+1} \), when deciding for an action \( u_i \) is denoted by \( c_i \). In figure 1 there are two possible successor states depicted for each action which permits the specification of stochastic outcomes.

In general, the nodes of a search tree represent choices. These choices are mutually exclusive and therefore partition the search space into two or more simpler sub-problems. At each time step, the controller observes the system’s current state \( s_i \), and selects a control action, \( u \in U_i \), where \( u \) is the action and \( U_i \) is the finite set of actions available to a controller in state \( s_i \). When the controller chooses an action \( u \in U_i \), the cost incurred by taking that action and subsequently transition to state \( s_{i+1} \) with probability \( p_{i,j}(u) \), is denoted by \( c(i) \). The objective of a traffic adaptive system is to find an optimal sequence of actions.

Looking at the various traffic adaptive systems we can discern the following features on which they differ:

- the optimization method. Is the optimal sequence of actions found by searching the decision tree using a rule-based method, or an
3. OPTIMIZATION METHOD

The objective of the system is to operate such that the total cost over the entire planning horizon is minimized. Thus, the task of the controller is to obtain a sequence of control actions \([u_0, u_1, \ldots, u_T]\), also referred to as a policy or control trajectory, such that the expected cost is minimized. In the case of an infinite planning horizon, a discount factor, \(\gamma < 1\), is typically applied to future costs to obtain a finite estimate of the cost-to-go from the current state \(i\), denoted by \(f(i)\).

The optimal cost-to-go value, denoted by \(f^*(i)\), is a function of the immediate cost of applying the control plus the expected cost-to-go from the subsequent state, a relationship encapsulated in the following recursive expression which is also known as Bellman’s Equation.

\[
f^*(i) = \min_{u \in U(i)} \left\{ c_i(u) + \gamma \sum_{j \in S} p_{i,j}(u)f^*(j) \right\} \quad (1)
\]

As the decision space has a tree-like structure, the search for the optimal sequence of decisions corresponds to building the tree. An exhaustive search of the entire decision space results in a full tree being built. Since search space size grows exponentially with problem size, it is not possible to explore all assignments except for the smallest problems. The only way out is to not look at the whole search space. Efficiency in searching the decision space is considered by the degree to which the entire tree will not have to be built to find an optimal path. In (Shelby, 2004) several well-known algorithms are assessed based on computational speed and on the quality of the results (in terms of delay).

Although the resulting state of traffic is subject to probabilistic outcomes as depicted in figure 1, all adaptive systems reviewed have chosen to employ a deterministic model. Due to this simplification, decision tree diagrams corresponding to the traffic signal control problem will generally neglect the use of explicit actions nodes as each action results in only one possible next state.

Dynamic programming and branch-and-bound (and combinations thereof) are the techniques that are predominantly used in traffic adaptive systems.

3.1 Dynamic Programming

The applicability of the approach depends on the opportunities for state aggregation within the decision tree. The strength of dynamic programming is that it can prevent that optimal solutions to subproblems it has already solved are recomputed. In order to do this, the solutions to already solved problems are saved. This approach is called memoization (not memorization, although this term also fits). It is however only possible to reuse a previous solution when states and thus the corresponding subproblems can be considered equal. This is why approaches like RHODES and PRODYN use an approximate state equivalence relation to attain greater efficiency. Figure 2(b) shows how dynamic programming can assist in not having to build the entire search tree as depicted in 2(a). RHODES, PRODYN, OPAC and SPPORT all employ dynamic programming as their method of optimization.
3.2 Branch-and-Bound

Branch-and-bound is a general method for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It belongs to the class of implicit enumeration methods. One way to do this is by proving that certain areas of the space contain no solutions. The core of the approach is the simple observation that (for a minimization task) if the lower bound for a sub-problem A from the search tree is greater than the upper bound for any other (previously examined) sub-problem B, then A may be safely discarded from the search. This is the bounding-part of the branch-and-bound approach. Figure 2(c) shows how branch-and-bound can assist in not having to build the entire search tree depicted in figure 2(a). Of the adaptive systems reviewed only ALLONS-D and SPOT employ the branch-and-bound method in its pure form. RHODES employs a hybrid system in which branch-and-bound techniques are applied within a dynamic programming framework.

