

Stochastic Modelling and Optimisation - Final Project

Monika, Jordi and Sebastian

March 30, 2019

1 Motivation

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 faced with the challenge of optimising traffic flows within the constraints of existing infrastructure, to achieve the most efficient use of it possible.

In this project we study how dynamic programming can be applied to optimise urban traffic flows. Specifically, we study the traffic response urban control (TUC) strategy initially proposed by Diakaki, Papageorgiou, & McLean (1999) to control traffic light signals. This strategy is based on a linear dynamic equation describing the number of cars present in a set of traffic link and a quadratic cost function, penalising the number of cars per link. This linear-quadratic system can be solved using infinite horizon dynamic programming. The strategy proceeds in two steps. First, a stationary policy prescribing green times to be used for any given state is computed off-line, using a set of parameters that describe the transport system, such as road capacity, cycle time, saturation flows and turning rates. Second, the dynamic policy responses are calculated for dynamically evolving state in a constrained minimisation problem, which ensures that the solution lies within a specified range of feasible green times, yet as close as possible to the optimal green times calculated in the first step.

The solution the strategy provides is not globally optimal, because constraints are applied only after solving for the optimal policy matrix. However, breaking up the problem in these two steps has the practical advantage that the dynamic system can be solved 'off-line' and does not need to be continuously re-evaluated. Minimising deviation from the 'ideal policy' is then trivial and can be decentralised (to a traffic light control), which makes it easier to apply the system in real time and in large networks. The strategy nonetheless yields a solution that is forward-looking in the sense that any policy takes into account the effect of a change in a particular part of the system on the rest of the system (Diakaki, Papageorgiou, & Aboudolas, 2001).

We simulate this strategy using a simple toy network of three junctions, connected by two main links, one of which is an 'upstream' link and one a 'downstream' link. The traffic flows only in one direction, from the upstream to the downstream link. We study how the strategy sets green times at the three junctions, and how it balances the benefit of allowing long green times to empty the upstream link with the resulting cost of potential downstream accumulation of cars. To gain a full understanding of the system dynamics, we vary parameters describing the infrastructure situation, such as road capacity, cycle time, saturation flow rates and turning rates and study the effect on the optimal green times chosen by the strategy.

We find that the transport network under our specification has [EXPAND with results, short preview of conclusion]

2 DP formulation

A transport network can be described as a set of links or approaches that connect a set of junctions. With respect to a junction, a link can be inflowing or outflowing, or both. The cycle time describes how long a full cycle of light changes takes at a given junction. Similarly, the total lost time describes how much of the cycle time is lost time, for example because all lights are red to allow pedestrians to pass. The number of stages divides the cycle time into a set of intervals during which a particular combination of links has right of way (r.o.w). The turning rates indicate the direction cars are turning into; they can be assumed to be an empirically established average. Saturation flows describe the capacity of a link; they can be assumed to be the maximum number of cars to pass the link during one control interval. The control interval can be set by the system administrator; it has to be at least as long as the shortest cycle time. The control interval determines how frequently green-light policies can be changed. If this interval is very long, then the system will react poorly to changing traffic situations. Demand and exit flows indicate whether cars enter the system or exit the system. Last but not least, green light times describe the length of time a particular link's signals give r.o.w. to a certain stage of the signal cycle. This is the policy variable in this problem. Section 2.1 provides the notation for the above described variables.

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 :	$t_{z,w}$
Control interval:	T
Period intervals:	$[kT, (k+1)T]$
Demand flow:	d_z
Exit flow:	s_z
Green time of stage i at junction j :	$g_{j,i}$
Number of cars in link z :	x_z

2.2 Constraints and identities

Several constraints have to be introduced to define a sensible transport network. Green times have an upper and lower bound, and cycle times are the sum of all green times for all stages plus the lost time. Exit flows for a given link are the product of the number of cars flowing into a link, and the turning rates indicating turns leading out of the network. Inflows to a given link are the sum of outflows from other links directed into the given link. Outflows are assumed to always occur at the maximum rate possible, whenever the respective stage has r.o.w.. If a given cycle includes more than one stage with r.o.w. for a given link, the sum of all green times for a given link are added to obtain the effective green time per cycle, which can in turn be used to calculate the average outflow from a given link per cycle. A steady state situation for a particular link is obtained when the outflows from the network equal the inflows plus the additional demand arising within the link (this can for example be parked cars entering the link, rather than cars entering from an inflowing link).

