Distributed Solvers for Online Data-Driven Network Optimization

Jorge Cortés

Mechanical and Aerospace Engineering
University of California, San Diego
http://carmenere.ucsd.edu/jorge

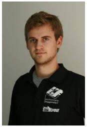
NREL Workshop:
Innovative Optimization and Control Methods
for Highly Distributed Autonomous Systems

Table Mountain Inn, Golden, CO
April 11-12, 2019

Acknowledgments


Ashish Cherukuri


Priyank Srivastava


Simon Niederlander


Erfan Nozari


David Mateos


Dean Richert

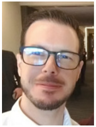

Michael McCreesh


Pavan Tallapragada


Bahman Gharesifard

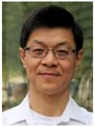

Solmaz Kia


Chin-Yao Chang


Miguel Vaquero


Sonia Martinez


Enrique Mallada


Steven Low

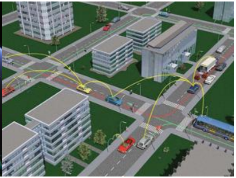

Marcello Colombino


Emiliano Dall'Anese

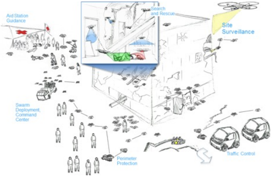
Network Optimization is Pervasive

Optimizing agent operation with limited network resources

Grid of the future



Intelligent transportation

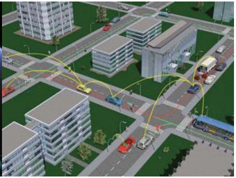


Disaster response

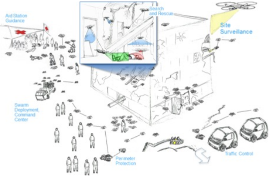
Network Optimization is Pervasive

Optimizing agent operation with limited network resources

Grid of the future



Intelligent transportation



Disaster response

Uncertainty pervasive too: interconnected world with

- unstructured environments
- complex, changing network dynamics
- humans in the loop
- contested scenarios, adversaries

Basic Taxonomy for Network Optimization

Simple, yet remarkably **ubiquitous** formulation in multi-agent applications

$$\begin{aligned} \text{minimize} \quad & \text{`aggregate objective function'}(x) \\ \text{subject to} \quad & \text{`ineq constraints'}(x) \\ & \text{`eq constraints'}(x) \end{aligned}$$

Large-scale systems

- **coupling** might come from **objective**, **constraints**, or both
- individual agents may seek to find **global** solution or only own **component**
- **coupling** topology versus **network** topology: varying degree of sparsity all the way to non-sparse at all

How do we solve optimization in a distributed way?

Distributed Solvers for Network Optimization

Network optimization w/ distributed structure (widespread in multi-agent scenarios)

$$\begin{aligned} \text{minimize} \quad & f(x) = \sum_{i=1}^{n} f_i(x_i) && \text{(separable objective)} \\ \text{subject to} \quad & g(x) \leq 0 \\ & Ax = b && \text{(locally expressible)} \end{aligned}$$

Distributed Solvers for Network Optimization

Network optimization w/ distributed structure (widespread in multi-agent scenarios)

$$
\begin{aligned}
\text{minimize} \quad & f(x) = \sum_{i=1}^{n} f_i(x_i) \quad && \text{(separable objective)} \\
\text{subject to} \quad & g(x) \leq 0 && \\
& Ax = b && \text{(locally expressible)}
\end{aligned}
$$

Optimizers of convex problems $\Leftrightarrow$ saddle points of convex-concave Lagrangian

$$L(x, y, z) = f(x) + y^\top g(x) + z^\top (Ax - b)$$

Distributed Solvers for Network Optimization

Network optimization w/ distributed structure (widespread in multi-agent scenarios)

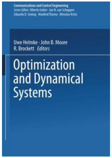
$$\begin{aligned} \text{minimize} \quad & f(x) = \sum_{i=1}^{n} f_i(x_i) && \text{(separable objective)} \\ \text{subject to} \quad & g(x) \leq 0 && \\ & Ax = b && \text{(locally expressible)} \end{aligned}$$

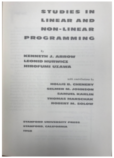
Optimizers of convex problems $\Leftrightarrow$ saddle points of convex-concave Lagrangian

$$L(x, y, z) = f(x) + y^\top g(x) + z^\top (Ax - b)$$

Dynamical systems approach to algorithms

dynamical systems that solve problems
in linear algebra, systems, optimization





Distributed Algorithm Design via Primal-Dual Dynamics

Saddle points of L can be found via saddle-point dynamics

$$\begin{aligned} \dot{x} &= -\nabla_x L(x, y, z) \\ \dot{y} &= [\nabla_y L(x, y, z)]_y^+ \\ \dot{z} &= \nabla_z L(x, y, z) \end{aligned}$$

Distributed Algorithm Design via Primal-Dual Dynamics

Saddle points of L can be found via saddle-point dynamics

$$\begin{aligned}\dot{x} &= -\nabla f(x) - y^T \nabla g(x) - A^T z \\ \dot{y} &= [g(x)]_y^+ \\ \dot{z} &= Ax - b\end{aligned}$$

Distributed Algorithm Design via Primal-Dual Dynamics

Saddle points of L can be found via saddle-point dynamics

$$\dot{x}_i = -\frac{\partial f_i}{\partial x_i}(x_i) - \sum_{\alpha} y_\alpha \frac{\partial g_\alpha}{\partial x_i}(x_i, x_{\mathcal{N}_i}) - \sum_{\beta} z_\beta A_{\beta i}$$

$$\dot{y}_\alpha = [g_\alpha(x)]^+_{y_\alpha}$$

$$\dot{z}_\beta = \sum_j A_{\beta j} x_j - b_\beta$$

From **agent** viewpoint, problem structure gives rise to distributed dynamics

Distributed Algorithm Design via Primal-Dual Dynamics

Saddle points of L can be found via saddle-point dynamics

$$\dot{x}_i = -\frac{\partial f_i}{\partial x_i}(x_i) - \sum_{\alpha} y_\alpha \frac{\partial g_\alpha}{\partial x_i}(x_i, x_{\mathcal{N}_i}) - \sum_{\beta} z_\beta A_{\beta i}$$

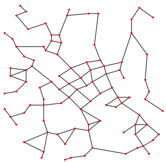
$$\dot{y}_\alpha = [g_\alpha(x)]^+_{y_\alpha}$$

$$\dot{z}_\beta = \sum_j A_{\beta j} x_j - b_\beta$$

From **agent** viewpoint, problem structure gives rise to distributed dynamics

Rich dynamical behavior

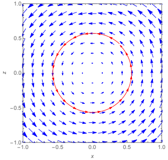
- characterization of stability and convergence properties
- appealing for large-scale systems: easily implementable by individual agents
- higher-order methods difficult to "distribute", errors in comm&sensing lead to errors in higher-order terms



