

# Approximate Gradient Descent Convergence Dynamics for Adaptive Control on Heterogeneous Networks

Jean Carpentier,<sup>1</sup> Sebastien Blandin<sup>2</sup>

<sup>1</sup>Ecole Polytechnique, Palaiseau, France

<sup>2</sup>IBM Research, Singapore

jean.carpentier@polytechnique.edu, sblandin@sg.ibm.com

## Abstract

Adaptive control is a classical control method for complex cyber-physical systems, including transportation networks. In this work, we analyze the convergence properties of such methods on exemplar graphs, both theoretically and numerically. We first illustrate a limitation of the standard backpressure algorithm for scheduling optimization, and prove that a re-scaling of the model state can lead to an improvement in the overall system optimality by a factor of at most  $\mathcal{O}(k)$  depending on the network parameters, where  $k$  characterizes the network heterogeneity. We exhaustively describe the associated transient and steady-state regimes, and derive convergence properties within this generalized class of backpressure algorithms. Extensive simulations are conducted on both a synthetic network and on a more realistic large-scale network modeled on the Manhattan grid on which theoretical results are verified.

## Introduction

We consider the scheduling problem on queuing networks, and specifically on urban road networks. Concretely, the problem consists of the allocation of time slots to traffic lights at intersections. While the routing policy may impact the stability of the scheduling solution (Boyer, Blandin, and Wynter 2015), the routing problem is often considered decoupled (Nikolova, Brand, and Karger 2006) and is not addressed here.

The computational complexity of the scheduling problem on large-scale road networks has motivated the search for efficient decentralized algorithms only requiring local knowledge of network properties and efficient in the absence of coordination. Such decentralized approaches have proven quite efficient in practice (Smith et al. 2013; Xie, Smith, and Barlow 2012), have connections with fluid dynamic models (Brett et al. 2016), and are amenable to agent-based learning methods such as reinforcement learning (Richter, Aberdeen, and Yu 2007).

In the context of communication networks, the *backpressure* algorithm (Tassiulas and Ephremides 1992) provides a throughput-maximizing control policy, i.e. a policy that

guarantees that given any feasible flow, the maximal network queue size is asymptotically bounded. Furthermore, the backpressure policy requires only evaluation of queue size on neighboring road links.

Several properties of the backpressure algorithm make it appealing for adaptive control of dynamical road networks. First, the backpressure solution is a policy, which by definition is able to handle variability of the network state. Second it can be implemented and deployed in a fully decentralized manner since it requires only local information. Lastly, it comes with theoretical guarantees on the queue size, and has been shown to perform very well in practice.

In the context of intelligent transport systems, significant research efforts have been dedicated to extensions of the backpressure algorithm in recent years, in particular to address the specifics of traffic light scheduling on road networks (Varaiya 2009). The case of unknown routing rates was also investigated (Gregoire et al. 2014), as well as the case of queues with finite capacity (Gregoire et al. 2015), see also (Hu and Smith 2017a) and (Hu and Smith 2017b).

In-depth theoretical analysis of the backpressure properties, such as its behavior under heavy load conditions (Bramson 1999), or its link with game theory (Tenbusch et al. 2014) have also been investigated (Singh and Stolyar 2015; Moharir and Shakkottai 2013). Similar results exist for the closely related max weight algorithm (Stolyar and others 2004) and an adapted backpressure algorithm (Venkataraman et al. 2010) has been devised in this context. The backpressure algorithm was also shown to be a greedy gradient descent over the quadratic potential (Wunder and Kasparick 2012). In this context, acceleration methods have been proposed (Zargham, Ribeiro, and Jadbabaie 2013). We refer to (Moeller et al. 2010) for application to wireless sensor networks.

One of the most explicit drawbacks of the backpressure algorithm occurs at a network scale: in steady state, for certain model networks, queue sizes strictly decrease from the origin to the destination along every possible path (Bui, Srikant, and Stolyar 2009), (Ying et al. 2011), meaning that commuters traveling over longer paths incur longer queues. In this body of work, adjustments have been proposed via the consideration of an additional design cost explicitly ac-

counting for path lengths, hence attempting to compensate that drawback. Other limitations have been investigated for specific network configurations (Stolyar 2011).

In this work, we propose to analyze the convergence dynamics of the backpressure algorithm. Specifically, we are interested in a fine-grained analysis of the backpressure algorithm in different regimes, and associated convergence properties. We first explicitly re-cast the backpressure algorithm as a more general approximate gradient descent method, and show that in that class of methods, significant performance gaps exist depending on the choice of parameters. These general conclusions are derived on the case of a fundamental building block for network flow analysis, namely the  $2 \times 1$  network including two upstream links connected to one downstream link.

We then conduct an in-depth theoretical analysis of the  $2 \times 1$  network, and verify these results in simulation. We confirm experimentally that the results from the theoretical analysis obtained on a simple network apply to realistic networks such as the Manhattan grid.

The main contributions of this work include:

- illustration of arbitrarily large performance gaps in the backpressure class of adaptive control algorithms,
- theoretical identification and characterization of transient and stationary regimes of the  $2 \times 1$  network under backpressure algorithms,
- numerical validation of theoretical properties and illustration on realistic networks such as the Manhattan grid.

The rest of this article is organized as follows. We first introduce notations and formulate the problem considered. We then present our main results on the convergence dynamics on the  $2 \times 1$  network. We subsequently analyze the relative convergence of two instances of approximate gradient descent algorithms. Finally, we present detailed numerical results of the algorithm performance, and conclude.

## Preliminaries

### Network model

We consider a discrete-time network of queues with  $q_{l,m}(t) \in \mathbb{R}^+$  denoting the (continuous) number of vehicles queuing at location  $l$  at time  $t$  with the intention of traveling to the downstream link  $m$  next. In the transportation context, each  $q_{l,m}(\cdot)$  represents a distinct queue of vehicles waiting to cross an intersection with segregated movements (e.g. turn left, go straight, turn right). We also note  $q$  the vector of  $q_{l,m}$  and omit the time dependency for compactness. Queuing networks can also model public transport and multi-modal networks (Horni, Nagel, and Axhausen 2016), although the emergence of mobile data often requires hybrid approaches (Blandin et al. 2019).

