Traffic Light Control by Reinforcement Learning

Henrik Sejer Pedersen
Supervisor: Marco Chiarandini

Tuesday 27th November, 2018

Contents

Abstract

1 Introduction

With a steady increase in urbanization on top of already existing traffic problems in urban areas, the need for improving the traffic flow becomes ever more apparent. In urban areas, it is likely to be very costly if at all feasible to tear up existing infrastructure to accommodate better roads, so we turn our focus to the dynamic element present in most urban areas today, traffic lights.

Traffic lights provide an opportunity to influence the traffic flow in urban areas utilizing software, be it simple fixed-time, scheduled light changes or sophisticated traffic-reactive systems.

In this project, we look into the online, adaptive optimization strategies through reinforcement learning. Some of the early adaptive approaches SCAT[1] which was implemented in Sydney around 1980 and SCOOT[2], which is in extensive use to this date, focus on the coordination of multiple intersections, often known as *urban traffic control.* In Robertson and Bretherton [2], the coordination serves to shift traffic light cycles such that average queue lengths and vehicle stops are minimized, but still rely on the individual traffic light cycle to be efficient.

In this project, we seek to explore the application of reinforcement learning to control the individual traffic light, using only existing infrastructure.

At a very high level, reinforcement learning consist of an *agent* placed into some *envrionment*, where it takes actions and receive some *reward* based on those actions. Over time, the goal of the agent is to learn which actions to take in what situations to maximize the expected future reward. We will only cover a small part of reinforcement learning in this project, for a wider overview refer to Sutton and Barto [3].

The application of reinforcement learning to the field of traffic light control has been examined many a time[4][5][6][7][8][9][10][11]. However, due to the data made available by swarco Danmark A/S, we can provide further insight into the details of applying reinforcement learning to traffic light control based on existing infrastructure and information.

Thorpe and Anderson [4] showed great potential of reinforcement learning applied to traffic light control with their implementation of the SARSA algorithm on three different complete-knowledge state representations on a grid of 4x4 intersections, approaching the performance of an optimal policy.

Wiering [5] presented three different state representations for the multi-agent problem, where varying degrees of global information was included in the states. He found that more global information did not necessarily mean better performance, but some global information did perform better than none.

Abdulhai et al. [6] implemented Q-learning which is a temporal-difference algorithm, on a single-agent basis, with comments on ideas to improve it for multi-agent use. They used state information which could be estimated from detector data, in the form of queue lengths and elapsed phase time. The reward used is a penalty in the form of total vehicle delay between decision points. Additionally, they made use of a neural-network-like structure (CMAC) to approximate the value function, which provided some generalization and should, to some extent, improve behavior in unobserved states.

Arel et al. [7] used an actual neural network for value function approximation, which should perform even better in unobserved states compared to the CMAC. The state used is also significantly improved, as it incorporates only what they refer to as *relative traffic flow*, calculated for every upstream lane as traffic flow in that lane divided by average traffic flow of all upstream lanes of the intersection. Such a state representation can be implemented using detector data only. The reward is based on average delay, giving a negative reward if an action increases the average delay, but a positive reward if it is decreased.

2 Instance definition

In this section, we introduce a traffic light located in southern Odense, Denmark and its implementation in the microscopic traffic simulator SUMO – *Simulation of Urban MObility*.

2.1 Network topology

The company SWARCO provided the data and network information we use in this project. We look at a traffic light controlled intersection located in southern Odense, Denmark, depicted in appendix A.1. We modeled the intersection in the microscopic traffic simulator SUMO, without the inclusion of bicycles, as the available data on bicycles is quite limited.

2.1.1 Induction Loops

In the network, a number of vehicle detectors are present, in the form of *induction loops*. In appendix A.1, the locations of induction loops relevant to the intersection is visualized. Figure 1 describes the meaning of the visualization.