Stability of Primal-Dual Dynamics

Saddle points not necessarily asymptotically stable

Primal-dual dynamics for convex-concave $F(x, z) = xz$ [Samuelson, 58]



$$\dot{x} = -\nabla_x F(x, z) = -z$$
$$\dot{z} = \nabla_z F(x, z) \quad = x$$

Saddle point $(0, 0)$ is stable, not asymptotically stable

Stream of results to understand **asymptotic convergence & properties**

- K. Arrow, L Hurwitz, and H. Uzawa. *Studies in Linear and Non-Linear Programming.* Stanford University Press, Stanford, California, 1958
- D. Feijer and F. Paganini. Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica*, 46:1974–1981, 2010
- J. Wang and N. Elia. Control approach to distributed optimization. In *Allerton Conf. on Communications, Control and Computing*, pages 557–561, Monticello, IL, October 2010
- J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *IEEE Conf. on Decision and Control*, pages 3800–3805, Orlando, Florida, 2011
- J. Chen and V. K. N. Lau. Convergence analysis of saddle point problems in time varying wireless systems - control theoretical approach. *IEEE Transactions on Signal Processing*, 60(1):443–452, 2012
- E. Mallada, C. Zhao, and S. H. Low. Optimal load-side control for frequency regulation in smart grids. *IEEE Transactions on Automatic Control*, 62(12):6294–6309, 2017
- T. Holding and I. Lestas. Stability and instability in saddle point dynamics - Part I. 2017. https://arxiv.org/abs/1707.07349
- T. Holding and I. Lestas. Stability and instability in saddle point dynamics - Part II: The subgradient method. 2017. https://arxiv.org/abs/1707.07351
- A. Cherukuri, E. Mallada, and J. Cortés. Asymptotic convergence of constrained primal-dual dynamics.

LaSalle Functions for Asymptotic Convergence

Positive-definite functions with negative semi-definite derivative

- **distance** to saddle point (x_*, y_*, z_*)

$$V_d(x, y, z) = \frac{1}{2}\left(\|x - x_*\|^2 + \|y - y_*\|^2 + \|z - z_*\|^2\right)$$

- **magnitude** of vector field

$$V_m(x, y, z) = \frac{1}{2}\Big(\|\nabla_x F(x, y, z)\|^2 + \|\nabla_z F(x, y, z)\|^2 + \sum_{j \text{ active}} \left((\nabla_y F(x, y, z))_j\right)^2\Big)$$

LaSalle Functions for Asymptotic Convergence

Positive-definite functions with negative semi-definite derivative

- **distance** to saddle point (x_*, y_*, z_*)

$$V_d(x,y,z) = \tfrac{1}{2}\left(\|x - x_*\|^2 + \|y - y_*\|^2 + \|z - z_*\|^2\right)$$

- **magnitude** of vector field

$$V_m(x,y,z) = \tfrac{1}{2}\Big(\|\nabla_x F(x,y,z)\|^2 + \|\nabla_z F(x,y,z)\|^2 + \sum_{j \text{ active}} \big((\nabla_y F(x,y,z))_j\big)^2\Big)$$

Global convergence under

- convexity-concavity plus local strong convexity-concavity
- convexity-linearity plus property of sets of saddle-points
- quasiconvexity-quasiconcavity plus property of sets of saddle-points
- local second-order information about saddle function

LaSalle Functions for Asymptotic Convergence

Positive-definite functions with negative semi-definite derivative

- **distance** to saddle point (x_*, y_*, z_*)

$$V_d(x,y,z) = \tfrac{1}{2}\left(\|x - x_*\|^2 + \|y - y_*\|^2 + \|z - z_*\|^2\right)$$

- **magnitude** of vector field

$$V_m(x,y,z) = \tfrac{1}{2}\Big(\|\nabla_x F(x,y,z)\|^2 + \|\nabla_z F(x,y,z)\|^2 + \sum_{j \text{ active}} \big((\nabla_y F(x,y,z))_j\big)^2\Big)$$

LaSalle arguments show convergence, but not enough for

- characterization of convergence rate
- dealing with errors in computation/comm/sensing
- characterization of robustness against disturbances

Beyond Asymptotic Stability

Identification of Lyapunov functions of primal-dual dynamics
—leading to systematic characterization of convergence properties

Beyond Asymptotic Stability

Identification of Lyapunov functions of primal-dual dynamics
—leading to systematic characterization of convergence properties

Lyapunov Function for Constrained Optimization

For $f: \mathbb{R}^n \to \mathbb{R}$ strongly convex, twice continuously differentiable, $g: \mathbb{R}^n \to \mathbb{R}^p$ convex, twice continuously differentiable,

$$V(x,y,z) = \frac{1}{2}\|(x,y,z)\|^2_{\mathsf{Saddle}(F)} + V_m(x,y,z)$$

Beyond Asymptotic Stability

Identification of Lyapunov functions of primal-dual dynamics
—leading to systematic characterization of convergence properties

Lyapunov Function for Constrained Optimization

For $f : \mathbb{R}^n \to \mathbb{R}$ strongly convex, twice continuously differentiable, $g : \mathbb{R}^n \to \mathbb{R}^p$ convex, twice continuously differentiable,

$$V(x, y, z) = \frac{1}{2}\|(x, y, z)\|^2_{\text{Saddle}(F)} + V_m(x, y, z)$$

1. V **positive definite** with respect to Saddle(F)
2. $\dot{V}$ **negative definite:** $t \mapsto V(x(t), y(t), z(t))$ right-continuous, a.e. differentiable,
   - $\frac{d}{dt}V(x(t), y(t), z(t)) < 0$ for t where derivative exists & $(x(t), y(t), z(t)) \notin$ Saddle(F)
   - $V(x(t'), y(t'), z(t')) \leq \lim_{t \uparrow t'} V(x(t), y(t), z(t))$ for all $t' \geq 0$

A. Cherukuri, E. Mallada, S. H. Low, and J. Cortés. The role of convexity in saddle-point dynamics: Lyapunov function and robustness. *IEEE Transactions on Automatic Control*, 63(8):2449–2464, 2018

Implications

Algorithm robustness against disturbances

`true dynamics + disturbances`

- characterization of input-to-state stability properties of primal-dual dynamics
- graceful performance degradation as a function of size of disturbance

Implications

Algorithm robustness against disturbances

`true dynamics + disturbances`

- characterization of input-to-state stability properties of primal-dual dynamics
- graceful performance degradation as a function of size of disturbance