The outflow  $s_{l,m}(t)$  of queue  $q_{l,m}(t)$  at time  $t$  is the maximum number of vehicles able to cross the intersection within a time slot, defined as the minimum of the queue size and the queue capacity, assumed static:

$$s_{l,m}(t) = \min(q_{l,m}(t), c_{l,m}). \quad (1)$$

When this minimum  $s_{l,m}(t)$  is reached at the queue size  $q_{l,m}(t)$ , the intersection is in *unsaturated regime*, and when the minimum is reached at the queue capacity  $c_{l,m} \in \mathbb{R}_*^+$ , the intersection is in *saturated regime*.

Given initial conditions  $q_{l,m}(t=0)$  for the queues, and prescribed source and sink flows  $e_{l,m}(t)$  specifying the number of vehicles entering and leaving the network, the conservation of vehicles reads:

$$q_{l,m}(t+1) = q_{l,m}(t) + r_{l,m}(t) \sum_k u_{k,l}(t) s_{k,l}(t) - u_{l,m}(t) s_{l,m}(t) + e_{l,m}(t) \quad (2)$$

where  $r_{l,m}(t) \in [0, 1]$  is the proportion of vehicles reaching node  $l$  intending to visit node  $m$  next, and such that  $\sum_m r_{l,m}(t) = 1$ , and  $u_{l,m}(t) \in \{0, 1\}$  is a control variable specifying whether queue  $q_{l,m}(t)$  is activated, i.e. has green light, at time  $t$ . For each intersection, the activation set follows standard constraints encoding compatible movements, (e.g. in the case of left-hand driving, turn right movements can be activated simultaneously, but not go straight and turn left movements).

Given an objective function  $V(q)$  satisfying Lyapunov properties, the *scheduling problem* is concerned with the design of an activation policy  $u_{l,m}(\cdot)$  with good properties with respect to  $V(\cdot)$ .

### Backpressure algorithm

In this section we recall some existing results. The *backpressure algorithm* (Tassiulas and Ephremides 1992) provides *maximal throughput stability* in the sense that if the inputs flows are feasible in expectation, then the queue sizes are asymptotically bounded.

**Definition 1.** *The backpressure policy is the solution  $u$  to the maximization problem :*

$$\max_u \sum_{l,m} \left( q_{l,m} - \sum_k q_{m,k} r_{m,k} \right) c_{l,m} u_{l,m}. \quad (3)$$

It can be shown, e.g. see (Wongpiromsarn 2014), that this objective function leads to activating at each decision point the traffic movement maximizing the difference between its upstream queue and its downstream queue, hence the term “backpressure”.

We first show that the back pressure policy, although arising from a local greedy formulation, corresponds to an approximate gradient descent step.

**Proposition 1.** *The backpressure algorithm (3) is an approximate gradient descent step update on the objective function  $V(q) = \frac{1}{2} \sum_{l,m} q_{l,m}^2 = \frac{1}{2} q^T q$ .*

*Proof.* If we note  $\delta(t+1) = q(t+1) - q(t)$ , the one-step temporal difference in the objective function reads:

$$V(q(t+1)) - V(q(t)) = \delta(t+1)^T q(t) + \frac{1}{2} \delta(t+1)^T \delta(t+1). \quad (4)$$

Expanding  $\delta(t+1)$  using the conservation equation (2) we can re-write  $\delta(t+1)^T q(t)$  as:

$$\delta(t+1)^T q(t) = E^T q - \sum_{l,m} \left( q_{l,m} - \sum_k q_{m,k} r_{m,k} \right) u_{l,m} s_{l,m}$$

where  $E$  is the vector of  $e_{l,m}$ , and the time-dependence is omitted on the right-hand side. In the saturated regime, the first term on the right-hand side of equation (4) dominates, and a steepest gradient descent step on the approximate temporal difference  $\delta(t+1)^T q(t)$  reads:

$$\arg \min_u \delta(t+1)^T q(t) = \arg \max_u \sum_{l,m} \left( q_{l,m} - \sum_k q_{m,k} r_{m,k} \right) u_{l,m} s_{l,m}. \quad (5)$$

Approximating the throughput (1) as  $s_{l,m} \approx c_{l,m}$  leads to the definition of the backpressure (3), which corresponds to making the approximation that the queues are in the saturated regime.  $\square$

Motivated by expression (5), in the following we define the *priority* of a queue as:

$$p_{l,m} = \left( q_{l,m} - \sum_k q_{m,k} r_{m,k} \right) c_{l,m}. \quad (6)$$

This view of backpressure as a general one step update for an approximate gradient descent in the context of adaptive control motivates us to consider a generalization of the objective function via re-scaling. Specifically, given  $\gamma_{l,m} > 0$ , we consider a generalized objective function  $V(q) = \frac{1}{2} \sum_{l,m} \gamma_{l,m} q_{l,m}^2$  associated with the generalized priorities:

$$p_{l,m} = \left( \gamma_{l,m} q_{l,m} - \sum_k \gamma_{m,k} q_{m,k} r_{m,k} \right) c_{l,m}. \quad (7)$$

We now illustrate that this re-scaling can impact the performance of the approximate gradient descent method by an arbitrary factor depending on the network heterogeneity. We focus the analysis on the comparison between two values of  $\gamma_{l,m}$ , the case of  $\gamma_{l,m} = 1$  which corresponds to the classical backpressure, and the case of  $\gamma_{l,m} = 1/c_{l,m}$  which corresponds to a variant of the backpressure algorithm where time spent in the queue is the quantity to be optimized (since  $q/c$  is the steady-state saturated regime approximation of time spent in the queue).

## Heterogeneous Flows

We now define a simple but fundamental example and show the limitations of the backpressure algorithm on that case. Consider a simple network with 2 upstream queues  $q_{1,3}$ ,  $q_{2,3}$  and 1 downstream queue  $q_{3,4}$ , where heterogeneity between the upstream queues is parameterized by a factor  $k$ . This topology corresponds to the classical *merge* junction in traffic engineering (Daganzo 1995), see (Garavella and Piccoli

2006) for the underlying mathematical theory of network fluid-dynamics model.