Green light constraints:

Cycle time constraint:

Exit flow constraint:

Inflow to link z :

Outflow from link z :

(Assuming space available in downstream link and $x_z > S_z$)

Average value for outflow from z :

Effective green time:

Steady state demand:

(Assuming nominal green times that lead to steady-state link queues under non-saturating constant nominal demand)

$$g_{j,i} \in [g_{j,i,\min}, g_{j,i,\max}]$$

$$\sum_{i \in F_j} g_{j,i} + L_j = C_j$$

$$s_z(k) = t_{z,0} q_z(k)$$

$$q_z(k) = \sum_{w \in I_M} t_{w,z} u_w(k)$$

$$u_z = \begin{cases} S_z & \text{if has r.o.w} \\ 0 & \text{otherwise} \end{cases}$$

$$u_z(k) = S_z G_z(k) / C$$

$$G_z(k) = \sum_{i \in v} g_{j,i}(k)$$

$$(1 - t_{z,0}) q_z^N + d_z^N - u_z^N = 0$$

2.3 DP equations

The number of cars in a given link z at the end of any given period k is the sum of the number of cars that remained in the link from the previous period, plus the cars that enter minus the cars that exit during the period. This yields the following dynamic equation:

Dynamics:

$$x_z(k+1) = x_z(k) + T [q_z(k) - s_z(k) + d_z(k) - u_z(k)] \quad (1)$$

Substituting gives:

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

In vector notation:

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

where \mathbf{x} is the state vector of the numbers of vehicles \mathbf{x}_z within links $z \in Z$ and $\Delta g_{j,i} = g_{j,i} - g_{j,i}^N$ is a vector of deviations from steady-state green times and $\Delta d_z = d_z - d_z^N$ 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 \mathbf{B} reflects the specific network topology, fixed staging, cycle, saturation flows, and turning rates. To solve the system we assume demand is in its steady-state: $\Delta\mathbf{d}(k) = 0$.

Cost:

To achieve the goal of minimising traffic accumulation in all links in the network, the number of cars in any given link is penalised quadratically. This introduces an automatic trade-off between links, as reducing the number of cars in one link leads to a higher number of cars in another link. This trade-off ensures that traffic is evenly distributed and no bottlenecks are created. Furthermore, deviations from the level of steady state green times are penalized quadratically. This ensures the system returns to a sensible equilibrium steady-state when green times are changed in a given period.

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

\mathbf{Q} and \mathbf{R} in the cost equation are non-negative definite, diagonal weighting matrices. Since 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} has its diagonal elements equal to the inverses of the storage capacities of the corresponding links. \mathbf{R} influences the magnitude of the control reactions. $\mathbf{R} = r\mathbf{I}$, where the choice of r is picked to ensure the best results for a given application network (i.e. simple trial-error procedure). Both matrices can be modified in case any particular link carries increased significance. The infinite sum in the cost equation (4) suggest an infinite time horizon, which is used in order to obtain a time-invariant feedback law according to LQ optimisation theory.

Solution of the DP Problem:

To minimise the cost presented in equation (4) subject to the dynamics in (3), we follow Bertsekas (2005, Vol 2., pp.140). The problem is a discrete-time linear quadratic control problem with infinite-horizon, which is solved via the algebraic Riccati equation.

