Distributed Augmentation-Regularization for Robust Online Convex Optimization

M. Vaquero, J. Cortés

University of California, San Diego, USA
(e-mail: {mivaquerovallina,cortes}@ucsd.edu)

Abstract: This paper studies the use of distributed, primal–dual dynamics to solve continuous, time–dependent optimization problems on the fly. When using primal–dual dynamics, the availability of a strongly convex–strongly concave Lagrangian is desirable, but this is a strong assumption not satisfied in many applications. To deal with this, we develop a new Lagrangian regularization technique that seeks to minimize the perturbation to the original solutions and is compatible with the distributed nature of the optimization problem. We provide analytic bounds of the tracking error of the optimal solution using standard Lyapunov stability analysis techniques. As an application, we consider a receding horizon formulation of a dynamic traffic assignment problem and illustrate the performance of our approach in simulation.

Keywords: Primal-dual dynamics, distributed convex optimization, input-to-state stability, online algorithms, dynamic traffic assignment.

1. INTRODUCTION

The areas of application of distributed algorithms are numerous, including telecommunications, robotics, social networks, power systems, and traffic. Due to this, distributed algorithms are nowadays ubiquitous. Thus, the importance of deepening into their analysis, uncovering their convergence and stability properties, is out of doubt. We focus here on the use of online, primal–dual methods to deal with optimization problems. The basic approach is to recast the optimization problem as a saddle point problem defined by the associated Lagrangian. These saddle points can then be computed using primal–dual dynamics (also called saddle-flow dynamics). Primal–dual dynamics have also found extensions and valuable applications in time–dependent optimization problems, tracking the trajectory of optimal solutions.

In general, when the Lagrangian has nice features, for instance it is strongly convex and strongly concave, the resulting dynamics enjoys desirable characteristics, like asymptotic convergence and robustness. When such properties are not available, one might obtain them at the expense of sacrificing the accuracy of the attained solutions. This can be done using regularization procedures, like Tikhonov regularization. Here we focus on solving the mentioned time–dependent optimization problems online with rigorous guarantees of performance and stability, and developing regularization methods that have a smaller impact on the solutions of the original problem. Motivated by the relevance of traffic flow problems in the context of networked systems, here we apply our results to the Dynamic Traffic Assignment (DTA) problem. The relevant question that we address is how to optimally coordinate the traffic flow online through different actuators, including routing.

Literature Review: The framework presented here is related to recent results on time–varying, distributed, multi–user optimization (Koshal et al., 2011; Simonetto and Leus, 2014) and control of power systems (Dall'Anese and Simonetto, 2018; Dall'Anese et al., 2018). However, the differences with respect to the present treatment are meaningful. One important difference is the way our convergence and error analysis is carried out. We also differ in the regularization method, as we propose an alternative to Tikhonov. See Koshal et al. (2011) for information about Tikhonov regularization procedure.

In this work we present applications to the DTA problem using the well-known Cell Transmission Model (CTM). The origins of CTM go back to (Lighthill and Whitham, 1955; Richards, 1956) and Daganzo's discretizations (Daganzo, 1994). Since then, much effort has been devoted to its understanding and extension. One natural approach is to recast the associated control problem as a finite horizon problem using Model Predictive Control (MPC) (Gomes and Horowitz (2006); Hegyi et al. (2005)). Some of the mentioned generalizations of CTM, also using MPC, appeared in (Lovisari et al., 2014; Ba and Savla, 2016) to deal with the DTA problem as well. However, the resulting optimization problems in these papers turn out to be non–convex. The work Como et al. (2016) proposes a convex relaxation based on controlling directly the cell-to-cell flows downstream as opposed to using the turning ratios and outflow control. We follow this setting to some extent, but rapidly diverge in our algorithmic solution and analysis. A distributed solution to the DTA problem is given in Ba and Savla (2016), where the distributed algorithm design is based on ADMM which has higher computational complexity.

Statement of Contributions: The main contributions of this paper are *i)* the study of the convergence and stability properties of primal—dual dynamics associated to strongly

⋆ This work was partially supported by NSF Award CNS-1446891.

convex-strongly concave Lagrangians for time-dependent problems. We proceed directly in the continuous-time setting, which allows us to obtain the results in an elegant way using Lyapunov theory. We show that these Lagrangian functions produce dynamics which enjoy strong stability properties. These properties provide robustness against perturbations and allow us to explicitly describe the tracking capabilities of the proposed algorithms. *ii)* Since strongly convex–strongly concave Lagrangians are desirable but do not appear naturally in practice, we resort to regularization. We present a novel regularization procedure that only modifies the saddle points in the necessary directions to obtain a strongly convex-strongly concave augmented Lagrangian. We want to stress the meaningful difference with other regularizations, like Tikhonov, which acts in all directions. The idea is that the smaller the perturbation, the smaller the modification to the location of the saddle points. The associated dynamics to these regularizations can be implemented in a distributed way in some situations and, due to the way is constructed, it is very resistant against perturbations. Finally, *iii)* we apply our results to solve the DTA problem on the fly in a distributed fashion. We discuss the implementation of the algorithm and illustrate its performance in simulation. We also discuss some current research directions aimed at implementing online algorithms with feedback. This seeks to allow us to use sophisticated, high–fidelity network models in our primal–dual dynamics, going beyond the simple analytic models usually found in the literature.