Real-time implementation of the dynamics

- adjust stepsize opportunistically based on system state
- aperiodic discrete-time implementation w/same convergence guarantees

Implications

Algorithm robustness against disturbances

`true dynamics + disturbances`

- characterization of input-to-state stability properties of primal-dual dynamics
- graceful performance degradation as a function of size of disturbance

Real-time implementation of the dynamics

- adjust stepsize opportunistically based on system state
- aperiodic discrete-time implementation w/same convergence guarantees

Tracking in **time-varying** optimization problems, **data-driven** implementations

`approx.dynamics = true dynamics + error`

- estimates/data in lieu of exact elements to close the loop/avoid expensive centralized computations/circumvent complex dynamics
- online formulations with tracking guarantees & handling of streaming data

Primal-Dual Dynamics as Versatile Tool

Characterization of rates, speed, and acceleration

N. K. Dhingra, S. Z. Khong, and M. R. Jovanović. The proximal augmented Lagrangian method for nonsmooth composite optimization.
IEEE Transactions on Automatic Control, 2018.
To appear

J. Cortés and S. K. Niederländer. Distributed coordination for nonsmooth convex optimization via saddle-point dynamics.
Journal of Nonlinear Science, 2019.
To appear

G. Qu and N. Li. On the exponential stability of primal-dual gradient dynamics.
2018.
Available online at https://arxiv.org/abs/1803.01825

W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: an exact first-order algorithm for decentralized consensus optimization.
SIAM Journal on Optimization, 25(2):944–966, 2015

M. McCreesh, J. Cortés, and B. Gharesifard. Accelerated convergence of saddle-point dynamics for convex-concave quadratic functions.
In *IEEE Conf. on Decision and Control*, Nice, France, December 2019.
Submitted

Graceful degradation as a function of size of disturbance

A. Cherukuri, E. Mallada, S. H. Low, and J. Cortés. The role of convexity in saddle-point dynamics: Lyapunov function and robustness.
IEEE Transactions on Automatic Control, 63(8):2449–2464, 2018

H. D. Nguyen, T. L. Vu, K. Turitsyn, and J. Slotine. Contraction and robustness of continuous time primal-dual dynamics.
IEEE Control Systems Letters, 2(4):755–760, 2018

Feedback-based, data-driven, online formulation, tracking guarantees

A. Bernstein and E. Dall'Anese. Real-time feedback-based optimization of distribution grids: a unified approach.
2017.
https://arxiv.org/abs/1711.01627

M. Colombino, E. Dall'Anese, and A. Bernstein. Online optimization as a feedback controller: Stability and tracking.
IEEE Transactions on Control of Network Systems, 2018.
Submitted

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

Continuous-time vs discrete-time dynamics, machine learning

Beyond Convergence #1: ISS with Respect to Saddle(F)

Robustness to errors in the gradient computation, noise in state measurements, errors in the controller implementation?

Theorem (Equality-Constrained Optimization)

For f strongly convex and C^2, $g = 0$, with $mI \preceq \nabla^2 f(x) \preceq MI \ \forall x \in \mathbb{R}^n$,

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\nabla_x F(x,z) \\ \nabla_z F(x,z) \end{bmatrix} + \begin{bmatrix} u_x \\ u_z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - A^\top z \\ Ax - b \end{bmatrix} + \begin{bmatrix} u_x \\ u_z \end{bmatrix}$$

is ISS with respect to Saddle(F)

Proof: $V_\beta(x,z) = \beta_1 \frac{1}{2} \|(x,z)\|^2_{\text{Saddle}(F)} + \beta_2 V_m(x,z)$ is ISS-Lyapunov function

Beyond Convergence #1: ISS with Respect to Saddle(F)

Robustness to errors in the gradient computation, noise in state measurements, errors in the controller implementation?

Conjecture (Constrained Optimization)

For f strongly convex and C^2, g convex and C^2, with $mI \preceq \nabla^2 f(x) \preceq MI$ $\forall x \in \mathbb{R}^n$,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\nabla_x F(x,y,z) \\ [\nabla_y F(x,y,z)]_y^+ \\ \nabla_z F(x,y,z) \end{bmatrix} + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - y^T \nabla g(x) - A^\top z \\ [g(x)]_y^+ \\ Ax - b \end{bmatrix} + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

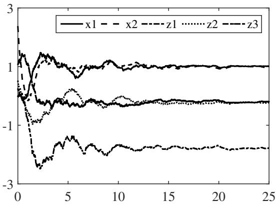
is ISS with respect to Saddle(F)

Proof: ISS-Lyapunov function theory for switched systems?

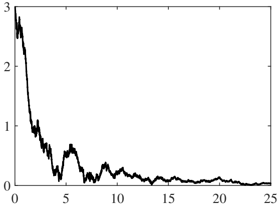
Example: Input-to-State Stability

$$f(x) = \|x\|^2, \quad A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

- $\text{Saddle}(F) = \{(x, z) \in \mathbb{R}^2 \times \mathbb{R}^3 \mid x = (1, 1), z = -(1, 1, 1) + \lambda(1, -1, -1), \lambda \in \mathbb{R}$
- f is $\mathcal{C}^2$, strongly convex, $\nabla^2 f(x) = 2I$



saddle-point dynamics



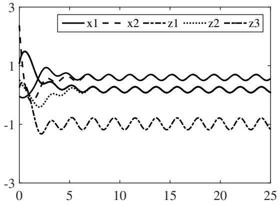
distance to saddle points

Vanishing disturbance

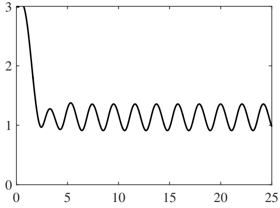
Example: Input-to-State Stability

$$f(x) = \|x\|^2, \quad A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

- $\text{Saddle}(F) = \{(x, z) \in \mathbb{R}^2 \times \mathbb{R}^3 \mid x = (1,1), z = -(1,1,1) + \lambda(1,-1,-1), \lambda \in \mathbb{R}$
- f is $\mathcal{C}^2$, strongly convex, $\nabla^2 f(x) = 2I$



saddle-point dynamics



distance to saddle points

"Constant + Sinusoid" disturbance

Beyond Convergence #2: Real Time Implementation

Opportunistic state-triggered implementation

- avoid continuous evaluation of the vector field
- adjust stepsize opportunistically based on state of the system

Beyond Convergence #2: Real Time Implementation

Opportunistic state-triggered implementation

- avoid continuous evaluation of the vector field
- adjust stepsize opportunistically based on state of the system