We define a symmetric positive definite **cost-to-go matrix** \mathbf{K} evolving backwards in time from $K_N = Q$ according to:

$$K_{k-1} = Q + A^T K_k A - A^T K_k B (B^T K_k B + R)^{-1} B^T K_k A \quad (5)$$

This is the discrete-time dynamic Riccati equation of this problem. The steady-state characterization of \mathbf{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 \mathbf{X} is characterized by removing the time subscripts from the dynamic equation. Resulting in the algebraic Riccati equation of this problem:

$$K = Q + A^T K A - (A^T K B) (R + B^T K B)^{-1} (B^T K A) \quad (6)$$

The solution to this problem yields a stationary policy given by the control matrix \mathbf{L} :

$$\mathbf{L} = (B^T K B + R)^{-1} B^T K A \quad (7)$$

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

$$\Delta g_k^* = - (B^T K_k B + R)^{-1} (B^T K_k A) x_{k-1} \quad (8)$$

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

$$\Delta g^* = - (B^T K B + R)^{-1} (B^T K A) x_{k-1} \quad (9)$$

And can equivalently be written as:

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

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

TUC strategy green times:

The above derived green times may not lie within the feasible range defined in section 2.2, because the LQ problem presented above does not consider control constraints. Those, however, can be imposed after the solution to (10) is computed. Hence, this calls for another optimisation problem that is solved in real-time for 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 (10).

$$\min_{G_{j,i}} \sum_{i \in F_j} (g_{j,i} - G_{j,i})^2 \quad (11)$$

subject to

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

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

3 Simulations

To understand the workings of the TUC strategy, we apply it to a simple toy network of three junctions (depicted in Figure 1).

Figure 1: Network Graph

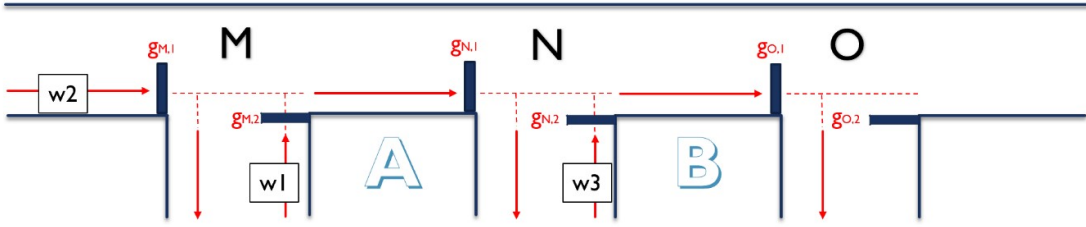
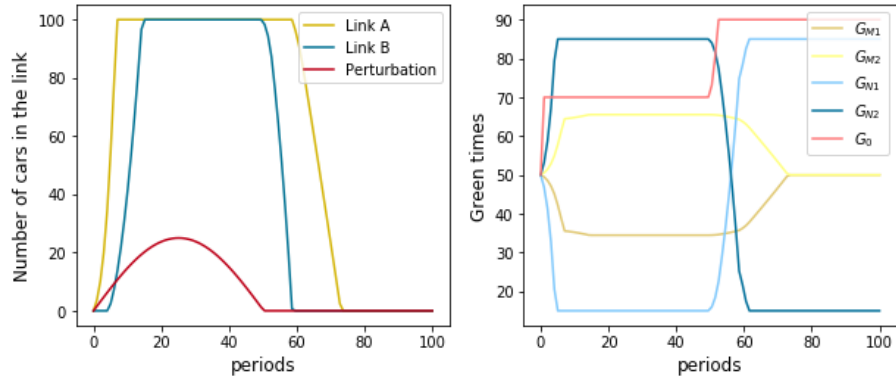


Figure 2: Network Graph

$t_{AB}=0.2$ $S_{in}=40$ $t_{out}=0.3$



4 Conclusion

References

- [1] Bertsekas, D. (2005). Dynamic Programming and Optimal Control (3rd ed.). Athena Scientific.
- [2] Diakaki, C., Papageorgiou, M. & Aboudolas, K. (2001). A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice* 10, 183-195.

