Stochastic Modelling and Optimisation - Final Project

Monika, Jordi and Sebastian

March 30, 2019

1 Introduction

Nobody likes traffic. Traffic decreases the productivity of urban spaces, and translates into lower prosperity. Controlling the amount of traffic is therefore a priority goal for urban planners. Yet, most cities face natural or financial constraints in the expansion of transport networks. Consequently, urban planners are often faced with the challenge of optimising traffic flows within the constraints of existing infrastructure.

In this project we study how dynamic programming can be applied to optimise traffic flows.

2 DP formulation

2.1 Variable definitions

Links (approaches):	$z \in Z$
Junctions:	$j \in J$
Set of inflowing links:	I_j
Set of outflowing links:	O_j
Cycle time (We assume $C_j = C$ for all junctions) :	C_j
Total lost time:	L_j
Set of stages:	F_j
Set of stages where link z has r.o.w.:	v_z
Saturation flow for link z:	S_z
Turning rates for inflowing link z and outflowing link w:	$t_{z,w}$
Green time of stage i at junction j:	$g_{j,i}$
Control interval:	T
Period intervals:	$[kT, (k+1)T]$
Demand flow:	d_z
Exit flow:	s_z

2.2 Constraints and indentities

Green light constraints:	$g_{j,i} \in [g_{j,i,\min}, g_{j,i,\max}]$
Cycle time constraint:	$\sum_{i \in F_j} g_{j,i} + L_j = C$
Exit flow constraint:	$s_z(k) = t_{z,0} q_z(k)$
Inflow to link z:	$q_z(k) = \sum_{w \in I_M} t_{w,z} u_w(k)$
Outflow from link z:	$u_z = \begin{cases} S_z & \text{if has r.o.w} \\ 0 & \text{otherwise} \end{cases}$
(Assuming space available in downstream link and $x_z > S_z$	
Average value for outflow from z:	$u_z(k) = S_z G_z(k)/C$
Effective green time:	$G_z(k) = \sum_{i \in v} g_{j,i}(k)$
Steady state demand:	$(1 - t_{z,0})\, q_z^N + d_z^N - u_z^N = 0$
(Assuming nominal green times that lead to steady-state link queues under non-saturating constant nominal demand)	

2.3 DP equations

Dynamics:

$$x_z(k+1) = x_z(k) + T\left[q_z(k) - s_z(k) + d_z(k) - u_z(k)\right] \tag{1}$$

Substituting gives:

$$x_z(k+1) = x_z(k) + T\left[(1 - t_{z,0}) \sum_{w \in I_M} \frac{t_{w,z} S_w \left(\sum_{i \in v_w} \Delta g_{M,i}(k)\right)}{C} + \Delta d_z(k) - \frac{S_z \left(\sum_{i \in v_z} \Delta g_{N,i}(k)\right)}{C}\right] \tag{2}$$

In vector notation:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{g}(k) + \mathbf{T}\Delta\mathbf{d}(k) \tag{3}$$

where $\mathbf{x}$ is the state vector of the numbers of vehicles $\mathbf{x}_z$ within links $z \in Z$ and $\Delta g_{j,i} = g_{j,i} - g_{j,i}^{\mathrm{N}}$ is the vector of deviations from the steady-state green times and $\Delta d_z = d_z - d_z^{\mathrm{N}}$ is the deviation from the steady-state demand flows. $\mathbf{A} = \mathbf{I}$, $\mathbf{B}$ and $\mathbf{T}$ are the state, input, and disturbance matrices, respectively. The input matrix B reflects the specific network topology, fixed staging, cycle, saturation flows, and turning rates. For this problem we assume demand is in its steady-state: $\Delta \mathrm{d}(k) = 0$.

Cost:

$$\mathcal{J} = \frac{1}{2} \sum_{k=0}^{\infty} \left(\|\mathbf{x}(k)\|_{\mathbf{Q}}^2 + \|\Delta\mathbf{g}(k)\|_{\mathbf{R}}^2\right) \tag{4}$$

Here $\mathbf{Q}$ and $\mathbf{R}$ are non-negative definite, diagonal weighting matrices. The infinite sum in the equation (4) suggest an infinite time horizon, which is taken in order to obtain a time-invariant feedback law according to the LQ optimisation theory. Intuitively, the first term in (4), $\|\mathbf{x}(k)\|_{\mathbf{Q}}^2$ is responsible for minimisation and balancing of the relative occupancies of the network links. $\mathbf{Q}$ is defined as a diagonal matrix with the diagonal elements equal to the inverses of the storage capacities of the corresponding links. Lastly, matrix $\mathbf{R}$ influences the magnitude of the control reactions. Therefore, $\mathbf{R} = r\mathbf{I}$, where the choice of r is picked to ensure the best results for a given application network (i.e. simple trial-error procedure).