Given sequence of triggering time instants $\{t_k\}_{k=0}^{\infty}$,

$$\dot{x}(t) = -\nabla_x F(x(t_k), z(t_k))$$
$$\dot{z}(t) = \nabla_z F(x(t_k), z(t_k))$$

for $t \in [t_k, t_{k+1})$ and $k \in \mathbb{Z}_{\geq 0}$

Beyond Convergence #2: Real Time Implementation

Opportunistic state-triggered implementation

- avoid continuous evaluation of the vector field
- adjust stepsize opportunistically based on state of the system

Given sequence of triggering time instants $\{t_k\}_{k=0}^{\infty}$,

$$\dot{x}(t) = -\nabla_x F(x(t_k), z(t_k))$$
$$\dot{z}(t) = \nabla_z F(x(t_k), z(t_k))$$

for $t \in [t_k, t_{k+1})$ and $k \in \mathbb{Z}_{\geq 0}$

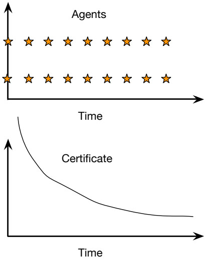
Objective: Design criterium to opportunistically select $\{t_k\}_{k=0}^{\infty}$ such that

- feasible executions: inter-trigger times lower bounded by positive quantity
- global asymptotic convergence is retained

Resource-Aware Control and Coordination

Continuous or **periodic** implementation paradigm

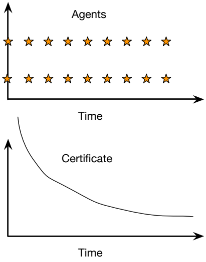
- costly-to-implement **synchronization** for information sharing, processing, decision making
- **'passive'** asynchronism, fixed agent time schedules
- **inefficient** implementations for processor usage, communication bandwidth, energy



Resource-Aware Control and Coordination

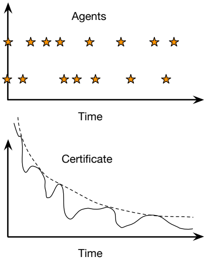
Continuous or **periodic** implementation paradigm

- costly-to-implement **synchronization** for information sharing, processing, decision making
- **'passive'** asynchronism, fixed agent time schedules
- **inefficient** implementations for processor usage, communication bandwidth, energy

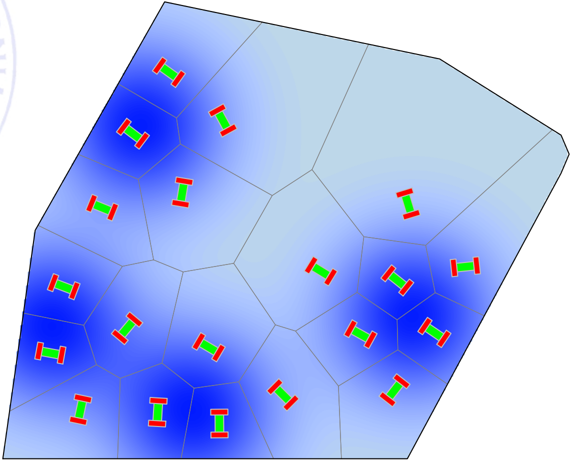


Opportunistic state-triggered paradigm

- **trade-offs**: comp, comm, sensing, control
- identify criteria to autonomously trigger actions based on task – **'active'** asynchronism
- **efficient** implementations, incorporates uncertainty



A Picture (A Movie) is Worth a Thousand Words



How to Decide When to Update?

Simplified setup: system $\dot{x} = f(x, u)$ on $\mathbb{R}^n$ with stabilization via

- controller: $k : \mathbb{R}^n \to \mathbb{R}^m$
- certificate: Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$

Synthesis for $\dot{x} = f(x, k(\bar{x}))$, w/ $\bar{x}$ sampled version of x?

How to Decide When to Update?

Simplified setup: system $\dot{x} = f(x, u)$ on $\mathbb{R}^n$ with stabilization via

- controller: $k : \mathbb{R}^n \to \mathbb{R}^m$
- certificate: Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$

Synthesis for $\dot{x} = f(x, k(\bar{x}))$, w/ $\bar{x}$ sampled version of x?

$$
\begin{aligned}
\dot{V} &= \nabla V(x) \cdot f(x, k(\bar{x})) \\
&= \nabla V(x) \cdot f(x, k(x)) + \nabla V(x) \cdot (f(x, k(\bar{x})) - f(x, k(x)))
\end{aligned}
$$

How to Decide When to Update?

Simplified setup: system $\dot{x} = f(x, u)$ on $\mathbb{R}^n$ with stabilization via

- controller: $k : \mathbb{R}^n \to \mathbb{R}^m$
- certificate: Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$

Synthesis for $\dot{x} = f(x, k(\bar{x}))$, w/ $\bar{x}$ sampled version of x?

$$\begin{aligned}\dot{V} &= \nabla V(x) \cdot f(x, k(\bar{x})) \\ &\leq \nabla V(x) \cdot f(x, k(x)) + h(x) \|\bar{x} - x\|\end{aligned}$$

How to Decide When to Update?

Simplified setup: system $\dot{x} = f(x, u)$ on $\mathbb{R}^n$ with stabilization via

- controller: $k : \mathbb{R}^n \to \mathbb{R}^m$
- certificate: Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$

Synthesis for $\dot{x} = f(x, k(\bar{x}))$, w/ $\bar{x}$ sampled version of x?

Trigger criterium: $\|\bar{x} - x\| \leq \frac{-\mathcal{L}_{f(x,k(x))} V(x)}{h(x)}$

How to Decide When to Update?

Simplified setup: system $\dot{x} = f(x,u)$ on $\mathbb{R}^n$ with stabilization via

- controller: $k : \mathbb{R}^n \to \mathbb{R}^m$
- certificate: Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$

Synthesis for $\dot{x} = f(x, k(\bar{x}))$, w/ $\bar{x}$ sampled version of x?

Trigger criterium: $\|\bar{x} - x\| \leq \frac{-\mathcal{L}_{f(x,k(x))} V(x)}{h(x)}$

Insights

- feasibility: guaranteed monotonic decrease of function, but aperiodic executions might not be feasible (accumulation of triggered times, Zeno)
- certificate: LaSalle function clearly not good enough
- trigger: specific challenges for network systems, both in design (local triggers) and analysis (asynchronism, Zeno)
- trigger: stabilization versus optimization

State-Triggered Implementation for Primal-Dual Dynamics

Given sequence of triggering time instants $\{t_k\}_{k=0}^{\infty}$,

$$\dot{x}(t) = -\nabla_x F(x(t_k), z(t_k))$$
$$\dot{z}(t) = \nabla_z F(x(t_k), z(t_k))$$

for $t \in [t_k, t_{k+1})$ and $k \in \mathbb{Z}_{\geq 0}$

Approach: Use V_β to design criterium to opportunistically select $\{t_k\}_{k=0}^{\infty}$