Figure 3: Network Graph
 $t_{AB}=0.5$ $S_{in}=25$ $t_{out}=0.3$

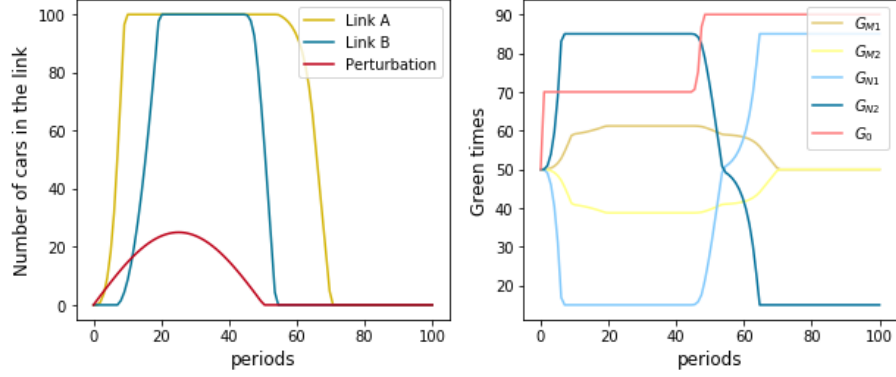


Figure 4: Network Graph
 $t_{AB}=0.5$ $S_{in}=40$ $t_{out}=0.1$

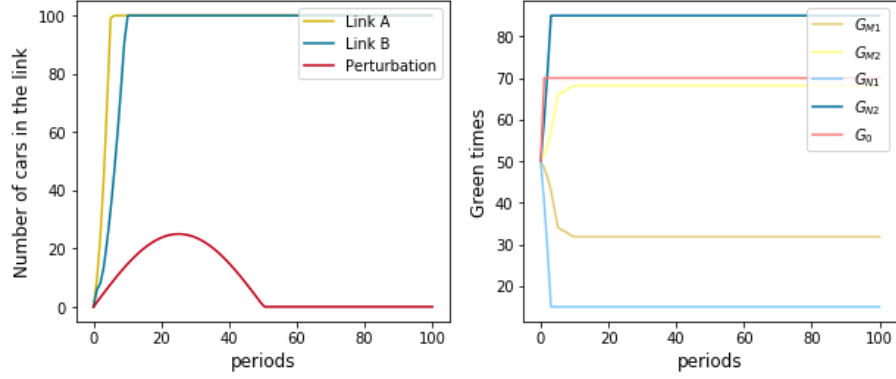
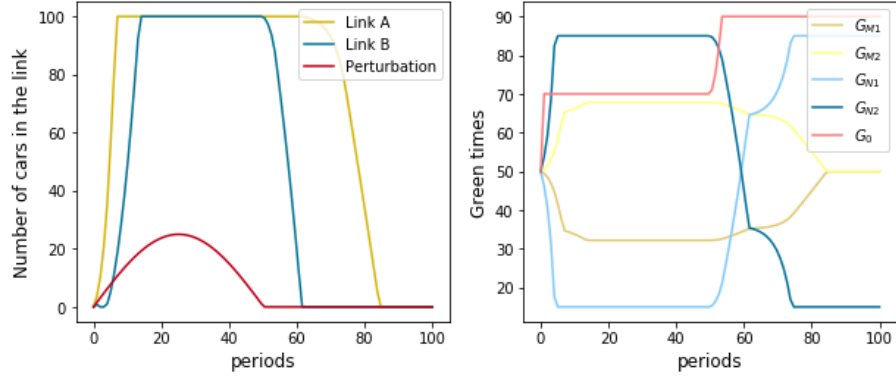


Figure 5: Network Graph
 $t_{AB}=0.5$ $S_{in}=40$ $t_{out}=0.3$



- [3] Diakaki, C., Papageorgiou, M., & McLean, T. (1997). Simulation studies of integrated corridor control in Glasgow. *Transportation Research C*, 5, 211-224.