2. PRIMAL–DUAL DYNAMICS

Consider the time–dependent problem

$$\begin{aligned} \min_{u} \;& f_t(u), \\ \text{s.t} \;& h_t(u) = 0, \\ & g_t(u) \leq 0, \end{aligned} \tag{1}$$

where u is a vector and the functions h_t and g_t may be vector–valuated. Vectors are supposed to be column vectors. The inequality is understood in the obvious sense. The functions are differentiable and convex, and h_t is affine ($h_t(u) = A(t)u - b(t)$). A classical approach to solve (1) is to find the saddle–points of the Lagrangian

$$\mathcal{L}^t = f_t(u) + p^T h_t(u) + y^T g_t(u). \tag{2}$$

That is,

$$\max_{p\in\mathbb{R}^{n_2}, y\in\mathbb{R}^{n_3}_+} \min_{u\in\mathbb{R}^{n_1}} \mathcal{L}^t. \tag{3}$$

To converge to these points, we use the associated primal–dual dynamics

$$\begin{aligned} \frac{du}{d\tau}(\tau) &= -\nabla_u \mathcal{L}^t(u(\tau), p(\tau), y(\tau)), \\ \frac{dp}{d\tau}(\tau) &= \nabla_p \mathcal{L}^t(u(\tau), p(\tau), y(\tau)), \\ \frac{dy}{d\tau}(\tau) &= [\nabla_y \mathcal{L}^t(u(\tau), p(\tau), y(\tau))]^+_{y(\tau)}, \end{aligned} \tag{4}$$

where $[a]^+_b = \max\{0, a\}$ if $b = 0$ and a otherwise. We extend "$[\cdot]^+$" to vectors in the evident way. There are two time–scales in this setting. We use "τ" to denote the "time" of the dynamics/algorithm and maintain "t" for the "real" time. Both variables are related by $c\tau = t$, where c is the computational time. We assume that Slater's condition holds.

2.1 Strongly Convex–Strongly Concave Lagrangians

If a Lagrangian, $\mathcal{L}^t$, happens to be strongly convex (primal) and strongly concave (dual) (which is never the case in (2), which is linear in the dual variables), as we assume from now on, then there exists a unique saddle point at any time, $(u^*(t), p^*(t), y^*(t))$. We assume that the changes with time of this saddle point are bounded, i.e., that there exists δ such that $\left\|\frac{d}{dt}(u^*(t), p^*(t), y^*(t))\right\| \leq \delta$. The following result characterizes the tracking capabilities of the continuous-time primal-dual dynamics in this case.

Theorem 1. If $\mathcal{L}^t$ is ν–strongly convex and ϵ–strongly concave, the dynamics (4) satisfies

$$\begin{aligned} &\limsup_{\tau\to\infty} \|(u(\tau), p(\tau), y(\tau)) - (u^*(t), p^*(t), y^*(t))\| \\ &\leq \frac{2c\delta}{\min\{\nu, \epsilon\}}. \end{aligned}$$

2.2 Tikhonov Regularization, Limitations and Alternatives

In applications the Lagrangian may be only convex–concave. For instance, the Lagrangians resulting from optimization problems always suffer this flaw, as they are linear in the dual variables. A well–known approach to fix this is Tikhonov regularization. For instance, the Lagrangian (2) could be regularized to

$$\begin{aligned} \hat{\mathcal{L}}^t_{\nu,\epsilon}(u, p, y) &= f_t(u) + p^T h_t(u) + y^T g_t(u) \\ &+\nu/2\,\|u\|^2 - \epsilon/2(\|p\|^2 + \|y\|^2). \end{aligned} \tag{5}$$

The reader should observe that the added terms are gathered in the second line of the expression and depend on the parameters ν and ϵ. Notice that this gives a ν–strongly convex and ϵ–strongly concave Lagrangian at the cost of modifying the saddle points in all possible directions in the space under the concern. This is because the term $\nu/2\,\|u\|^2$ acts everywhere, treating all directions in an equal way.

Here we would like to have a "smarter" regularization that acts only where it is necessary. As a means to achieve this goal, first we propose to add the "augmentation" term $\nu_1/2\,\|h_t(u)\|^2$ (remember that $h_t(u) = A(t)u - b(t)$) to the "initial" Lagrangian (2), where ν_1 is a parameter. We assume from now on that A is time–independent, this is due to technical reasons that will be explained later on. Since the term $\nu_1/2\,\|h_t(u)\|^2$ vanishes on the constraint set, it does not modify the saddle points of the original problem (3). But a simple computation shows that it does modify the hessian of the original Lagrangian (2), adding the term $A^T A$ which is a symmetric positive definite matrix. This means that the augmented Lagrangian is strongly convex in the directions u such that Au does not vanish. Now, to make the Lagrangian strongly convex in the whole space, the key observation is that we only need to complement the augmentation term in the remaining directions, that is, the ones coming from the kernel of A. Let $\{v_l\}$ be a normalized basis of the kernel of A, then we propose

$$\begin{aligned} \mathcal{L}^t_{\nu,\epsilon}(u, p, y) = f_t(u) + p^T h_t(u) + y^T g_t(u) \\ +\nu_1/2\,\|h_t(u)\|^2 + \nu_2/2\sum_l (v_l^T u)^2 - \epsilon/2(\|p\|^2 + \|y\|^2), \end{aligned} \tag{6}$$

where ν_2 is another parameter. These results may be extended to time–varying matrices, $A(t)$, under regularity assumptions, as the dimension of the kernel may change with time.