Given a reference capacity  $c$ , link capacities are defined as  $c_{2,3} = k c_{1,3} = kc$ . Link inflows are defined as  $f_{2,3} = k f_{1,3} = k\eta c$ . Without loss of generality, we also assume that the downstream queue is constant since the main object of this study is the competing dynamics of upstream queues given an arbitrary downstream queues. For stability, we also make the classical assumption that the inflow is lower than the uniform capacity, i.e.  $\eta_i = \frac{f_i}{c_i} \leq 0.5$ .

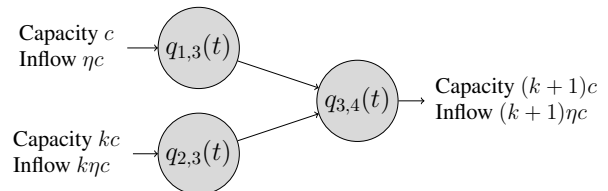


Figure 1:  $2 \times 1$  **parametric junction** with two upstream queues and one downstream queue.

We now prove that the classical backpressure algorithm (3) can perform arbitrarily poorly for heterogeneous networks. We present the results for the case  $q_{3,4}(t) = 0$ .

**Proposition 2.** *Consider the  $2 \times 1$  network from Figure 1, with  $\eta = 0.5$  and  $q_{3,4}(t) = 0$ . A basic scheduling alternating activation of each upstream queue leads to:*

$$q_{1,3}(t) \approx \eta c \quad \text{and} \quad q_{2,3}(t) \approx k\eta c \quad (8)$$

while the backpressure activation rule (3) yields:

$$q_{1,3}(t) \geq k^2 \eta c \quad \text{and} \quad q_{2,3}(t) \geq k\eta c \quad (9)$$

*Proof.* First, given that in this discrete time setting each vehicle spends at least one time step in the queue, we have  $q_{l,m} \geq f_{l,m}$ . Here, since the demand is feasible, an alternating schedule would result in  $q_{1,3}(t) \approx f_{1,3}$ , and similarly for  $q_{2,3}$ , which proves (8).

Second, on this example, the backpressure priorities from equation (6) read  $p_{l,m}(t) = c_{l,m} q_{l,m}(t)$ . Since  $q_{l,m}(t) \geq f_{l,m}$ , we have  $p_{2,3}(t) = c_{2,3} q_{2,3}(t) \geq k^2 c^2 \eta$ . The queue  $q_{1,3}$  is activated only if  $p_{1,3}(t) = c_{1,3} q_{1,3}(t) > p_{2,3}(t) \geq k^2 c^2 \eta$ , or equivalently  $q_{1,3}(t) > k^2 c \eta$ . When  $q_{1,3}$  goes below that value, queue  $q_{2,3}$  is activated because it has higher priority.

Hence the minimal queue size over time under backpressure is  $k$  times the queue size under the alternating schedule. A similar analysis can be conducted with non-zero  $q_{3,4}$ , the result is obtained for a sufficient upstream queue heterogeneity compared to the downstream queue value.  $\square$

In the rest of the article, we investigate the convergence dynamics of the approximate gradient descent dynamics under the generalized backpressure priorities (7).

## Stability Domain

We first characterize the asymptotic convergence of the parametric  $2 \times 1$  network from Figure 1. To simplify the notations on this network, we index the upstream queues by  $i \in \{1, 2\}$ .

## Regime Types

For  $\eta < 0.5$ , the demand is feasible, capacities exceed input flows, which means that at least one queue is activated more often than needed to process the input flow, which means  $s_{u,i}(t) < c_i$ . For the  $2 \times 1$  junction, either only one upstream queue is in the unsaturated regime, and we call this network regime *R1*, or both upstream queues are in the unsaturated regime, and we call this network regime *R2*.

Due to space limitation, we focus on the regime *R1* which exhibits more complex behavior (indeed under the condition  $\eta < 0.5$  regime *R2* can be proven to be a transient regime evolving into *R1* eventually) and serves as illustrative example of a limitation of the classical backpressure algorithm in the example of previous section. With a slight abuse of notations, we use the saturation state,  $s$  for saturated and  $u$  for unsaturated, to index the queue (e.g.  $(u, s) = (1, 2)$  when queue 1 is unsaturated and queue 2 is in the saturated state).

## A re-scaled backpressure algorithm

As discussed earlier, a queue is bounded below by a  $q_{i,min}$  which is the value of its input flow:

$$q_{i,min} = f_i \leq q_i(t). \quad (10)$$

The expressions of the associated minimal priority  $p_{i,min}$  can be derived by instantiating equation (7) with the priority value (10):

$$p_{i,min} = p_i(q_i = q_{i,min}) \leq p_i(t)$$

In order for a queue to be activated, its priority must be at least greater than the minimal priority of every competing queue. We call  $p_{act}$  this a-priori minimal priority to be activated:

$$p_{act} = \max_j p_{j,min}, \quad (11)$$

and the expression associated  $q_{i,act}$  follows from (7) as:

$$q_{i,act} = q_i(p_i = p_{act}) = \frac{\gamma_0}{\gamma_i} Q + \frac{1}{\gamma_i c_i} p_{act}. \quad (12)$$

In regime *R1* one queue is saturated and the other queue is unsaturated. It follows from (12) that:

$$q_{s,act} = \frac{\gamma_0}{\gamma_s} Q + \frac{\gamma_j f_j - \gamma_0 Q}{\gamma_s c_s} c_j \geq c_s \quad (13)$$

where  $j$  is such that  $p_j = p_{act}$ . If  $j = s$ , meaning that queue  $j$  is in the saturated state, then equation (13) simplifies to  $\eta \geq 1$ , which is impossible by assumption. Hence the queue  $j$  must be in the unsaturated state when reaching its activation priority, and the other queue is in the saturated state:

$$p_{act} = p_{u,min} = p_u(q_u = f_u) \quad (14)$$

We can now express sufficient conditions for each of the queues to be in saturated or unsaturated state, depending on the exogenous network parameters.