State-Triggered Implementation for Primal-Dual Dynamics

Given sequence of triggering time instants $\{t_k\}_{k=0}^{\infty}$,

$$\dot{x}(t) = -\nabla_x F(x(t_k), z(t_k))$$
$$\dot{z}(t) = \nabla_z F(x(t_k), z(t_k))$$

for $t \in [t_k, t_{k+1})$ and $k \in \mathbb{Z}_{\geq 0}$

State-triggered criterium

$$t_{k+1} = t_k - \frac{\mathcal{L}_{X_{\text{sp}}} V_\beta(x(t_k), z(t_k))}{\xi(x(t_k), z(t_k)) \|X_{\text{sp}}(x(t_k), z(t_k))\|^2}$$

Next triggering time computable with information available at current one

State-Triggered Implementation for Primal-Dual Dynamics

Given sequence of triggering time instants $\{t_k\}_{k=0}^{\infty}$,

$$\dot{x}(t) = -\nabla_x F(x(t_k), z(t_k))$$
$$\dot{z}(t) = \nabla_z F(x(t_k), z(t_k))$$

for $t \in [t_k, t_{k+1})$ and $k \in \mathbb{Z}_{\geq 0}$

Theorem

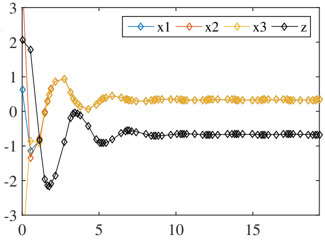
For f strongly convex and C^2, with $mI \preceq \nabla^2 f(x) \preceq MI$ and $x \mapsto \nabla^2 f(x)$ Lipschitz, A full row rank,

- *trajectories of self-triggered dynamics converge to (x_*, z_*)*
- *inter-event times are lower bounded by positive quantity*

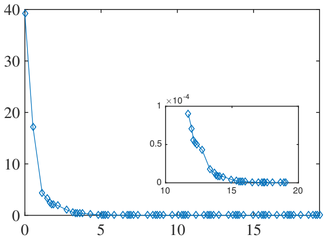
Example: Self-Triggered Implementation

$$F(x,z) = \|x\|^2 + z(x_1 + x_2 + x_3 - 1).$$

- $f(x) = \|x\|^2$ satisfies hypotheses, $A = [1,1,1]$ full row-rank
- $\text{Saddle}(F) = \{((\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), -\frac{2}{3})\}$



primal-dual dynamics



V_β

Beyond Convergence #3: Tracking in Time-Varying Opt

Time-varying optimization problems

$$\begin{aligned} \text{minimize} \quad & f_t(x) \\ \text{subject to} \quad & g_t(x) \leq 0 \\ & h_t(x) = 0 \end{aligned}$$

ISS characterization leads to tracking guarantees

$$\limsup_{\tau \to \infty} \|(x(\tau), y(\tau), z(\tau)) - (x^*(t), y^*(t), z^*(t))\| \leq \gamma(c\delta)$$

where γ is class $\mathcal{K}$ function and

- δ is upper bound on rate of change of saddle points

$$\|\frac{d}{dt}(x^*(t), y^*(t), z^*(t))\| \leq \delta$$

- c time scale of primal-dual dynamics

$$\frac{d}{d\tau} = c\frac{d}{dt}$$

Dynamic Traffic Assignment

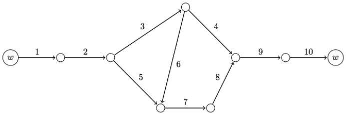
Dynamic traffic assignment problem tunes traffic flows to optimize total travel distance, total travel time given time-varying inflows to network

Cell transmission model

$x(t)$	– traffic volume in cells at time t
$f(t)$	– flows between cells at time t
$\lambda(t)$	– inflow to the network
$\mu(t)$	– outflow from the network

supply&demand functions of infrastructure

+



Dynamic Traffic Assignment

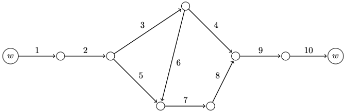
Dynamic traffic assignment problem tunes traffic flows to optimize total travel distance, total travel time given time-varying inflows to network

Cell transmission model

$x(t)$ – traffic volume in cells at time t
$f(t)$ – flows between cells at time t
$\lambda(t)$ – inflow to the network
$\mu(t)$ – outflow from the network

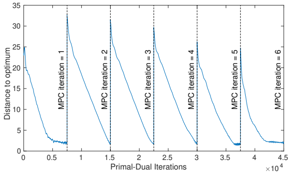
supply&demand functions of infrastructure

+



Distributed primal-dual dynamics to solve MPC formulation

- Two groups of 5 cars enter network at time 1 (red) and 3 (blue)
- Capacity in cell 4 drops to zero at time 5 due to an accident



Dynamic Traffic Assignment

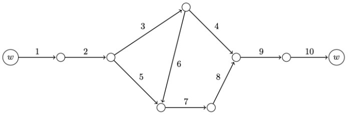
Dynamic traffic assignment problem tunes traffic flows to optimize total travel distance, total travel time given time-varying inflows to network

Cell transmission model

$x(t)$	– traffic volume in cells at time t
$f(t)$	– flows between cells at time t
$\lambda(t)$	– inflow to the network
$\mu(t)$	– outflow from the network

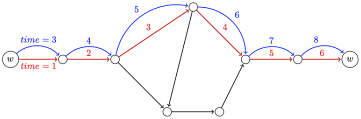
supply&demand functions of infrastructure

$+$



Distributed primal-dual dynamics to solve MPC formulation

- Two groups of 5 cars enter network at time 1 (red) and 3 (blue)
- Capacity in cell 4 drops to zero at time 5 due to an accident



Dynamic Traffic Assignment

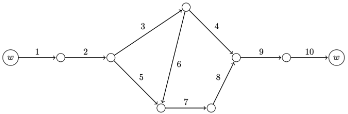
Dynamic traffic assignment problem tunes traffic flows to optimize total travel distance, total travel time given time-varying inflows to network

Cell transmission model

$x(t)$ – traffic volume in cells at time t
$f(t)$ – flows between cells at time t
$\lambda(t)$ – inflow to the network
$\mu(t)$ – outflow from the network

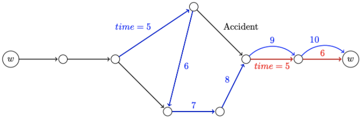
supply&demand functions of infrastructure

+



Distributed primal-dual dynamics to solve MPC formulation

- Two groups of 5 cars enter network at time 1 (red) and 3 (blue)
- Capacity in cell 4 drops to zero at time 5 due to an accident