Solution:

The aim of this problem is to minimise the cost presented in equation (4) subject to constraints in (3). Having recognised the problem as a discrete-time linear quadratic control problem with and infinite-horizon subject to state equation (3), one can make use of an algebraic Riccati equation to find the solution.

We define a symmetric positive definite **cost-to-go matrix X** evolving backwards in time from $X_T = Q$ according to:

$$X_{k-1} = Q + A^T X_k A - A^T X_k B \left(B^T X_k B + R\right)^{-1} B^T X_k A \tag{5}$$

which is known as the discrete-time dynamic Riccati equation of this problem. The steady-state characterization of X, relevant for the infinite-horizon problem in which T goes to infinity, can be found by iterating the dynamic equation repeatedly until it converges; then X is characterized by removing the time subscripts from the dynamic equation. Resulting in:

$$X = Q + A^T X A - \left(A^T X B\right) \left(R + B^T X B\right)^{-1} \left(B^T X A\right) \tag{6}$$

Solution to this problem is therefore given by a matrix (called the control) **L**:

$$\mathbf{L} = \left(B^T X B + R\right)^{-1} B^T X A \tag{7}$$

Putting it into DP framework that uses backwords induction, the optimal control solution at each time k is equivalent to:

$$\Delta g_k^* = -\left(B^T X_k B + R\right)^{-1} \left(B^T X_k A\right) x_{k-1} \tag{8}$$

Dropping the k's because of the infinite horizon assumption, the equation transforms to:

$$\Delta g^* = -\left(B^T X B + R\right)^{-1} \left(B^T X A\right) x_{k-1} \tag{9}$$

And can equivalently by written as:

$$\mathbf{g}(k) = \mathbf{g}^{\mathrm{N}} - \mathbf{L}\mathbf{x}(k) \tag{10}$$

where $\Delta \mathbf{g} = \mathbf{g}(k) - \mathbf{g}^{\mathrm{N}}$.

The optimasation of this network is not done yet. This is because the LQ problem presented above does not consider control constraints. Those, however, can be imposed after the solution to (9) is computed. Hence, this calls for another optimasation problem that is solved in real-time forfor each junction j so as to specify feasible green times $G_{j,i}$ that are closest in distance to the non-feasible regulator-based green times $g_{j,i}$ resulting from (9).

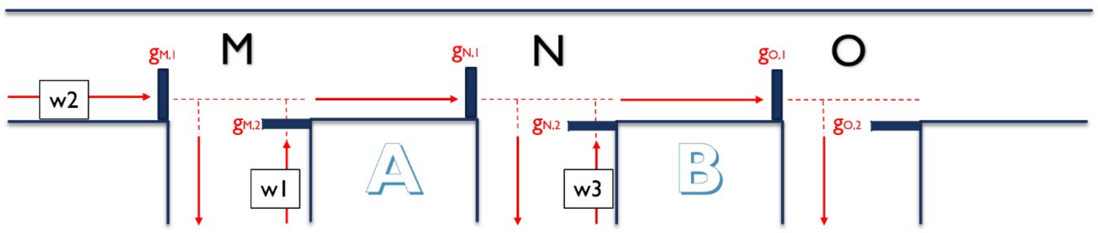
$$\min G_{j,i} \sum_{i \in F_j} \left(g_{j,i} - G_{j,i}\right)^2 \tag{11}$$

subject to

$$\sum_{i \in F_j} G_{j,i} + |L_j = C \tag{12}$$

$$G_{j,i} \in [g_{j,i,\min}, g_{j,i,\max}]\, \forall i \in F_j \tag{13}$$

Figure 1: Network Graph



$g_{M,1}$
M
$g_{N,1}$
N
$g_{O,1}$
O
w2
$g_{M,2}$
$g_{N,2}$
$g_{O,2}$
w1
A
w3
B