**Proposition 3.** *In the R1 regime, the following states can exist:*

- $(u, s) = (1, 2)$  when the following conditions are satisfied:

- for  $\gamma = \mathbf{1}_n$ :  $Q \geq \frac{k^2 - \eta}{k-1} c$
- for  $\gamma = [\frac{1}{c_i}]_{i \in \{1, \dots, n\}}$ :  $Q \geq \frac{k-\eta}{k-1} (k+1)c$
- $(u, s) = (2, 1)$  when the following conditions are satisfied:
  - for  $\gamma = \mathbf{1}_n$ :  $Q \leq \frac{k^2 \eta - 1}{k-1} c$
  - for  $\gamma = [\frac{1}{c_i}]_{i \in \{1, \dots, n\}}$ :  $Q \leq \frac{k\eta - 1}{k-1} (k+1)c$

*Proof.* The bounds follow from instantiating equation (13) on the cases  $(u, s) = (1, 2)$  and  $(u, s) = (2, 1)$ .  $\square$

We represent the different phases characterized by Proposition 3 in Figure 2.

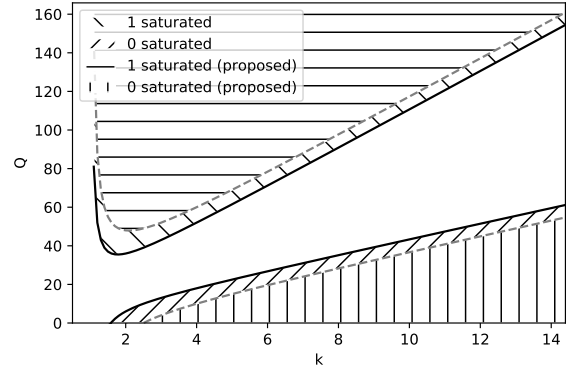


Figure 2: **Saturated and unsaturated** states for each link in *R1* regime. When  $k$  is large compared to  $Q$  (bottom right part of the chart), i.e. in case of significant heterogeneity, the smaller capacity queue (named 0) is saturated, while for larger  $Q$  values (top left part of the chart), i.e. for high network load and resulting high coupling between high-demand links, the larger capacity queue (named 1) is saturated.

We observe that with the proposed non-uniform weights  $\gamma_{l,m} = 1/c_{l,m}$ , the saturation region decreases, which suggests a better utilization of the network capacity.

## Convergence properties

In this section we characterize the transient and stationary phases of the system, and provide theoretical results on the impact of the weights  $\gamma$  on the limit of the gradient descent.

### Transient state

In regime *R1*, under the assumption that  $\eta < 0.5$ , we first prove that in general, there exists a transient regime during which the maximal priority decreases.

**Definition 2.** *The rolling min-max over 2 time slots reads:*

$$\tilde{p}_{max}(t) = \min_{v \in \{t-1, t\}} \max_{i \in \{1, 2\}} p_i(v)$$

Lemma 1 provides conditions to ensure the overall decrease in queue size during the transient phase.

**Lemma 1.** For the  $2 \times 1$  network, with  $\eta < 0.5$ , we have:

$$\forall t, \tilde{p}_{max}(t+2) \leq \tilde{p}_{max}(t). \quad (15)$$

Furthermore, with  $i^* = \arg \max p_i(t)$  and  $j$  the other queue, and with  $\Delta p(t) = p_{i^*}(t) - p_j(t) \geq 0$ , we have  $\forall t$ :

$$q_{i^*}(t) > 2f_{i^*} \text{ and } (q_j(t) > f_j \text{ or } \Delta p(t) > 0) \implies \tilde{p}_{max}(t+2) < \tilde{p}_{max}(t) \quad (16)$$

*Proof.* The technical proof can be found in the supplemental material.  $\square$

We can now use the specificity of the R1 regime to describe the dynamics of the  $2 \times 1$  network:

**Corollary 1.** If  $p_{max}(t) = p_s$ , i.e.  $i^*(t) = s$ , the saturated queue is activated for the time slot  $t$  and:

$$q_u(t) > q_{u,min} \implies \tilde{p}_{max}(t+2) < p_{max}(t)$$

and

$$\begin{cases} q_u(t) = q_{u,min} \\ \Delta p(t) = 0 \end{cases} \implies \begin{cases} i^*(t) = s \\ i^*(t+1) = i^*(t+2) = u \end{cases} \quad (17)$$

In other words  $q_s$  is activated at most once consecutively and it is followed by two activations of  $q_u$ .

*Proof.* The proof is obtained via a disjunction over the values of  $q_u(t)$ , details included in the supplemental material.  $\square$

Corollary 1 states first that the end of the transient state is related to the time when the unsaturated queue takes its minimal value. Equation (17) describes the queuing process in steady state: the saturated queue cannot be activated more than once consecutively. This implies that the unsaturated queue remains very close to its minimum value.

### Characterization of the steady state

We now generalize the result from Corollary 1.

**Proposition 4.** The steady state is reached after a finite time  $t_0$  and is characterized by:

$$\forall t \geq t_0, \tilde{p}_{max}(t) = p_{act}$$

where  $p_{act}$  is defined by (14).

*Proof.* First we prove that if queue  $s$  is regularly activated then  $\tilde{p}_{max}(t)$  strictly decreases overtime. It eventually reaches the minimal value  $p_{act}$ .

Lemma 1 states that  $\tilde{p}_{max}(t)$  is non-increasing therefore bounded above by  $P$ . The input flow within one time slot is bounded (actually equal to  $f_i$ ) so between two time slots,  $p_{max}(t)$  remains bounded above with  $p_{max}(t) \leq P + \max_i f_i$ . In the  $2 \times 1$  network,  $p_i$  is affine of  $q_i$ , so queues are also bounded above. Hence there exists an interval such that every queue is activated at least once within this interval otherwise the constants input flows would make it diverge.