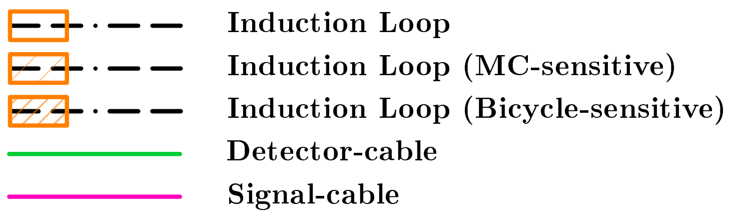

Induction Loop
Induction Loop (MC-sensitive)
Induction Loop (Bicycle-sensitive)
Detector-cable
Signal-cable

Figure 1: Induction loop signatures

The induction loops in the intersection cover vehicles quite well, but are quite lacking when it comes to bicycles, for this reason, bicycles are excluded from the project.

2.1.2 Traffic light movements

Appendix A.2 visualizes the traffic signal groups in the intersection. Table 1 describes the traffic signal strings on the drawing. A single signal can control multiple of these movements, denoted by, for instance, A1+A1v denotes a single signal controlling both left-turn and straight movements of A1.

todo: better string representation

string	meaning
A/B/C/. . . #	Upstream movement following road
A#Cy	Straight-ahead bicycle movement from A#
A#v	Left-turn movement from A#
A#h	right-turn movement from A#

Table 1: Explanation of traffic light movement strings for appendix A.2

2.2 SUMO Introduction

SUMO, short for *Simulation of Urban MObility*, is a microscopic, well-established general-purpose traffic simulator. It has been around since 2001 and is an open source project. What makes it particularly interesting for this project, is the ability to control elements of the simulation externally, through a well-defined API. With this API, gathering information from the simulation for our agent is made simple, while also providing a way for our agent to control each traffic light.

2.3 Implementing the intersection in SUMO

SUMO uses a directed graph representation to define a traffic network, with some extra information. *Nodes* in the graph are points connected by one or more *edges*. Nodes contain connection data, which is (potentially) a many-to-many mapping of incoming lanes to outgoing lanes. The edges carry the lane information, allowing any number of adjacent lanes (within computational reason) to follow any single edge between two nodes. As such, each edge defines a set of up- or down-stream lanes, possibly consisting of multiple traffic movements.

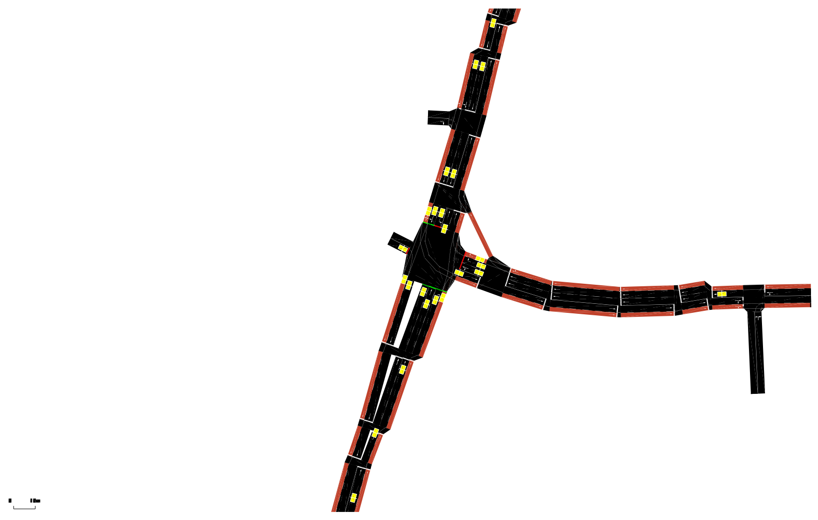


Figure 2: Intersection as implemented in SUMO, visualised with SUMO-GUI

todo: better image The primary discrepancies between the actual intersection and the one implemented in SUMO are the missing induction loop D18. The induction loop is not present in the SUMO implementation as SUMO only allows placing induction loops on lanes.

