Traffic Light Control by Reinforcement Learning

Henrik Sejer Pedersen
Supervisor: Marco Chiarandini

Tuesday 30th October, 2018

Contents

1 Introduction

1.1 Literature review

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 SWARCOprovided 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. 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

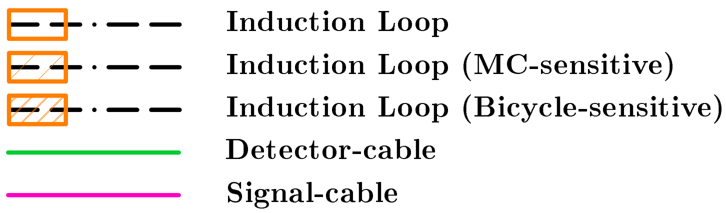


Figure 1: Induction loop signatures

2.1.2 Traffic light movements

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.

picture of SUMO network implementation

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, $\forall i \in \{1, 2, \ldots, |D|\}$

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

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

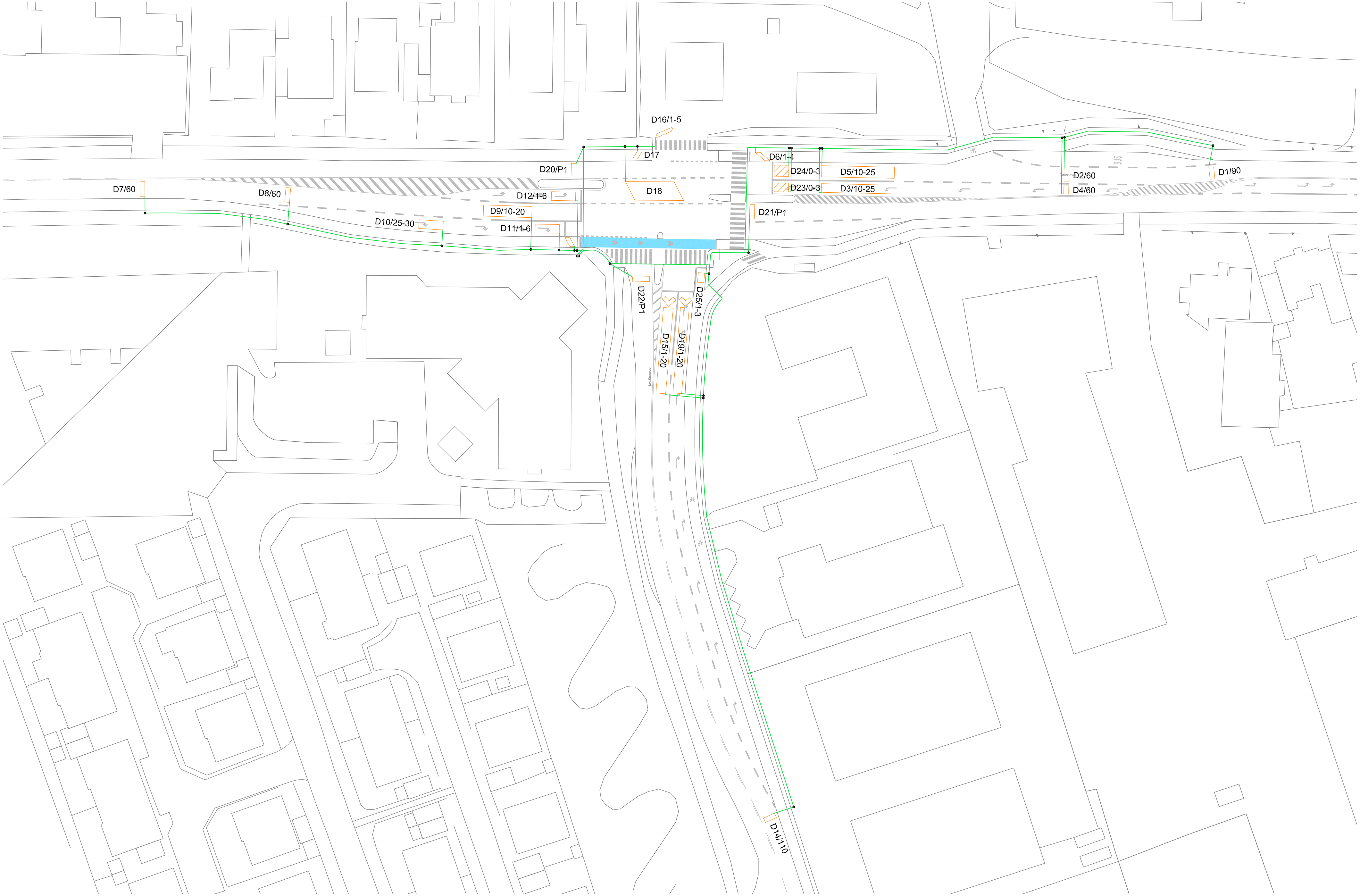
The objective function maximizes the total sum of vehicles to insert, by multiplying the number of vehicles on route some route, by the sum of detectors it passes.
The first constraint ensures that the sum of vehicles passing a detector does not surpass its capacity.
The second constraint ensures that 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