Let us consider such an interval and a time  $t_s$  in that interval at which  $q_s$  is activated. Equation (15) states that  $\tilde{p}_{max}$  is non increasing. Corollary 1 states that while  $q_u(t) > q_{u,min}$ ,  $\tilde{p}_{max}$  is decreasing strictly. Let us consider the other case, for which  $q_u(t_s) = q_{u,min}$ , corresponding to  $p_u(t_s) = p_{act}$  according to (14). We have that  $p_s(t_s) = p_{max}(t_s)$ :

- if  $p_s(t_s) > p_{act}$ , equivalently  $\Delta p(t_s) > 0$  and (16) yields  $\tilde{p}_{max}(t)$  strictly decreasing,
- else,  $\Delta p(t_s) = 0$ ,  $p_{max}(t) = p_{act}$ , so  $\forall t > t_s, \tilde{p}_{max}(t) = p_{act}$  which is the lower bound.

In any case, either  $\tilde{p}_{max}(t)$  decreases or has reached its lower bound after the activation of  $q_s$ , which means that in finite time,  $\tilde{p}_{max}(t)$  is arbitrarily close to  $p_{act}$ .  $\square$

It follows that  $\tilde{p}_{max}(t)$  being constant bounds the size of the queues below and above. Experimentally, these bounds are quite tight, see Figure 3.

**Theorem 1.** In the R1 regime, given  $\eta < 0.5$ , the  $2 \times 1$  network converges to a steady state where under a generalized backpressure algorithm with  $\gamma > 0$ :

- for the unsaturated queue  $q_u$ :

$$f_u \leq q_u(t) \leq 2f_u$$

- and for the saturated queue:

$$q_{s,act} + (f_s - c_s) \leq q_s(t) \leq q_{s,act} + f_s$$

with

$$q_{s,act} = \frac{\gamma_0}{\gamma_s} \left(1 - \frac{c_u}{c_s}\right) Q + \frac{\gamma_u c_u}{\gamma_s c_s} f_u$$

*Proof.* The proof consists of expanding the results from Proposition 4.

**For the unsaturated queue** we obtain:

$$p_u(t) \leq \tilde{p}_{max}(t) + \gamma_u f_u c_u = p_{act} + \gamma_u f_u c_u \quad (18)$$

Equation (14) states that  $p_{act} = p_u(q_u = f_u)$  Consequently the queues associated with the priorities from equation (18) read:

$$q_u(t) \leq 2f_u$$

and the lower bound corresponds to (10).

**For the saturated queue:**

$$\begin{aligned} p_s(t) &\leq \tilde{p}_{max}(t) + \gamma_s f_s c_s \\ &= p_{act} + \gamma_s f_s c_s \\ &= \underbrace{(\gamma_u f_u - \gamma_0 Q)}_{p_{act} \text{ (cf. (14))}} c_u + \gamma_s f_s c_s \\ &= p_{s,max} \end{aligned}$$

We define as well the upper bound  $q_{s,max}$  of  $q_s$  s.t.  $p_{s,max} = p_s(q_s = q_{s,max}) = (\gamma_s q_{s,max} - \gamma_0 Q)$ . Consequently:

$$q_{s,max} = f_s + \frac{\gamma_0}{\gamma_s} \left(1 - \frac{c_u}{c_s}\right) Q + \frac{\gamma_u c_u}{\gamma_s c_s} f_u$$

For the minimum  $q_{s,min}$ , we note that it is reached after an activation done with  $p_s(t) = p_{act}$  (defined as the minimal priority to be activated):

$$\begin{aligned} p_{s,min} &= p_{act} + \gamma_s (f_s - c_s) c_s \\ &= p_{s,max} - \gamma_s c_s c_s \end{aligned}$$

This corresponds to associated queue values  $q_{s,min} = q_{s,max} - c_s$  and the result follows.  $\square$

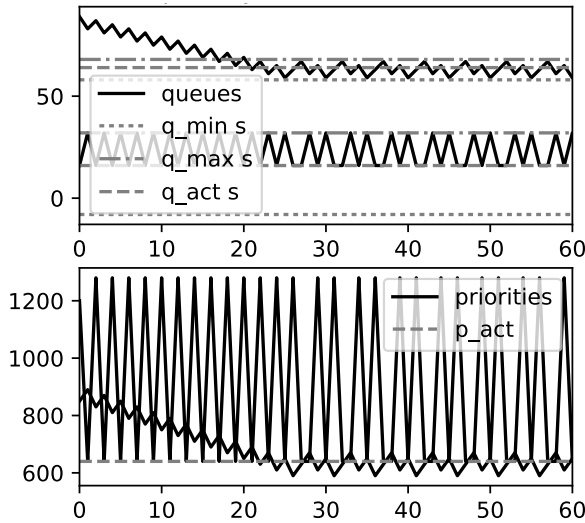


Figure 3: **Convergence to the steady state** for queues (top) and priorities (bottom). As predicted by the theory, the priority decreases (here until time step 25) and then oscillates around a constant activation priority.

We illustrate the results of this section using a numerical simulation reported in Figure 3.

After a transient state corresponding to priority values higher than  $p_{act}$ , the system stabilizes to a periodic state. Priorities oscillate around  $p_{act}$  and  $\tilde{p}_{max}(t)$  is constant, as expected from the theory.

### Total time spent in the network

We now derive the total time spent in the network, assuming that the average queue size is the average of the bounds from Theorem 1.

**Definition 3.** The average queue size of  $\bar{q}_s$  is approximated by:

$$\bar{q}_s = q_{act} + f_s - \frac{1}{2}c_s$$

The queue size translates directly into time spent given our implicit choice of a unit time step. We can now compare the classical backpressure algorithm and the proposed backpressure algorithm based on total time spent in the network.