2.4 Vehicle data

Along with technical drawings of the intersection, files describing the traffic flow exist in the form of 5 minutes aggregated readings from the 25 induction loops present in the intersection. To use this data in a meaningful way, we define orderings of the induction loops, each of which defines a route in the network.

Populating these sets with the input data can be modeled as an integer linear programming problem, and draws many parallels with the generalized set covering problem. Unfortunately, the induction loop readings have proven non-perfect, so we propose two different models.

The first model assumes that the induction loops never miss any vehicles, but may overcount. The second model assumes that the induction loops do not overcount, but may miss vehicles.

Both models give a vehicle count for each route for a single 5-minute interval and are called once for each such period. Solving for shorter intervals increase the network load in the simulation, we use the second model in this project.

2.4.1 Overcounts, no misses

Given a set of routes R, a set of detectors D, a binary route representation as defined below by B_{ij}, and upper bounds for *total* number of vehicles passing a detector, defined below by C_i. Select the number of vehicles to follow each route, such that the number of vehicles passing each detector is maximized for all detectors, given the single constraint:

- No more than C_i vehicles can pass detector D_i, $\forall i \in \{1, 2, \dots, |D|\}$

For convenience we define the sets $I = \{1, 2, \dots, |D|\}$, and $J = \{1, 2, \dots, |R|\}$.

Let Bij be a binary constant such that:

$$B_{ij} = \begin{cases} 1 & \text{if route j passes detector i} \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in I, j \in J$$

Let C_i be an an detected number of vehicles at detector D_i, $\forall i \in I$ Let x_j be an integer variable with a lower bound of 0, indicating the number of vehicles following route $j, \forall j \; inJ$

$$\begin{array}{rrcll} \text{Maximize} & \sum_{j \in J} x_j \cdot \sum_{i \in I} B_{ij} & & & \\ \text{s.t.} & \sum_{j \in J} B_{ij} \cdot x_j & \leq & C_i & \forall i \in I \\ & x_j & \in & \mathbb{N} & \forall j \in J \\ & x_j & \geq & 0 & \forall j \in J \end{array}$$

The objective function maximizes the total sum of vehicles passing detectors. The outer sum sums over each route, and the second sum computes the number of detectors in the route from the outer sum.

The first constraint ensures that the sum of vehicles passing a detector does not surpass its capacity.

The second constraint ensures that the number of vehicles entering the simulation is integer.

The Third constraint ensures that the number of cars on each route must be non-negative.

2.4.2 Misses, no overcounts

The previous model can with a few changes be modified, such that the vehicle counts act as lower bounds rather than upper bounds. Treating vehicle counts as lower bounds will put a more significant strain on the network, as more vehicles enter the simulation, which is desired.

$$\begin{array}{rrcll}
\text{Minimize} & \sum_{j\in J} x_j \cdot \sum_{i\in I} B_{ij} & & & \\
\text{s.t.} & \sum_{j\in J} B_{ij}\cdot x_j & \geq & C_i & \forall i\in I \\
 & x_j & \in & \mathbb{N} & \forall j\in J \\
 & x_j & \geq & 0 & \forall j\in J
\end{array}$$

The objective of the new model is to minimize the number of vehicles passing induction loops, while enforcing that at least the observed number of vehicles pass.

It is important to note that in this model, $B_{i\cdot}$ must contain at least *one* 1 to be feasible for all i, whereas this was not the case for the previous model.

3 Introduction to reinforcement learning

Reinforcement learning is a machine learning paradigm dedicated to solving sequential decision processes. The definition of a reinforcement learning algorithm given by Sutton and Barto [3, chap. 3], is an algorithm which can solve a specific kind of sequential decision problem, namely those which can be described formally by a *finite Markov decision process.*

3.1 Finite Markov Decision Processes

The Markov decision process provides a formal description of sequential decision problems.