A Intersection Technical Drawings

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

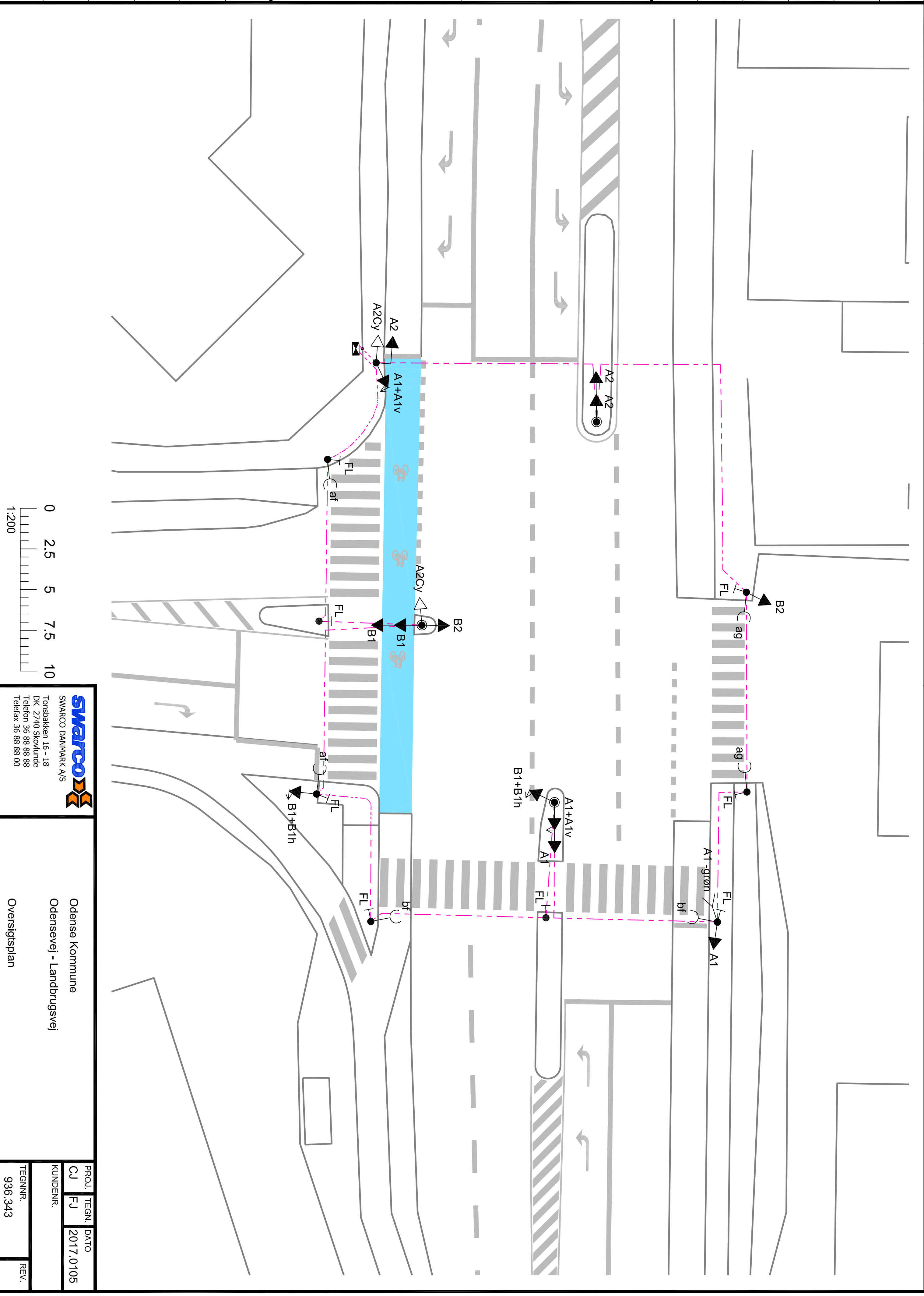
Odense Kommune

Odensevej - Landbrugsvej

Oversigtsplan

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

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
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.

References