**Theorem 2.** In regime R1, the ratio of total time spent in the network as the heterogeneity increases is:

- with  $(u, s) = (1, 2)$ :  $\frac{\bar{q}_{s, classical}}{\bar{q}_{s, proposed}} \sim_{k \rightarrow \infty} 1$
- with  $(u, s) = (2, 1)$ :  $\frac{\bar{q}_{s, classical}}{\bar{q}_{s, proposed}} \sim_{k \rightarrow \infty} k$

*Proof.* We expand the expression of total time spent from Definition 3 using the detailed values from Theorem 1.  $\square$

## Numerical Results

We first introduce the experimental setup, then go over results of benchmark experiments, and finally analyze performance on scenarios mimicking realistic conditions.

### Experimental setup

We consider a heterogeneous Manhattan grid wherein some links have high capacity (major arterial roads) and other links have low capacity (secondary arterial roads). At each junction, 3 distinct movements are authorized (left, straight, right) with the straight movement having double capacity.

The average demand for each origin node, destination node pair is drawn from an exponential distribution. For every origin-destination pair, the path minimizing the travel-time at the speed limit is computed, and flow is assigned accordingly.

Routing rates at each junction for the aggregate flow are computed from the full assignment, by computing the proportion of flow using each movement compared to the incoming flow. Input flow is drawn from a Poisson law with mean defined for each origin-destination pair as explained above.

Performance metrics such as time spent in the network are computed by summing over time the sizes of queues.

### Benchmark experiments

In this section we perform controlled experiments to investigate and validate various properties of the proposed algorithm, compared to the classical backpressure algorithm. We consider a  $10 \times 10$  Manhattan grid with a major arterial every 5 blocks, and a time step of 30 seconds.

First we analyze the impact of the parameter  $\rho$  characterizing the magnitude of the demand. Figure 4 displays the ratio of time spent in the network for 500 time steps.

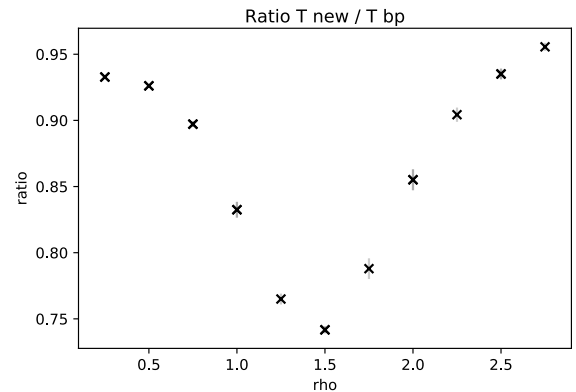


Figure 4: **Ratio of time spent in the network** for the two algorithms for increasing demand. Every point represents 300 simulations and displays the deviation of the sample for classical backpressure (“bp”) and proposed algorithm (“new”).

These results confirm the theoretical results on the existence of three regimes depending on the values taken by  $k, Q$ : for low demand (low  $\rho$  values), the flow is not really

constrained so both algorithms have similar performance, for demand around and slightly above capacity there is little available supply and the new algorithm improves on the classical backpressure, and for higher demand the network is too saturated to leave room for optimization and both algorithms have similar performance. In this setting, the network is unstable for  $\rho > 2$ . The improvement is at most 25%.

Second we analyze the impact of the parameter  $h$  characterizing the network heterogeneity, with  $1/h$  being the density of major arterial roads;  $h = 0$  corresponds to no major arterial,  $h = 1$  corresponds to all roads being major arterial, 2 to one over 2, etc. The results of simulations are presented in Figure 5.

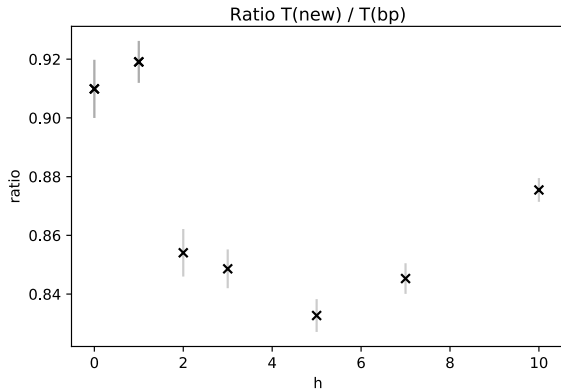


Figure 5: **Ratio of time spent in the network** for the two algorithms for different distances between parallel highways. Every point represents 300 simulations, for classical backpressure (“bp”) and proposed algorithm (“new”).

Low values of  $h$  correspond to a homogeneous network and the experiments confirm ( $h = 0, 1, 2$ ) that the proposed algorithm has no significant impact on traffic compared to the classical backpressure. For higher  $h$  the network is more heterogeneous and as expected the proposed algorithm has better relative performance.

We now analyze the impact of the ratio of the capacities between major arterial roads and secondary arterial roads. The results of the simulations are displayed in Figure 6.

For a homogeneous network (capacity ratio close to 1), the performance of both algorithms is similar. The comparative performance of the new algorithm with respect to the regular backpressure increases as the capacity ratio increases. The higher the capacity ratio the higher the flow on the major arterials (because their attractivity increases) so the more flow heterogeneity there is between major arterial and other roads.

These cases correspond to the lower part of Figure 2 for the single junction case where the proposed algorithm was proven theoretically to have better performance. In order to understand how the proposed algorithm achieves better performances we compare the average size of each queue for a heterogeneous network in Figure 7.

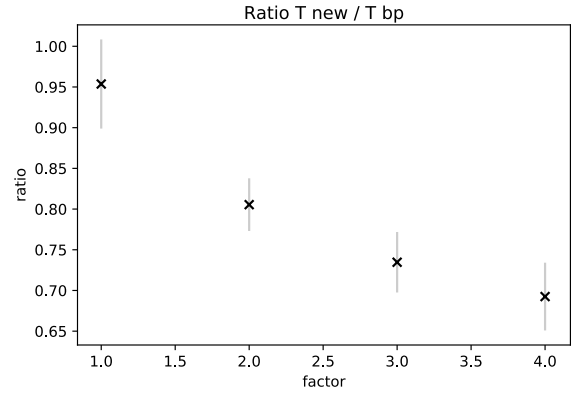


Figure 6: **Ratio of time spent in the network** for the two algorithms for increasing capacity heterogeneity. Every point represents 300 simulations, for classical backpressure (“bp”) and proposed algorithm (“new”).