Figure 6: Network Graph
 $t_{AB}=0.5$ $S_{in}=40$ $t_{out}=0.5$

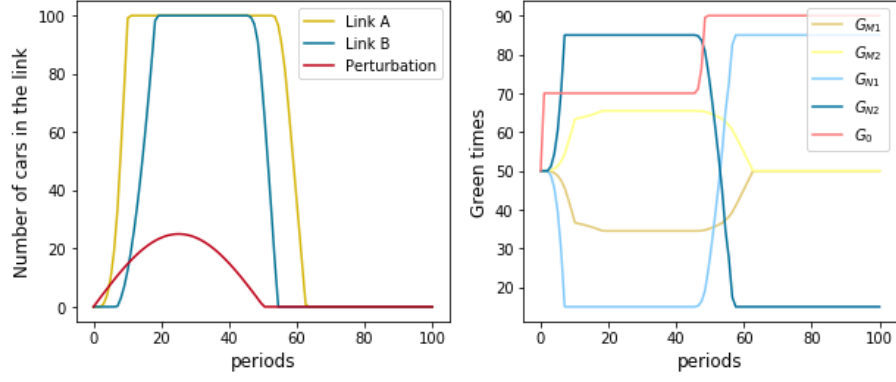


Figure 7: Network Graph
 $t_{AB}=0.5$ $S_{in}=40$ $t_{out}=0.9$

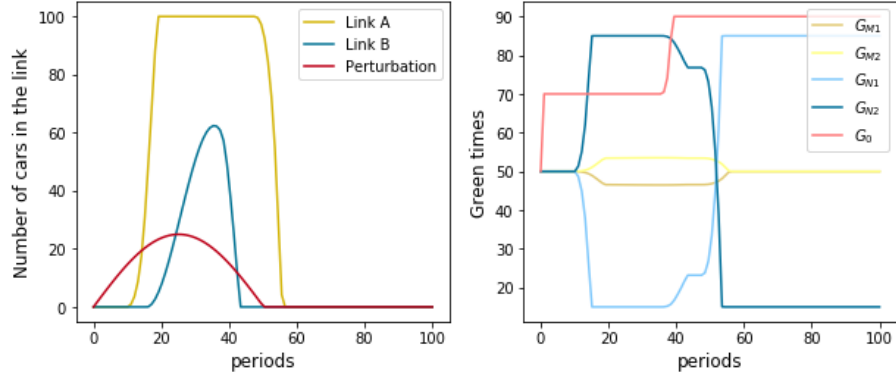


Figure 8: Network Graph
 $t_{AB}=0.5$ $S_{in}=55$ $t_{out}=0.3$

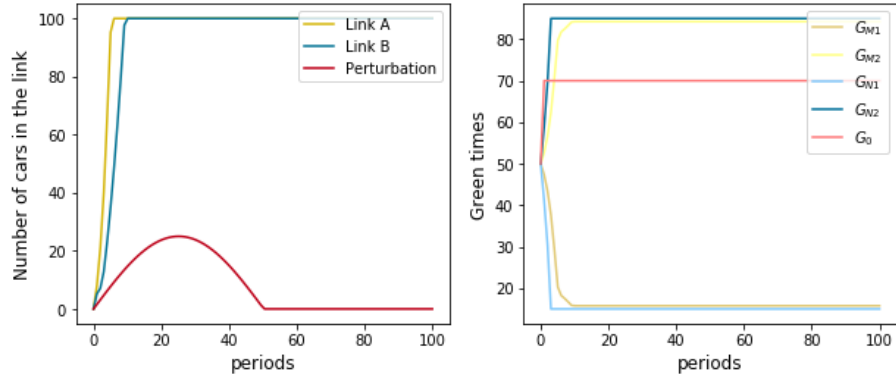


Figure 9: Network Graph
 $t_{AB}=0.8$ $S_{in}=40$ $t_{out}=0.3$