In this project, we define a Markov decision process by a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$, where $\mathcal{S}$ is the finite state-space, $\mathcal{A}$ is the finite action-space, $\mathcal{R} \subset \mathbb{R}$ is the finite reward-space, and the dynamics function $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, defined in eq. (1), describes the conditional probability of entering a state $s' \in \mathcal{S}$ with reward $r \in \mathcal{R}$ given the previous state was $s \in \mathcal{S}$ and action $a \in \mathcal{A}(s)$ was chosen.

$$p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \tag{1}$$

Some like to include γ, a *discount factor* in the definition of finite MDP. We, however, leave this definition to the individual reinforcement learning algorithm.

When solving Markov decision processes, a sequence of random variables is introduced. Let $T = \{1, 2, \ldots\}$ be a sequence of discrete time steps, S_t is then a random variable which indicates the state of the environment at time step $t \in T$, A_t is the action chosen at time step $t \in T$, from the set $\mathcal{A}(s)$ which denotes the actions available from state s. R_t is the reward received after choosing action A_{t-1} from state S_{t-1}

Figure 3 visualises the agent-environment loop. Initially, the agent is placed in some environment $\mathcal{E}$, from where the agent observes S_0. The agent the chooses some action

$A_0 \in \mathcal{A}(S_0)$, after the action has been performed, state S_{t+1} and R_t is presented to the agent, this results in a sequence of the form $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots$

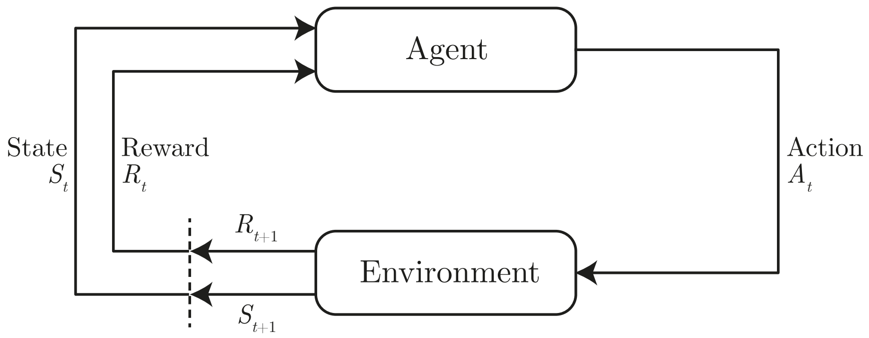


Figure 3: The Agent-Environment interface adapted from Sutton and Barto [3, chap. 3]

As such, the agent will learn the optimal *policy* π^* (or estimate thereof) by iteratively taking actions, and observing the following reward, and state of the environment.

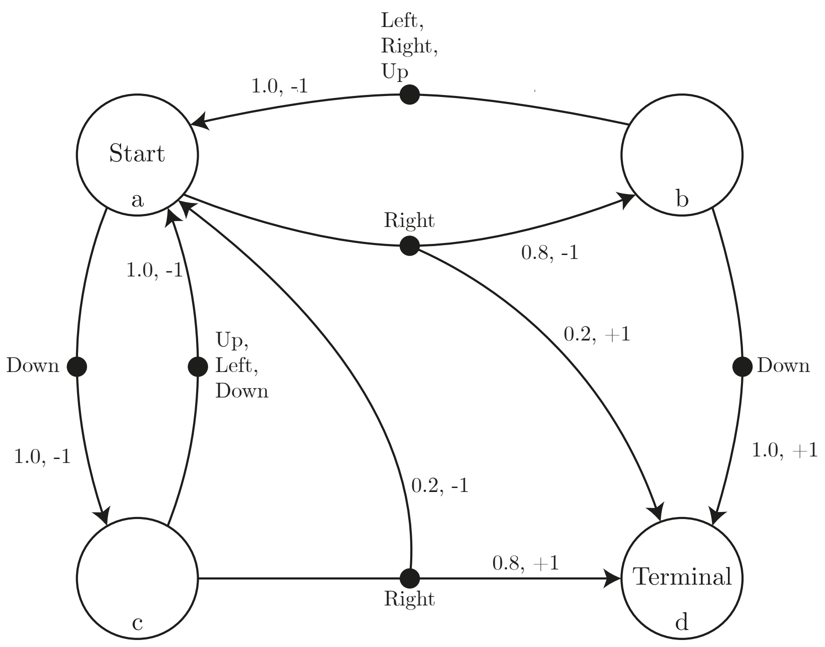


Figure 4: **TODO**

Visual MDP representation for small example?