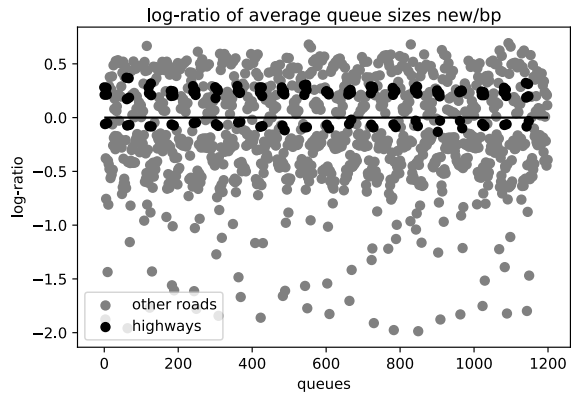


Figure 7: **Log-ratio of the average queue size** for the proposed algorithm versus the classical algorithm, for every link in the network, including major arterials (highways) and secondary arterials (other roads).

The results highlight that the proposed algorithm results in a moderate queue size increase on major arterials but yields significant queue size reduction on secondary arterial roads, meaning that the proposed algorithm is better able to take advantage of the network heterogeneity. In the next section we further explore the algorithm performance in realistic scenarios.

### Peak hour scenario

In this section, we analyze a scenario modeled after a peak hour. The specific network considered is a Manhattan network with a  $50 \times 10$  grid with a major arterial road every 4 blocks. The factor  $\rho$  varies over time in a triangular shape from 0 to 3 to model a demand temporally exceeding the network capacity ( $\rho > 2$  corresponds to an instable network).

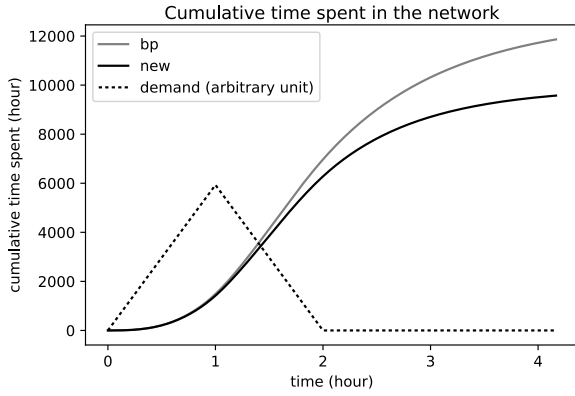


Figure 8: **Cumulative time spent in the network** for both algorithms (the unscaled value of  $\rho$  is also represented).

Figure 8 illustrates that the excess demand causes the cumulative time spent in the network to increase in all cases. However the increase is less important in the case of the proposed algorithm, in particular when the demand returns to a low value, and subsequently. This can be explained by the fact that in the first part of the peak period, the network is capacity-constrained, hence no algorithm is given sufficient freedom to optimize. However in the second phase of the peak time, and subsequently, the proposed algorithm is able to take better advantage of available capacity.

### Incident scenario

The second scenario that we consider is an incident modeled as a link with zero capacity for a one hour period in a Manhattan network. The network size and simulation parameters are chosen such that the boundary links are never impacted, and the incident happens after the network loading period, and clears before the end of the simulation.

We first analyze in Figure 9 the extent to which queues are smoothed in space via the backpressure effect of preserving already large queues from further inflow.

For both the classical backpressure and the proposed algorithm, the queues at the incident location are much lower compared to the fixed cycle policy, which in practice reduces the chances of grid-lock. This is achieved at the cost of having slightly longer queues upstream of the incident.

We now consider the cumulative queue length across the network as a function of time in Figure 10.

We observe that both the classical backpressure and the proposed algorithm outperform the fixed cycle policy, and the proposed algorithm slightly improves on the classical backpressure, with the difference between all methods increasing as the network gets more saturated, since the key benefit of the adaptive scheduling algorithms lies in efficiently using available link capacities around saturated conditions.

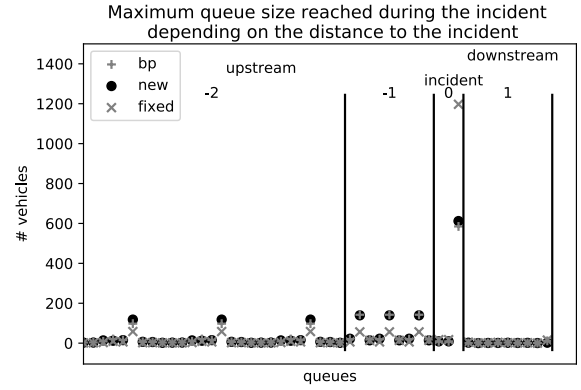


Figure 9: **Maximum queue size:** as a function of distance to incident (at the intersection directly connected to the incident link, 1 hop and 2 hops upstream, as well as 1 hop downstream) for the classical backpressure (“bp”), the proposed algorithm (“new”), and the fixed cycle policy (“fixed”).

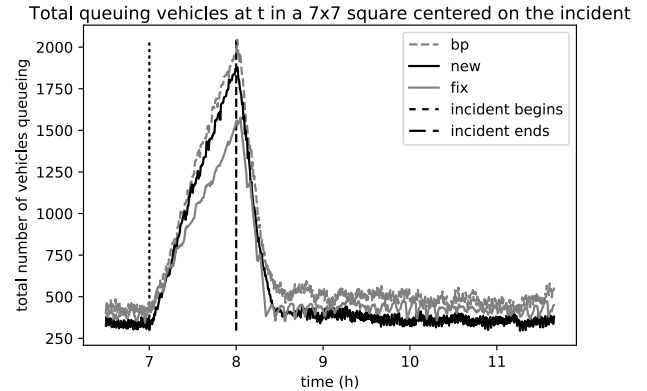


Figure 10: **Queue size in the vicinity of the incident link:** under the classical backpressure policy (“bp”), the proposed control policy (“new”), and a fixed cycle policy (“fix”) as a benchmark. The incident start and end times are indicated with vertical dashed lines.