Proposition 2. The Lagrangian described in (6) is strongly convex–strongly concave for any $\nu_1,\ \nu_2,\ \epsilon > 0$, in the convex set $u \in \mathbb{R}^{n_1},\ p \in \mathbb{R}^{n_2},\ y \in \mathbb{R}^{n_3}_+$.

2.3 Robustness: Stability and Tracking Capabilities

In real–world problems perturbations of the primal–dual dynamics may happen. When the saddle–flow dynamics comes from a strongly convex and strongly concave Lagrangian the same robustness properties that allowed us to develop the tracking capabilities also make the dynamics resistant against perturbations. Let $\mathcal{L}^t$ be a ν–strongly convex and ϵ–strongly concave Lagrangian. Consider the perturbed dynamics

$$\begin{aligned}\frac{du}{d\tau}(\tau) &= -\nabla_u \mathcal{L}^t(u(\tau),p(\tau),y(\tau)) + e_1(t),\\ \frac{dp}{d\tau}(\tau) &= \nabla_p \mathcal{L}^t(u(\tau),p(\tau),y(\tau)) + e_2(t),\\ \frac{dy}{d\tau}(\tau) &= [\nabla_y \mathcal{L}^t(u(\tau),p(\tau),y(\tau)+e_3(t)]^+_{y(\tau)},\end{aligned} \tag{7}$$

where $e_i(t)$ are time–dependent, vector–valued functions.

Assumption 1. We assume

$$\|(e_1(t), e_2(t), e_3(t))\| \leq \delta_e.$$

Next, we bound the error of using the dynamics (7) to chase the saddle points of $\mathcal{L}^t$.

Proposition 3. Under the previous assumptions, let $(u(\tau),p(\tau),y(\tau))$ be an integral curve of (7) and $(u^*(t),p^*(t),y^*(t))$ the saddle point of $\mathcal{L}^t$ at time t. Then,

$$\begin{aligned}&\limsup_{\tau\to\infty} \|(u^*(t),p^*(t),y^*(t)) - (u(\tau),p(\tau),y(\tau))\|\\ &\leq \frac{2(c\delta+\delta_e)}{m}.\end{aligned}$$

3. APPLICATION: THE DTA PROBLEM

In this section we apply the obtained results to the DTA problem. The goal is to describe a distributed, online algorithm to solve the problem.

3.1 Problem Setup

Here we follow the approach in Como et al. (2016); Lovisari et al. (2014); Ba and Savla (2016). The topology of the network under concern is described by a directed graph, $\mathcal{N} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ the edge set. Given an edge i, ζ^i stands for its head and σ^i for its tails. The edges of the graph represent cells, and the vertices embody junctions (including merge, diverge or mixed type) between two consecutive cells. Two cells i and j are consecutive if $\zeta^i = \sigma^j$. One of the nodes, usually denoted w, represents the exterior world and so all the sources (on–ramps) or sinks (off-ramps) of the network are connected to it. More precisely, sources are characterized by having w as its tail, and sinks have w as its head. The network state is totally characterized by a time–dependent vector $x(t) \in \mathbb{R}^{\mathcal{E}}$, such that each component $x^i(t)$ denotes the traffic volume at time t and cell i. The sets of sources and sinks are $\mathcal{R}$ and $\mathcal{S}$ respectively. The keystones of the CTM are the law of mass conservation and the fundamental diagram. The former describes the variation of the traffic volume at each cell

$$\dot{x}^i(t) = r^i(t) - z^i(t), \tag{8}$$

where $r^i(t)$ and $z^i(t)$ refer to the traffic inflow and outflow at cell i and time t. These flows are determined by

$$r^i(t) = \lambda^i(t) + \sum_{j\in\mathcal{E}} f^{ji}(t),\ z^i(t) = \mu^i(t) + \sum_{j\in\mathcal{E}} f^{ij}(t), \tag{9}$$

where $\lambda^i(t)$ is the inflow at the sources and zero otherwise. In a similar fashion, $\mu^i(t)$ is the outflow from the network, positive only at the sinks. Meanwhile, $f^{ij}(t)$ is the traffic flow from cell i to j, which is assumed to be zero if the cells are not consecutive. We assume that