Figure 5: MDP represented as a graph

3.2 RL

according to some policy π either *exploitatively* or *exploratively*.

TODO: standardize references

References

[1] A. G. Sims and K. W. Dobinson. The Sydney Coordinated Adaptive Traffic (SCAT) System Philosophy and Benefits. *IEEE Transactions on Vehicular Technology*, 29(2): 130–137, 1980.

[2] Dennis I. Robertson and R. David Bretherton. Optimizing networks of traffic signals in real time – the SCOOT method. *IEEE Transactions on Vehicular Technology*, 40 (1):11–15, Feb 1991. ISSN 0018-9545. doi: 10.1109/25.69966.

[3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

[4] Thomas L. Thorpe and Charles W. Anderson. Traffic Light Control Using SARSA with Three State Representations. Technical report, IBM Corporation, 1996.

[5] MA Wiering. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pages 1151–1158, 2000.

[6] Baher Abdulhai, Rob Pringle, and Grigoris J. Karakoulas. Reinforcement Learning for True Adaptive Traffic Signal Control. *Journal of Transportation Engineering*, 129 (3):278, 2003. ISSN 0733947X.

[7] Itamar Arel, Cong Liu, Thomas Urbanik, and A. G. Kohls. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 4(2):128–135, 06 2010.

[8] Bram Bakker, Shimon Whiteson, Leon Kester, and Frans CA Groen. Traffic light control by multiagent reinforcement learning systems. In *Interactive Collaborative Information Systems*, pages 475–510. Springer, 2010.

[9] Saad Touhbi, Mohamed A. Babram, Tri Nguyen-Huu, Nicolas Marilleau, Moulay L. Hbid, Christophe Cambier, and Serge Stinckwich. Adaptive Traffic Signal Control : Exploring Reward Definition For Reinforcement Learning. *Procedia Computer Science*, 109:513–520, 2017.

[10] Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Computing Surveys (CSUR)*, 50(3):1–38, 2017.