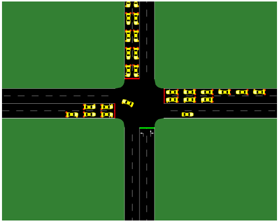


M. Vaquero and J. Cortés. Distributed augmentation-regularization for robust online convex optimization. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 230–235, Groningen, The Netherlands, 2018

Beyond Convergence #4: Coupled w/Network Dynamics

Select optimized setpoints x (power injection of distributed energy resources, traffic flows among contiguous arterial roads)

$$\begin{aligned} \min \quad & f(x, y(x)) \\ \text{subject to} \quad & g(x, y(x)) \leq 0 \\ & h(x, y(x)) = 0 \end{aligned}$$



that drive physical $y(x)$ (bus frequencies/voltages, traffic density) dynamics

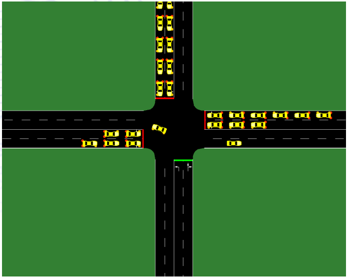
$$\begin{aligned} \dot{\xi} &= \Phi(\xi, x) \\ y &= \Psi(\xi, d) \end{aligned}$$

Implementation of primal-dual dynamics requires evaluation of $y(x)$, $\frac{\partial y}{\partial x}(x)$

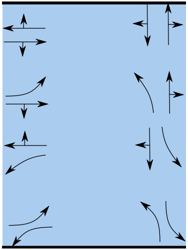
substitute by data/high-fidelity sim/estimates $\Rightarrow$ *approx. primal-dual dynamics*

Traffic Intersection Control

with Simulation of Urban MObility (SUMO) simulator



Stages



Controlled traffic light (green-red stage time)

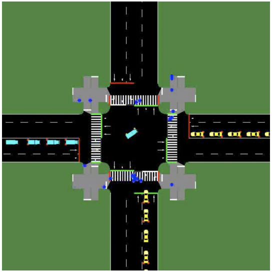
- Total cycle time = 160 sec, s_i = is time of stage i
- Flows = 10 cars/cycle, but and = 100 cars/cycle
- $\mathcal{F}(s)$ is number of cars out given s and flows

Data-driven Optimization of Stage Times

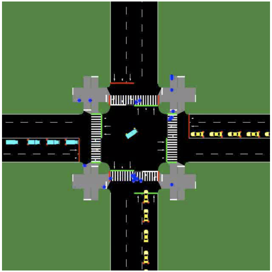
SUMO is complex agent-based simulator that incorporates

- network characteristics: road topology, traffic lights, sidewalks
- number of vehicles entering network at any time and any lane
- destination of vehicles: route, turning ratios, origin/destination
- type of vehicle: car, truck, bus, van, bicycle, pedestrian (w/ length, width, acceleration, max speed)
- driver's behavior: intersection model, lane changing model, car following model

$$
\begin{aligned}
\text{minimize} \quad & -x \\
\text{subject to} \quad & x = \mathcal{F}_t(s) \\
& \sum_{i=1}^{8} s_i = 160 \\
& s_i \geq 5
\end{aligned}
$$



data-driven optimized strategy



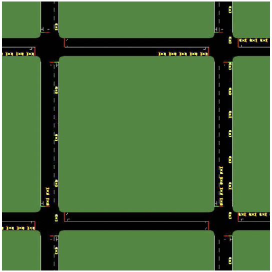
constant, equal stage times

Data-driven Optimization of Stage Times

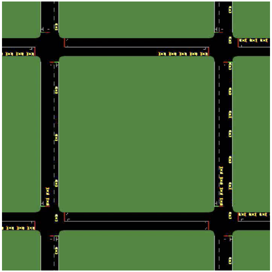
SUMO is complex agent-based simulator that incorporates

- network characteristics: road topology, traffic lights, sidewalks
- number of vehicles entering network at any time and any lane
- destination of vehicles: route, turning ratios, origin/destination
- type of vehicle: car, truck, bus, van, bicycle, pedestrian (w/ length, width, acceleration, max speed)
- driver's behavior: intersection model, lane changing model, car following model

$$\begin{aligned} \text{minimize} \quad & -x \\ \text{subject to} \quad & x = \mathcal{F}_t(s) \\ & \sum_{i=1}^{8} s_i = 160 \\ & s_i \geq 5 \end{aligned}$$



data-driven optimized strategy



adaptive signal control – Webster

Taking the Basic Taxonomy Further

$$\begin{aligned} \text{minimize} \quad & \text{‘aggregate objective function’}(x) \\ \text{subject to} \quad & \text{‘ineq constraints’}(x) \\ & \text{‘eq constraints’}(x) \end{aligned}$$

- **Hedging against uncertainty** via data-driven distributionally-robust optimization

  A. Cherukuri and J. Cortés. Cooperative data-driven distributionally robust optimization.
  IEEE Transactions on Automatic Control, 2018.
  Submitted

- **Protecting privacy** of individual information via differential privacy

  E. Nozari, P. Tallapragada, and J. Cortés. Differentially private distributed convex optimization via functional perturbation.
  IEEE Transactions on Control of Network Systems, 5(1):395–408, 2018

- **Multi-layer optimization:** competition+coordination in power systems

  A. Cherukuri and J. Cortés. Iterative bidding in electricity markets: rationality and robustness.
  IEEE Transactions on Network Science and Engineering, 2018.
  Submitted

  P. Srivastava, C.-Y. Chang, and J. Cortés. Participation of microgrids in frequency regulation markets.
  In *American Control Conference*, pages 3834–3839, Milwaukee, WI, May 2018

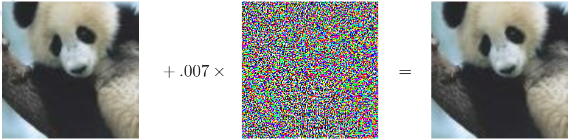
- Dealing with **non-sparse constraints** through data

  C.-Y. Chang, M. Colombino, J. Cortés, and E. Dall'Anese. Saddle-flow dynamics for distributed feedback-based optimization.
  IEEE Control Systems Letters, 2019.
  Submitted

Hedging Against Uncertainty

Stochastic optimization works well when distribution is known and datasets are large

Risky with "uncertainty about uncertainty"



$\boldsymbol{x}$ "panda" 57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "nematode" 8.2% confidence

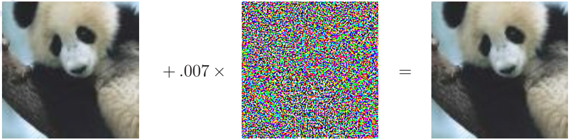
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "gibbon" 99.3 % confidence

[Goodfellow, Shlens, Szegedy '15]

Hedging Against Uncertainty

Stochastic optimization works well when distribution is known and datasets are large