$$f^{ij}(t) = R^{ij}(t) z^i(t) \tag{10}$$

where $R^{ij}(t)$ are time–varying matrices such that $R^{ij} = 0$ if i and j are not consecutive, and such that $\sum_{j\in\mathcal{E}} R^{ij} = 1$ and $R^{ij} \geq 0$ otherwise. Since inflows and outflows are limited by the underlying infrastructure, it is necessary to introduce a demand $(d^i(x^i))$ and supply $(s^i(x^i))$ functions for each cell in the network. The maximum flow capacity is denoted by C^i. The demand functions are non–decreasing and $d^i(0) = 0$. At the non–source cells, the supply function is non–increasing and $s^i(0) >$ and $\inf\{x^i\colon s^i(x^i) < 0\}$ represents the point where the cell achieves its maximum capacity, x^{jam}. At the sources the supply functions are assumed to take the value ∞. These concepts constitute the fundamental diagram, $\min\{d^i(x^i), C^i, s^i(x^i)\}$. Mathematically, this implies

$$r^i(t) \leq s^i(x^i(t)),\ z^i(t) \leq \min\{d^i(x^i(t)), C^i\}. \tag{11}$$

Assumption 2. The demand and supply functions are concave and smooth.

We assume control over the outflow from each cell, which may be deployed by ramp metering at the sources and free-flow speed control at the non–source cells, with the equations

$$\begin{aligned}\overline{d}^i(x^i,\alpha) &= \min\{\alpha d^i(x^i), C^i\}, \quad i \in \mathcal{E}\backslash\mathcal{R},\\ \overline{d}^i(x^i,\alpha) &= \min\{d^i(x^i), \alpha C^i\}, \quad i \in \mathcal{R},\end{aligned} \tag{12}$$

where $\alpha(t) \in [0,1]$. With these controls the expressions governing the outflow from each sink cell read

$$z^i(t) = \mu^i(t) = \overline{d}(x^i(t), \alpha^i(t)), \quad i \in \mathcal{S}, \tag{13}$$

while different strategies can be used at the other cells. One of them is $z^i = \gamma\overline{d}(x^i,\alpha^i)$, where

$$\max_{\substack{k\,\in\,\mathcal{E}\\ \zeta^i = \sigma^k}} \left(\gamma \sum_{h\in\mathcal{E}} R^{hk}\overline{d}^h(x^h,\alpha^h) - s^k(x^k)\right) \leq 0. \tag{14}$$

This provides a First–in–First–Out (FIFO) policy, but other policies could be implemented, see Lovisari et al. (2014) for some related discussions.

3.2 Model Predictive Control Formulation

In MPC, in each iteration the control inputs are computed for a fixed time horizon. Next, the control policy is applied only during the first time step. This loop may be repeated indefinitely. We fix a time horizon T and, at any time t, we also assume that functions $\lambda_t^i(s)$ and $\overline{f}_t^{ij}(s)$, defined on the interval $[T, t+T]$, are provided. These functions encode the exogenous inflows at the sources and the drivers' desire to follow certain routes from the current instant to the next T seconds. We design a cost function of the form

$$\psi(x,z) + \eta \sum_{i,j\in\mathcal{E}} (f^{ij} - \overline{f}^{ij})^2,$$

where η is a parameter to be tuned depending on the weight associated to satisfy the driver's desires. ψ is supposed to be separable $\psi = \sum_{i\in\mathcal{E}} \psi^i(x^i, z^i)$ and can take different forms. Commonly encountered examples are: Total Travel Time (TTT), $\psi^i = x^i$, and Total Travel Distance (TTD), $\psi^i = -l^i z^i$, where l^i is the length of cell i.

Assumption 3. The functions ψ^i are smooth and convex.

Returning to the MPC, at any instant we face the problem

$$\begin{aligned} \min \quad & \int_t^{t+T} \psi^i(x_t^i(s), z_t^i(s)) + \eta \sum_{i,j} (f_t^{ij}(s) - \overline{f}_t^{ij}(s))^2 ds \\ \text{s.t.} \quad & x_t^i(t) = \overline{x}_t^i \text{ for } i \in \mathcal{E}, \\ & \text{Constraints (8)–(14),} \end{aligned} \tag{15}$$

where the unknowns are the functions $x_t^i(s)$, $r_t^i(s)$, $z_t^i(s)$, $\alpha_t^i(s)$, $R_t^{ij}(s)$ defined on the interval $[t, t+T]$ and $\overline{x}_t^i$ is the state of the network at the beginning of each problem. Unfortunately, these problems are non–convex due to the non–linear equality constraints. This issue has been considered in Como et al. (2016), where the idea is to directly control the outflows, f_t^{ij} at the non–sink cells and μ_t^i at the sinks. The suggested solution gets rid of the equality constraints that were causing the non–convexity and the convex relaxation reads