In order to obtain a tight upper bound an initial path must be established through the search tree for which it is most likely to obtain a good solution. This involves that initially parts of the search space that are unlikely to contain good solutions are ignored. This is done by using heuristics. Heuristics are used to explore promising areas of the search tree first. This can be done by using problem specific knowledge (often borrowed from current practices in tuning traffic responsive and vehicle-actuated controllers) or by reusing information gained from previous optimizations.

4. ACTION SPACE

The width of the tree to be searched is dependent on the number of decisions that can be made at each point in time. In its simplest form the choice available is that between extending the current phase or switching to the next phase. This is the approach taken in OPAC, ALLONS-D, PRODYN and SPPORT. Although this approach significantly reduces the number of options to consider, it does not allow arbitrary phase sequencing. In its most elaborate form the choice available is that between phases. This approach allows the arbitrary sequencing of phases but comes at a cost in the width of the search tree. This is the approach chosen by UTOPIA-SPOT. Both of these approaches are shown in figure 3.

A compromise between these two extremes is found in allowing phase skipping. When the skipping of phases is allowed any phase sequence can be attained. This is shown in figure 4. This is the approach taken in the COP-system. The downside of this approach is that when the initial phase sequence is chosen wrong the gain in width is counteracted with an increase in tree depth. This is not strictly necessary however. It is in principle possible to choose the next phase dynamically.

5. PLANNING HORIZON

Traffic adaptive systems employ a traffic model to evaluate alternative traffic signal timings over a planning horizon. The length of the planning horizon as well as how the horizon is split up into successive intervals differs between each adaptive system. Typically however the horizon has a fixed length (of typically 1 to 2 minutes) and is subdivided into fixed intervals. From their descriptions we can deduce that OPAC, PRODYN, SPPORT, and ALLONS-D all use or have used 5-second time-steps.

If the horizon is chosen to short and the optimization algorithm is faced with the choice whether to a) completely serve a phase dispersing at a slow rate, or b) preempt that phase in order to switch to a phase with a higher dispersion rate, it would counter-intuitively chose for the latter. This is why many of the adaptive systems that employ shorter horizons have introduced terminal
6. UPDATE FREQUENCY

Traffic adaptive systems rely on predicted arrivals. As the distance over which these arrivals are predicted increases, the reliability of these predictions often decreases. This is why a rolling horizon is often applied. The concept of a rolling horizon originated in operations research and is used to determine a short-term policy based on a longer-term analysis. All adaptive systems reviewed that depend on arrival predictions employ the concept of a rolling horizon. These algorithms implement only the first (few) action(s) of the control plan after which a new optimization is performed.

The rolling horizon concept is visualized in Figure 6, where each horizontal bar denotes the calculated control plan for a decision horizon. The system commits to this control plan until the optimization process refreshes it. The period between updates is denoted by the commitment period.

7. DELAY MODEL

All adaptive systems reviewed consider individual vehicles in determining the control delay brought about by a chosen control plan. In that respect, all adaptive models can be considered to use a microscopic model. However, because the delay model is applied many times when exploring the search tree, all models have to make some sacrifices with respect to the level of detail on the employed model.

Known, commercially available, microscopic simulation models like Paramics, VISSIM, and AIM-SUN are unfit for use in real-time optimization. This is why simple event-based and cellular automaton models are predominantly used within adaptive systems. At first, these models employed vertical queuing models, but many adaptive systems have since switched to using horizontal queuing models so that queue spill back to upstream
intersections can explicitly be considered in the optimization.

8. CONCLUSIONS

The previous has shown that there are many different ways to configure a traffic adaptive system. Although the core of each of the traffic adaptive systems reviewed is based on the idea of finding a short term policy on the basis of a long term analysis, they differ with respect to the search algorithm applied, the length and resolution of the planning horizon, the update frequency and the delay model used.

Unfortunately computational boundaries still prevent the configuration of a traffic adaptive system in which no compromises have to be made in order to end up with a workable system that is a) able to come up with good signal timings and b) is able to deliver them on time.

As the base performance of an adaptive system is at least as good as that of an actuated controller there are considerable advantages to the deployment of an adaptive system. However, in order to gain the full advantage of traffic adaptive control, the system should be carefully tuned. Computationally complexity, geometry of an intersection, and demand patterns should be considered.

9. ACKNOWLEDGEMENTS

Research funded by the TU Delft spearhead program “Transport Research Centre Delft: Towards Reliable Mobility”, the BSIK project “Towards Sustainable Mobility (TRANSUMO)” and the TNO spearhead program “Sustainable Mobility Intelligent Transport Systems (SUMMITS)”

REFERENCES