Risky with "uncertainty about uncertainty"



$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda"	"nematode"	"gibbon"
57.7% confidence	8.2% confidence	99.3 % confidence

[Goodfellow, Shlens, Szegedy '15]

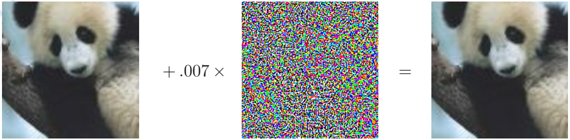
Many scenarios w/decisions need to be made before large datasets can be collected

- deadlines imposed by performance or safety considerations
- timescale of system's evolution faster than speed at which data can be collected
- acquiring samples is expensive: adversary purposely hides in environment

Hedging Against Uncertainty

Stochastic optimization works well when distribution is known and datasets are large

Risky with "uncertainty about uncertainty"



$\boldsymbol{x}$ "panda" 57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "nematode" 8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "gibbon" 99.3 % confidence

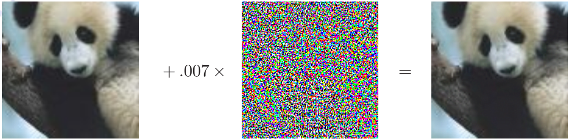
[Goodfellow, Shlens, Szegedy '15]

Traditional robust optimization might be overly conservative

Hedging Against Uncertainty

Stochastic optimization works well when distribution is known and datasets are large

Risky with "uncertainty about uncertainty"



$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda"	"nematode"	"gibbon"
57.7% confidence	8.2% confidence	99.3 % confidence

[Goodfellow, Shlens, Szegedy '15]

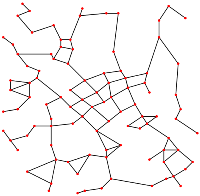
The best of both worlds: **distributionally robust optimization**

empirical samples + 'what-if' scenarios

Attractive b/c of formal guarantees valid for small datasets

Network Optimization under Uncertainty

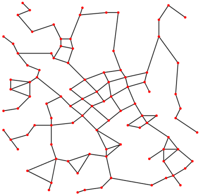
Stochastic optimization: $\inf_{x \in \mathcal{X} \subset \mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(x,\xi)]$



- objective $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ encodes network goal
  —total travel time, total distance covered, -network outflow
- decision variable x at central/aggregate/local level
  —max velocity, inflows at control nodes, routing at intersections
- random variable ξ distributed w/ unknown probability $\mathbb{P}$
  —inflows/outflows at non-control nodes, road densities, vehicle locations

Network Optimization under Uncertainty

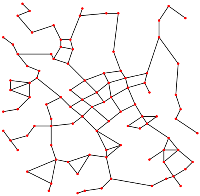
Stochastic optimization: $\inf_{x \in \mathcal{X} \subset \mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(x,\xi)]$



- objective $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ encodes network goal
  —total travel time, total distance covered, -network outflow
- decision variable x at central/aggregate/local level
  —max velocity, inflows at control nodes, routing at intersections
- random variable ξ distributed w/ unknown probability $\mathbb{P}$
  —inflows/outflows at non-control nodes, road densities, vehicle locations

Network Optimization under Uncertainty

Stochastic optimization: $\inf_{x \in \mathcal{X} \subset \mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(x, \xi)]$



- objective $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ encodes network goal

  —total travel time, total distance covered, -network outflow
- decision variable x at central/aggregate/local level

  —max velocity, inflows at control nodes, routing at intersections
- random variable ξ distributed w/ unknown probability $\mathbb{P}$

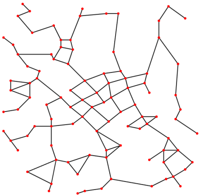
  —inflows/outflows at non-control nodes, road densities, vehicle locations

Given N i.i.d samples $\{\widehat{\xi}^k\}_{k=1}^N$, discrete empirical probability distribution

$$\hat{\mathbb{P}}_N = \frac{1}{N} \sum_{k=1}^{N} \delta_{\widehat{\xi}^k}$$

Network Optimization under Uncertainty

Stochastic optimization: $\inf_{x \in \mathcal{X} \subset \mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(x,\xi)]$



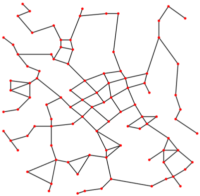
- objective $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ encodes network goal
  —total travel time, total distance covered, -network outflow
- decision variable x at central/aggregate/local level
  —max velocity, inflows at control nodes, routing at intersections
- random variable ξ distributed w/ unknown probability $\mathbb{P}$
  —inflows/outflows at non-control nodes, road densities, vehicle locations

Given N i.i.d samples $\{\widehat{\xi}^k\}_{k=1}^N$, discrete empirical probability distribution

$$\hat{\mathbb{P}}_N = \frac{1}{N}\sum_{k=1}^N \delta_{\widehat{\xi}^k} \qquad \mathbb{E}_{\hat{\mathbb{P}}_N}[f(x,\xi)] = \frac{1}{N}\sum_{i=1}^N f(x,\xi^k)$$

Network Optimization under Uncertainty

Stochastic optimization: $\inf_{x \in \mathcal{X} \subset \mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(x,\xi)]$



- objective $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ encodes network goal
  —total travel time, total distance covered, -network outflow
- decision variable x at central/aggregate/local level
  —max velocity, inflows at control nodes, routing at intersections
- random variable ξ distributed w/ unknown probability $\mathbb{P}$
  —inflows/outflows at non-control nodes, road densities, vehicle locations

Given N i.i.d samples $\{\widehat{\xi}^k\}_{k=1}^N$, discrete empirical probability distribution

$$\hat{\mathbb{P}}_N = \frac{1}{N}\sum_{k=1}^{N} \delta_{\widehat{\xi}^k} \qquad \mathbb{E}_{\hat{\mathbb{P}}_N}[f(x,\xi)] = \frac{1}{N}\sum_{i=1}^{N} f(x,\xi^k)$$

Sample-average approximation has

- almost sure convergence guarantee as $N \to \infty$
- poor out-of-sample performance for small N

Distributionally Robust Optimization

Account for ignorance of true data-generating distribution $\mathbb{P}$

Distributionally robust (DRO) formulation to hedge against uncertainty

$$\inf_{x \in \mathcal{X}} \sup_{\mathbb{Q} \in \widehat{\mathcal{P}}_N} \mathbb{E}_{\mathbb{Q}}[f(x, \xi)]$$

$\widehat{\mathcal{P}}_N$ is ambiguity set of probability distributions

Distributionally Robust Optimization

Account for ignorance of true data-generating distribution $\mathbb{P}$

Distributionally robust (DRO) formulation to hedge against uncertainty

$$\inf_{x \in \mathcal{X}} \sup_{\mathbb{Q} \in \widehat{\mathcal{P}}_N} \mathbb{E}_{\mathbb{Q}}[f(x, \xi)]$$