$$\begin{aligned} \min \quad & \int_t^{t+T} \psi^i(x_t^i, z_t^i) + \eta \sum_{i,j} (f_t^{ij} - \overline{f}_t^{ij})^2 ds \\ \text{s.t.} \quad & x_t^i(t) = \overline{x}_t,\ \dot{x}_t^i(s) = r_t^i(s) - z_t^i(s), \\ & r_t^i(s) = \lambda_t^i(s) + \sum_{j\in\mathcal{E}} f_t^{ji}(s), \\ & z_t^i(s) = \mu_t^i(s) + \sum_{j\in\mathcal{E}} f_t^{ij}(s), \\ & r_t^i(s) \le s^i(x_t^i(s)),\ z_t^i(s) \le \min\{d^i(x_t^i(s)), C^i\}, \\ & 0 \le \mu_t^i(s) \text{ if } i \in \mathcal{S},\ \mu_t^i(s) = 0 \text{ if } i \in \mathcal{E}\backslash\mathcal{S}, \\ & 0 \le f_t^{ij}(s),\ f_t^{ij}(s) = 0 \text{ if } i \text{ and } j \text{ non consec.} \end{aligned}$$

After discretizing the (t–dependent) problems (using trapezoidal quadrature rule to compute the integral and Euler's explicit rule for the dynamics) and using the regularization procedure developed above, we have the Lagrangian below, in terms of the primal variables (x, f, μ).

$$\begin{aligned} \mathcal{L}_{\nu,\epsilon}^t = & \sum_{i\in\mathcal{E}} \Big(\frac{h}{2} \big(\psi^i(x_t^i(0), \sum_{j\in\mathcal{E}} f_t^{ij}(0) + \mu_t^i(0)) \\ & + \eta \sum_{j\in\mathcal{E}} (f_t^{ij}(0) - \overline{f}_t^{ij}(0))^2 \big) \\ & + \sum_{k=1}^{N-1} h \big(\psi^i(x_t^i(k), \sum_{j\in\mathcal{E}} f_t^{ij}(k) + \mu_t^i(k)) \\ & + \eta \sum_{j\in\mathcal{E}} (f_t^{ij}(k) - \overline{f}_t^{ij}(k))^2 \big) \\ & + h/2 \big(\psi^i(x_t^i(N), \sum_{j\in\mathcal{E}} f_t^{ij}(N) + \mu_t^i(N)) \\ & + \eta \sum_{j\in\mathcal{E}} (f_t^{ij}(N) - \overline{f}_t^{ij}(N))^2 \big) \Big) \\ & + \sum_{k=0}^{N-1} \sum_{i\in\mathcal{E}} p_{1,t}^i(k) \big(x_t^i(k+1) - x^i(k) \\ & - h(\lambda_t^i(k) + \sum_{j\in\mathcal{E}} f_t^{ji}(k) - \mu_t^i(k) - \sum_{j\in\mathcal{E}} f_t^{ij}(k)) \big) \\ & - \sum_{k=0}^{N-1} \sum_{i\in\mathcal{S}} y_{1,t}^i(k) \mu_t^i(k) - \sum_{k=0}^{N} \sum_{i,j\in\mathcal{E}} y_{2,t}^{ij}(k) f_t^{ij}(k) \\ & + \sum_{k=0}^{N} \sum_{i\in\mathcal{E}} y_{3,t}^i(k) (\sum_{j\in\mathcal{E}} f_t^{ji}(k) - s^i(x_t^i(k))) \\ & + \sum_{k=0}^{N} \sum_{i\in\mathcal{E}} y_{4,t}^i(k) (\sum_{j\in\mathcal{E}} f_t^{ij}(k) - d^i(x_t^i(k))) \\ & + \sum_{k=0}^{N} \sum_{i\in\mathcal{E}} y_{5,t}^i(k) (\sum_{j\in\mathcal{E}} f_t^{ij}(k) - C^i) \\ & + \sum_{k=0}^{N-1} \sum_{i\in\mathcal{S}} y_{6,t}^i(k) (\mu_t^i(k) - d^i(x_t^i(k))) \\ & + \sum_{k=0}^{N-1} \sum_{i\in\mathcal{S}} y_{7,t}^i(k) (\mu_t^i(k) - C^i) - \epsilon/2 \left\| (p,y) \right\|^2 \\ & + \nu_1/2 \left\| A \cdot (x,f,\mu)^T - b \right\|^2 + \nu_2/2 \sum_l (v_l^T (x,f,\mu)^T)^2. \end{aligned} \tag{16}$$

We are making some abuse of notation as we are referring to a function and its discrete version without distinction. There is no room for confusion as we are only using the discrete functions from now on. One should understand that $x_t^i(k)$ in the discrete setting corresponds to the value $x_t^i(t+kh)$ in the continuous framework. We also assume that only flows f_t^{ij} of consecutive cells are taken into account. It is easy to see that everything but the addend $\nu_2/2 \sum_l (v_l^T (x,f,\mu)^T)^2$ can be implemented in a distributed way.

Proposition 4. The basis of the kernel, $\{v_l\}$, can be chosen such that $\nu_2/2 \sum (v_l^T (x,f,\mu)^T)^2$ and its partial derivatives can be computed in a distributed way.