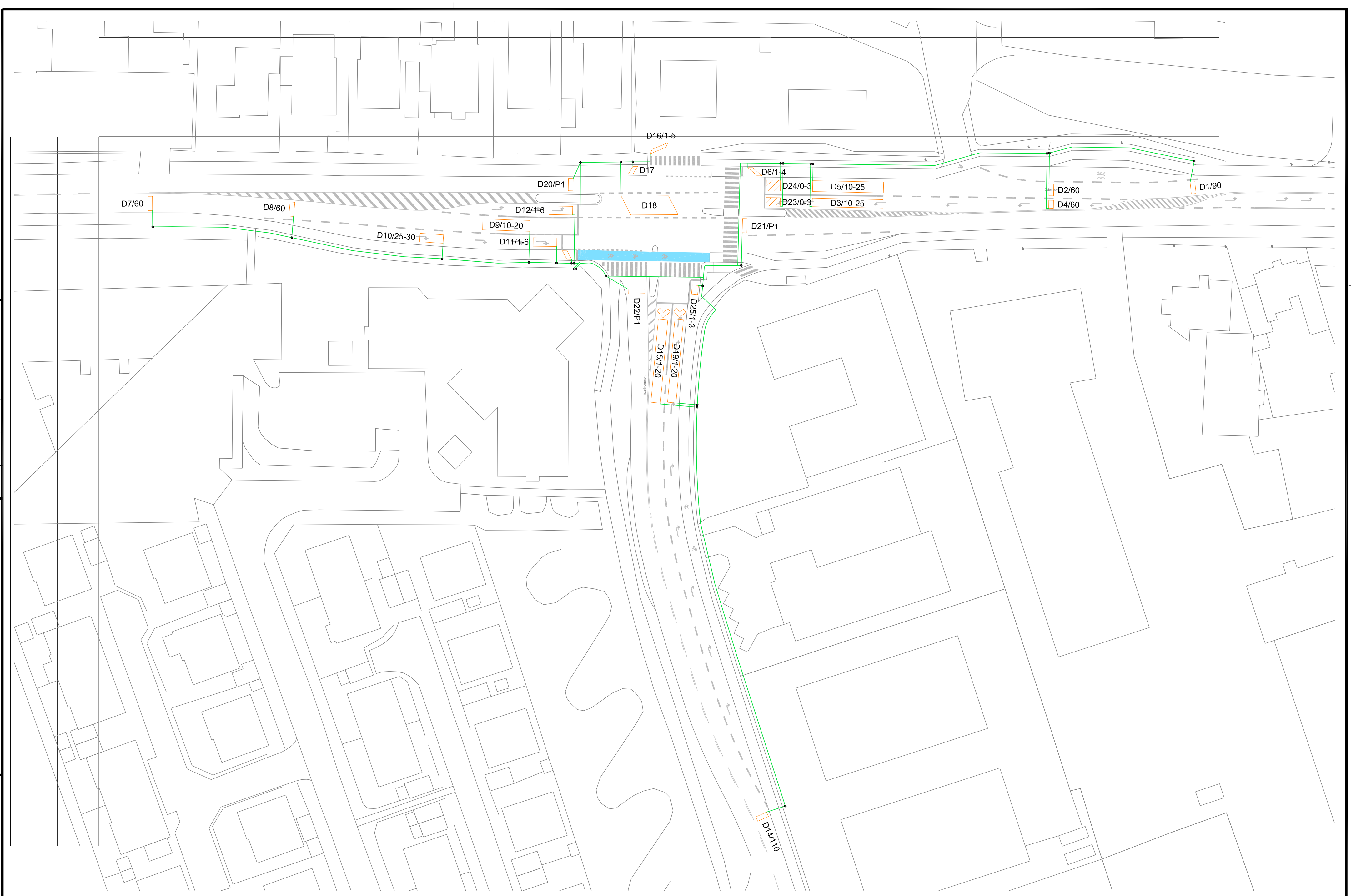
[11] Wade Genders and Saiedeh Razavi. Evaluating reinforcement learning state representations for adaptive traffic signal control. *Procedia Computer Science*, 130:26–33, 2018.

[12] Volodymyr Mnih, Adrià P. Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *CoRR*, abs/1602.01783, 2016. URL http://arxiv.org/abs/1602.01783.

[13] Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Comput. Surv.*, 50(3):34:1–34:38, June 2017. ISSN 0360-0300. doi: 10.1145/3068287. URL http://doi.acm.org/10.1145/3068287.

A Intersection Technical Drawings

A.1 Technical drawing of intersection induction loops



Rev.1	Rev.2	Rev.3	Rev.4	Rev.5	Rev.6

Rev.7	Rev.8	Rev.9	Rev.10	Rev.11	Rev.12

SWARCO DANMARK A/S
Tonsbakken 16 - 18
DK 2740 Skovlunde
Telefon 36 88 88 88
Telefax 36 88 88 00

Odense Kommune
Odensevej - Landbrugsvej

Oversigtsplan

PROJ.	TEGN.	DATO
CJ	FJ	2017.0105

KUNDENR.