### Conclusion

In this work we investigated the problem of the convergence properties of an approximate gradient descent method for network adaptive control.

Using a fundamental  $2 \times 1$  network, we proved that different regimes exist depending on exogenous parameters such as the magnitude of the demand with respect to the network capacity, and the heterogeneity of the flow across competing links. We characterized each of these regimes theoretically, and verified in simulation on realistic network the expected theoretical properties.

As part of this analysis, we also showed that appropriate calibration of the weights in the objective function can significantly improve the asymptotic objective value, by up to  $\mathcal{O}(k)$  with  $k$  the ratio of competing queues capacity.



## References

- Blandin, S.; Wynter, L.; Poonawala, H.; Laguna, S.; and Dura, B. 2019. FASTER: Fusion AnalyticS for public Transport Event Response. In *Proceedings of the 18<sup>th</sup> Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.
- Boyer, S.; Blandin, S.; and Wynter, L. 2015. Stability of Transportation Networks Under Adaptive Routing Policies. *Transportation Research Procedia* 7:578–597.
- Bramson, M. 1999. Stability and heavy traffic limits for queueing networks. In *Perplexing Problems in Probability*. Springer. 247–267.
- Brett, C.; Jabari, S.; Blandin, S.; and Wynter, L. 2016. In *95th Annual Meeting of the Transportation Research Board, Washington, D.C.*
- Bui, L.; Srikant, R.; and Stolyar, A. 2009. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In *INFOCOM 2009, IEEE*, 2936–2940.
- Daganzo, C. 1995. The cell transmission model, part II: Network traffic. *Transportation Research Part B: Methodological* 29(2):79–93.
- Garavello, M., and Piccoli, B. 2006. *Traffic flow on networks*. American Institute of Mathematical Sciences, Springfield, MO.
- Gregoire, J.; Frazzoli, E.; De La Fortelle, A.; and Wongpiromsarn, T. 2014. Back-pressure traffic signal control with unknown routing rates. *IFAC Proceedings Volumes* 47(3):11332–11337.
- Gregoire, J.; Qian, X.; Frazzoli, E.; De La Fortelle, A.; and Wongpiromsarn, T. 2015. Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems* 2(2):164–173.
- Horni, A.; Nagel, K.; and Axhausen, K. W. 2016. *The multi-agent transport simulation MATSim*. Ubiquity Press London.
- Hu, H.-C., and Smith, S. F. 2017a. Coping with large traffic volumes in schedule-driven traffic signal control. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling*, 154–162.
- Hu, H.-C., and Smith, S. F. 2017b. Softpressure: a schedule-driven backpressure algorithm for coping with network congestion. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4324–4330.
- Moeller, S.; Sridharan, A.; Krishnamachari, B.; and Gnawali, O. 2010. Routing without routes: The back-pressure collection protocol. In *Proceedings of the 9<sup>th</sup> ACM/IEEE International Conference on Information Processing in Sensor Networks*, 279–290. ACM.
- Moharir, S., and Shakkottai, S. 2013. Maxweight vs. back-pressure: Routing and scheduling in multi-channel relay networks. In *INFOCOM, 2013 Proceedings IEEE*, 1537–1545. IEEE.
- Nikolova, E.; Brand, M.; and Karger, D. R. 2006. Optimal route planning under uncertainty. In *Proceedings of the 16th International Conference on Automated Planning and Scheduling*, volume 6, 131–141.
- Richter, S.; Aberdeen, D.; and Yu, J. 2007. Natural actor-critic for road traffic optimisation. In *Advances in Neural Information Processing Systems (NIPS)*, 1169–1176.
- Singh, R., and Stolyar, A. 2015. Maxweight scheduling: Asymptotic behavior of unscaled queue-differentials in heavy traffic. In *ACM SIGMETRICS Performance Evaluation Review*, volume 43, 431–432.
- Smith, S. F.; Barlow, G. J.; Xie, X.-F.; and Rubinstein, Z. B. 2013. Smart urban signal networks: Initial application of the surtrac adaptive traffic signal control system. In *Proceedings of the 13th International Conference on Automated Planning and Scheduling*.
- Stolyar, A. L., et al. 2004. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability* 14(1):1–53.
- Stolyar, A. L. 2011. Large number of queues in tandem: Scaling properties under back-pressure algorithm. *Queueing Systems* 67(2):111–126.
- Tassioulas, L., and Ephremides, A. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control* 37(12):1936–1948.
- Tenbusch, S.; Löding, C.; Radmacher, F.; and Gross, J. 2014. Guaranteeing stability and delay in dynamic networks based on infinite games. In *11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 461–469.
- Varaiya, P. 2009. A universal feedback control policy for arbitrary networks of signalized intersections. *Published online. URL: <http://paleale.eecs.berkeley.edu/varaiya/papers\ps.dir/090801-Intersections.v5.pdf>*.
- Venkataramanan, V.; Lin, X.; Ying, L.; and Shakkottai, S. 2010. On scheduling for minimizing end-to-end buffer usage over multihop wireless networks. In *INFOCOM, 2010 Proceedings IEEE*, 1–9.
- Wongpiromsarn, T. 2014. Throughput Optimal Distributed Traffic Signal Control. *arXiv preprint arXiv:...* 1–23.
- Wunder, G., and Kasparick, M. 2012. Universal stability and cost optimization in controlled queueing networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 3069–3073.
- Xie, X.-F.; Smith, S. F.; and Barlow, G. J. 2012. Schedule-driven coordination for real-time traffic network control. In *Proceedings of the 12th International Conference on Automated Planning and Scheduling*.
- Ying, L.; Shakkottai, S.; Reddy, A.; and Liu, S. 2011. On combining shortest-path and back-pressure routing over multihop wireless networks. *IEEE/ACM Transactions on Networking (TON)* 19(3):841–854.
- Zargham, M.; Ribeiro, A.; and Jadbabaie, A. 2013. Accelerated backpressure algorithm. In *IEEE Global Communications Conference (GLOBECOM)*, 2269–2275.