3.3 Data–Driven/Feedback Implementation

Recent work, cf. Dall'Anese and Simonetto (2018) and Dall'Anese et al. (2018), takes advantage of measured data to implement the primal–dual algorithm. The approach taken in the mentioned references has two main advantages, *i)* it allows to directly obtain the value of certain magnitudes that cannot be computed in a distributed way (for instance, the voltage in Dall'Anese and Simonetto (2018)); and *ii)* the feedback is able to cope with model mismatches, so the resulting algorithms are less dependent on synthetic models. Related ideas are carried out in Gan and Low (2016); Tang et al. (2017), where the feedback is used to solve the implicit power flow equations. The main idea is that lack of knowledge of certain functions or dynamics can be overcome with data.

In our context, the idea is to replace analytic models of traffic flow with information obtained by other means (e.g., high–fidelity simulations, measurements). More precisely, these means would provide us with a mapping $\mathcal{F}$ which expresses the state variables (x^i) as a function of the control variables (f^{jk}, μ^l). From now on we drop the indices for the sake of simplicity. Thus,

$$x = \mathcal{F}(f, \mu). \tag{17}$$

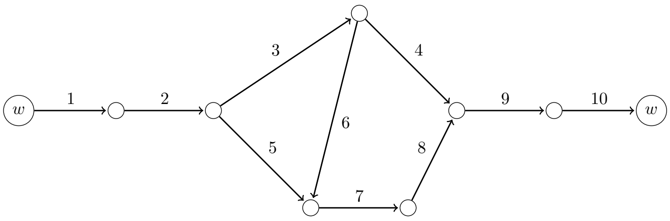
Then, we can substitute (17) into the Lagrangian (16) (we only remove the equality constraints associated to the dynamics from the Lagrangian, as we are replacing them with the dynamics given by $\mathcal{F}$) and obtain $\overline{\mathcal{L}}^t_{\nu,\epsilon}(f,\mu,p,y) = \mathcal{L}^t_{\nu,\epsilon}(\mathcal{F}(f,\mu),f,\mu,p,y)$. We compute the corresponding primal–dual dynamics for the variables f,

$$\begin{aligned}\dot f &= -\frac{\partial \overline{\mathcal{L}}^t_{\nu,\epsilon}}{\partial f} = -\frac{\partial \mathcal{L}^t_{\nu,\epsilon}(\mathcal{F}(f,\mu),f,\mu,p,y)}{\partial f}\\ &= -\frac{\partial \mathcal{L}^t_{\nu,\epsilon}}{\partial x}(\mathcal{F}(f,\mu),f,\mu,p,y)\frac{\partial \mathcal{F}}{\partial f}(f,\mu)\\ &\quad - \frac{\partial \mathcal{L}^t_{\nu,\epsilon}}{\partial f}(\mathcal{F}(f,\mu),f,\mu,p,y).\end{aligned}$$

Analogous expressions hold for the remaining variables and are even simpler for the dual variables, where only the value of the function $\mathcal{F}$ needs to be know at any iteration. Observe that the partial derivatives of the function $\mathcal{F}$ are also necessary in order to compute the primal–dual dynamics. These partial derivatives are obtained through the same means we computed $\mathcal{F}$, and so they are also considered as feedback. When discretized, this approach has some nice benefits: *i)* the reduction of the number of variables and especially *ii)* we may not know the analytic expression of $\mathcal{F}$ which may be very complicated, but only need its value at certain points and an approximation of its derivatives. This is reminiscent of the previous references in power flow systems where some of the functions are not known, but they are supplied with data and first-order information. The potential drawback of this approach is that, since the function $\mathcal{F}$ is unknown, the original optimization problem might actually be non-convex. The search for these convergence results is the subject of our current research effort.

3.4 Numerical Simulations

We implemented the results of this paper in the network below (cells are numbered), where all the cells are assumed to have equal length and equal supply and demand function ($s^i(x) = 20 - x$, $d^i(x) = x$, $C^i = 20$), $\eta = 0$, time horizon $=$ 14. Initially (time $=$ 0) the network is empty. We want to illustrate two aspects of the algorithm proposed here: stability and MPC solution tracking performance.



Stability Against Perturbations: We want to show here the nice stability properties of the proposed regularization. In order to do that we assume that at times 1, 2 and 3, 10 cars enter the network. Now we would like to show the performance of the primal–dual approach to compute one iteration of MPC in this scenario. In Figure 1 we show the behavior of the dynamics described in this paper for the first iteration of MPC in the setting described before, we compare the regularization proposed here ($\nu_1 = 50$, $\nu_2 = \epsilon = 0.01$) with Tikhonov. After 20000 iterations a constant perturbation is produced, which stops after 25000 iterations. It is observed that the proposed regularization adapts better to the perturbation.