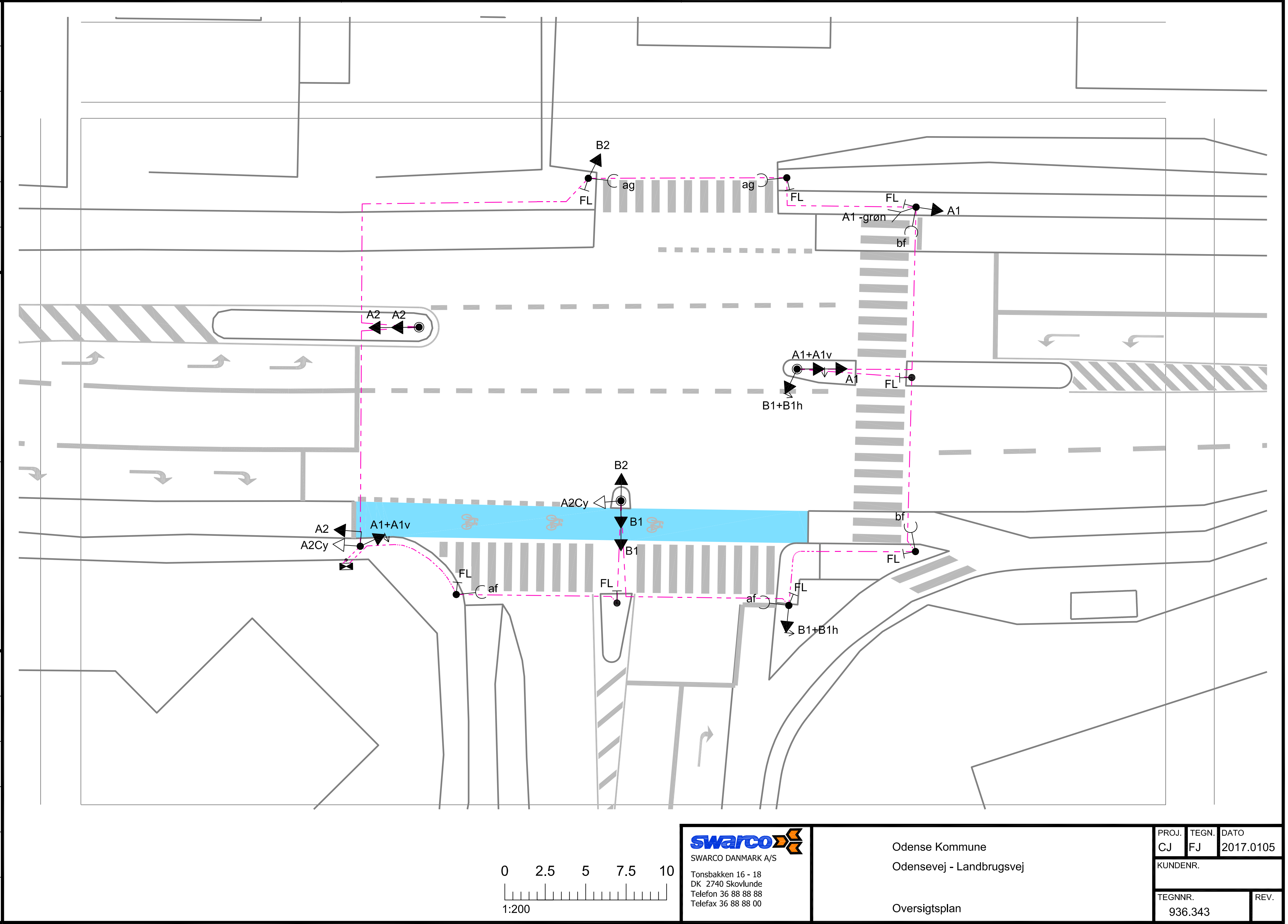
TEGNNR. 936.343

REV.

A.2 Technical drawing of intersection movements

Rev.1	Rev.2	Rev.3	Rev.4	Rev.5	Rev.6

Rev.7	Rev.8	Rev.9	Rev.10	Rev.11	Rev.12



SWARCO DANMARK A/S

Tonsbakken 16 - 18
DK 2740 Skovlunde
Telefon 36 88 88 88
Telefax 36 88 88 00

Odense Kommune

Odensevej - Landbrugsvej

Oversigtsplan

PROJ.	TEGN.	DATO
CJ	FJ	2017.0105
KUNDENR.		
TEGNNR. 936.343		REV.