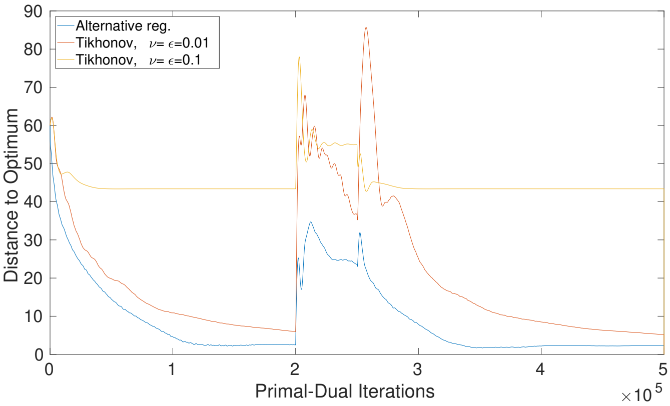


Fig. 1. Comparison: Tikhonov – Proposed regularization. Approximating a solution of MPC

MPC Performance: Here we show the performance of the algorithm tracking the optimal solution of the MPC control. Consider two different groups of 5 cars enter the network at times 1 (in red) and 3 (in blue). The supply, demand and cell capacity are the same as before, but we assume that the capacity in cell 4 drops to zero at time 5 due to an accident. Notice that in each of the MPC iterations the optimum changes and it has to be recomputed: the primal–dual dynamics chases these optimums permanently. In Figure 2, we show the corresponding plots of the tracking for the 6 first iterations of MPC. The plot shows the distance from the current iteration of primal–dual to the solution of the corresponding MPC iteration. We run 7500 iterations of primal–dual before changing the MPC iteration. The "jumps" in the distance observed in the figure below come from the change of optimum, not by discontinuities in the primal–dual dynamics. Tables 1 and 2 show the computed approximation of iterations 1 and 5 of MPC by the primal–dual dynamics. Figures 3 and 4 interpret this outcome. We maintain the same colors for the two groups, where the colored numbers represent the times the two groups reach the associated cell and we use bent arrows to avoid superposition. At the beginning, the algorithm provides the shortest path for both groups of cars, cf. Figure 3. The plan is followed until iteration 5. At iteration 5, due to an accident, the algorithm deviates the second group of cars, cf. Figure 4.

Table 1. Primal–dual approximation to MPC, policy computed in iter.= 1. The table shows the number of cars in the cells at any time.

Time	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	5.03	-0.06	0.15	0.07	0.25	-0.05	-0.30	0.14	-0.22	0.01
2	-0.03	4.92	-0.04	0.15	0.04	0.04	0.17	-0.25	0.10	-0.19
3	4.92	-0.02	4.96	0.13	-0.08	0.24	-0.05	-0.21	-0.18	0.26
4	0.03	4.84	-0.03	4.94	-0.02	0.05	0.05	0.04	-0.12	0.09
5	-0.01	-0.09	4.93	0.01	-0.05	-0.04	0.03	0.23	5.08	-0.01
6	-0.06	-0.05	0.03	4.93	0.05	0.09	0.01	0.12	-0.02	5.06
7	-0.01	-0.01	0.05	-0.01	0.04	0.08	0.10	0.06	4.87	0.02
8	0.02	0.02	0.09	0.07	-0.04	-0.03	0.15	0.05	-0.08	4.88
9	-0.08	0.09	0.15	0.14	-0.02	0.03	0.01	0.12	-0.07	0.04
10	-0.04	0.03	0.08	0.12	-0.02	0.01	0.01	0.12	-0.00	-0.08
11	0.05	0.00	0.00	-0.03	0.02	-0.02	-0.09	0.10	0.02	0.05
12	0.11	-0.07	-0.07	0.06	-0.02	0.07	-0.19	0.02	-0.19	0.11
13	-0.01	0.18	-0.00	-0.06	-0.20	-0.01	-0.26	-0.14	-0.04	-0.08
14	-0.12	-0.13	-0.04	-0.07	0.03	-0.18	-0.27	0.02	-0.20	0.22

Table 2. Primal–dual approximation to MPC, policy computed in iteration 5.

Time	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
0	-0.02	0.11	4.91	-0.05	0.01	0.01	0.04	-0.02	5.14	0.05
1	0.02	0.08	0.07	0.06	0.17	4.83	-0.00	-0.01	-0.21	5.04
2	0.03	0.07	-0.08	-0.12	0.06	-0.05	4.98	-0.03	-0.15	0.03
3	0.19	0.07	-0.26	-0.01	-0.15	0.01	-0.10	4.89	0.04	0.03
4	-0.16	-0.21	-0.09	0.04	0.04	-0.04	0.05	-0.00	5.15	0.11
5	-0.03	-0.02	0.11	0.13	0.03	0.01	-0.08	-0.05	-0.08	4.88
6	-0.04	0.04	0.12	0.01	-0.12	-0.04	0.03	-0.02	-0.13	0.04
7	-0.07	0.08	0.03	-0.17	-0.10	-0.09	-0.10	0.01	0.15	0.16
8	0.01	0.10	-0.02	0.02	-0.01	-0.12	-0.03	0.08	0.06	0.03
9	-0.06	0.03	0.00	0.02	0.02	-0.08	0.20	0.07	-0.09	0.01
10	0.07	0.01	-0.01	-0.13	0.01	0.00	0.02	0.07	0.03	0.01
11	-0.03	0.03	0.15	0.07	-0.06	-0.05	0.14	0.12	-0.20	-0.15
12	0.04	0.10	-0.01	-0.20	0.05	-0.04	-0.02	-0.04	-0.09	0.06
13	0.09	-0.07	0.00	-0.04	-0.07	-0.05	0.05	0.03	-0.08	0.01
14	0.04	-0.01	-0.01	-0.15	0.06	-0.08	0.06	0.02	0.03	-0.00

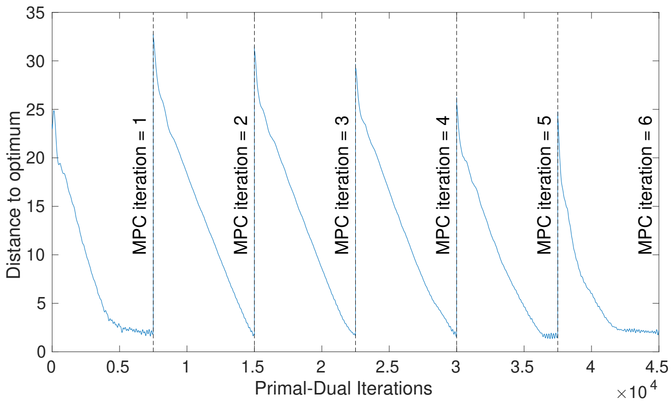


Fig. 2. Primal–dual tracking MPC optimums.

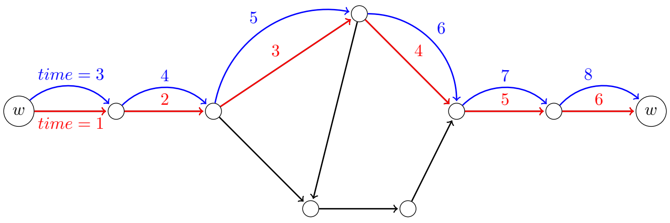


Fig. 3. Optimal evolution computed by MPC, iter.= 1. The first group of cars is in red and the second one in blue. Numbers represent the arrival time of the corresponding group.

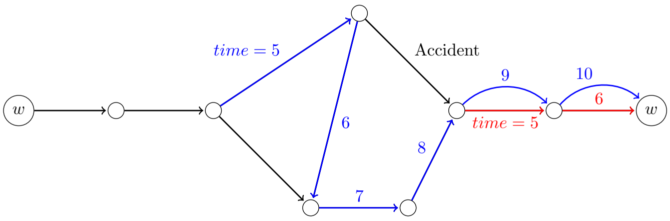


Fig. 4. Optimal evolution computed by MPC, iter.= 5.

4. CONCLUSIONS

We have presented an approach for the regularization of time-dependent convex optimization problems that builds on augmentation techniques to minimize the perturbation to the location of the saddle points. This technique respects the distributed character of the original problem and results in strongly convex-strongly concave Lagrangians, for which primal-dual dynamics exhibits robust tracking performance. Future work will explore the analytic characterization of the perturbation of the saddle points and the combination of the proposed approach with data-driven, feedback techniques to deal with increasingly realistic scenarios.

REFERENCES

Ba, Q. and Savla, K. (2016). On distributed computation of optimal control of traffic flow over networks. *4th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 5, 1102–1109.

Como, G., Lovisari, E., and Savla, K. (2016). Convexity and robustness of dynamic traffic assignment and freeway network. *Transportation Research Part B: Methodological*, 91, 446–465.

Daganzo, C. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28, 269–287.

Dall'Anese, E., Guggilam, S.S., Simonetto, A., Chen, Y.C., and Dhople, S.V. (2018). Optimal regulation of virtual power plants. *IEEE Transactions on Power Systems*, 33(2), 1868–1881.

Dall'Anese, E. and Simonetto, A. (2018). Optimal power flow pursuit. *IEEE Transactions on Smart Grid*, 9, 942–952.

Gan, L. and Low, S.H. (2016). An online gradient algorithm for optimal power flow on radial networks. *IEEE Journal on Selected Areas in Communications*, 34(3), 625–638.

Gomes, G. and Horowitz, R. (2006). Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4), 244 – 262.

Hegyi, A., Schutter, B.D., and Hellendoorn, H. (2005). Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, 13(3), 185 – 209.

Koshal, J., Nedić, A., and Shanbhag, U. (2011). Multiuser optimization: Distributed algorithms and error analysis. *SIAM J. Optim.*, 21, 1046–1081.

Lighthill, M. and Whitham, G. (1955). On kinematic waves ii. a theory of traffic flow on long crowded roads. *Phil. Trans. R. Soc. A*, 229, 317–345.

Lovisari, E., G. Como, A.R., and Savla, K. (2014). Stability analysis and control synthesis for dynamical transportation networks. *arxiv/1410.5956*.

Richards, P. (1956). Shock waves on the highway. *Operations Research*, 4, 42–51.

Simonetto, A. and Leus, G. (2014). Double smoothing for time-varying distributed multiuser optimization. *IEEE Global Conference on Signal and Information Processing*, 852–856.

Tang, Y., Dvijotham, K., and Low, S. (2017). Real-time optimal power flow. *IEEE Transactions on Smart Grid*, 8(6), 2963–2973.