$\widehat{\mathcal{P}}_N$ is ambiguity set of probability distributions

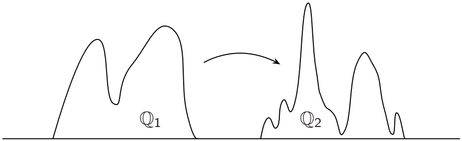
Desirable properties in ambiguity set

- rich enough to contain true data-generating distribution with high confidence
- small enough to exclude pathological distributions (avoid conservativeness)
- easy to parameterize from data
- facilitate tractable solution of optimization problem

Wasserstein metric: cost of optimal transportation plan of probability mass

$$d_{W_2}(\mathbb{Q}_1, \mathbb{Q}_2) = \left(\inf \left\{ \int_{\Xi^2} \|\xi_1 - \xi_2\|^2 \Pi(d\xi_1, d\xi_2) \;\middle|\; \Pi \in \mathcal{H}(\mathbb{Q}_1, \mathbb{Q}_2) \right\} \right)^{\frac{1}{2}}$$

—$\mathcal{H}(\mathbb{Q}_1, \mathbb{Q}_2)$ is set of distributions w/ marginals $\mathbb{Q}_1$ and $\mathbb{Q}_2$



Π is transportation plan for moving mass distribution

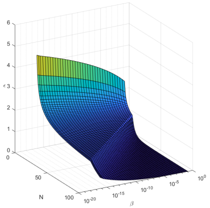
norm $\|\cdot\|$ encodes transportation cost

Data-Driven DRO

Ambiguity set via Wasserstein metric

$$\widehat{\mathcal{P}}_N = \mathbb{B}_{\epsilon_N(\beta)}(\hat{\mathbb{P}}_N)$$

Then, $\mathbb{P}^N(\mathbb{P} \in \widehat{\mathcal{P}}_N) \geq 1 - \beta$

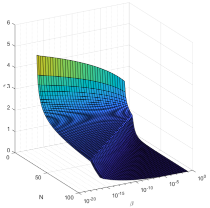


Data-Driven DRO

Ambiguity set via Wasserstein metric

$$\widehat{\mathcal{P}}_N = \mathbb{B}_{\epsilon_N(\beta)}(\hat{\mathbb{P}}_N)$$

Then, $\mathbb{P}^N(\mathbb{P} \in \widehat{\mathcal{P}}_N) \geq 1 - \beta$



Out-of-sample guarantee & tractability [Esfahani & Kuhn '16]

Let $\widehat{J}_N$, $\widehat{x}_N$ be optimal value, optimizer of DRO. Then

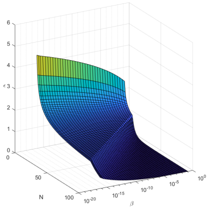
$$\mathbb{P}^N\big(\mathbb{E}_{\mathbb{P}}[f(\widehat{x}_N, \xi)] \leq \widehat{J}_N\big) \geq 1 - \beta$$

Data-Driven DRO

Ambiguity set via Wasserstein metric

$$\widehat{\mathcal{P}}_N = \mathbb{B}_{\epsilon_N(\beta)}(\hat{\mathbb{P}}_N)$$

Then, $\mathbb{P}^N(\mathbb{P} \in \widehat{\mathcal{P}}_N) \geq 1 - \beta$



Out-of-sample guarantee & tractability [Esfahani & Kuhn '16]

Let $\widehat{J}_N$, $\widehat{x}_N$ be optimal value, optimizer of DRO. Then

$$\mathbb{P}^N\Big(\mathbb{E}_{\mathbb{P}}[f(\widehat{x}_N, \xi)] \leq \widehat{J}_N\Big) \geq 1 - \beta$$

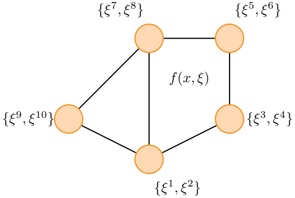
Additionally, if $x \mapsto f(x, \xi)$ convex $\forall \xi \in \Xi$, then $\widehat{J}_N$ equal to convex optimization

$$\inf_{\lambda \geq 0, x \in \mathcal{X}} \left\{ \lambda \epsilon_N^2(\beta) + \frac{1}{N} \sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m} \Big(f(x, \xi) - \lambda \|\xi - \hat{\xi}^k\|^2 \Big) \right\}$$

Network Optimization Problem

Cooperative network of n agents collecting data, communicating w/ neighbors, no central coordinator

- communication modeled by graph $(\mathcal{V}, \mathcal{E})$
- each agent knows objective function f, constraint set $\mathcal{X}$
- each agent gathers i.i.d samples $\widehat{\Xi}_i$ ($\widehat{\Xi}_i \cap \widehat{\Xi}_j = \emptyset$)
- data available across network $\cup_{i=1}^n \widehat{\Xi}_i = \{\widehat{\xi}^k\}_{k=1}^N$



Leverage **power-of-many** to collectively improve out-of-sample guarantee

- individual agents can solve DRO with own data, but
- can benefit from others' contributions to obtain higher-quality solution

Distributed Reformulation of Data-Driven DRO

Each agent i with own estimate x^i (and λ^i) of optimal solution

$$\begin{aligned} \min_{x_v, \lambda_v \geq \mathbf{0}_n} \quad & \frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \frac{1}{N}\sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m} \Big(f(x^{v_k}, \xi) - \lambda^{v_k} \|\xi - \widehat{\xi}^k\|^2 \Big) \\ \text{subject to} \quad & \mathsf{L}\lambda_v = \mathbf{0}_n \quad \text{and} \quad (\mathsf{L} \otimes \mathsf{I}_d) x_v = \mathbf{0}_{nd} \end{aligned} \qquad (\star)$$

(Here $x_v = (x^1; \ldots; x^n)$, $\lambda_v = (\lambda^1; \ldots; \lambda^n)$)

Distributed Reformulation of Data-Driven DRO

Each agent i with own estimate x^i (and λ^i) of optimal solution

$$\min_{x_v, \lambda_v \geq \mathbf{0}_n} \quad \frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \frac{1}{N}\sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m} \Big(f(x^{v_k}, \xi) - \lambda^{v_k} \|\xi - \widehat{\xi}^k\|^2 \Big) \qquad (\star)$$

$$\text{subject to} \quad \underbrace{L\lambda_v = \mathbf{0}_n}_{\text{agreement on } \lambda^i\text{'s}} \quad \text{and} \quad \underbrace{(L \otimes I_d)x_v = \mathbf{0}_{nd}}_{\text{agreement on } x^i\text{'s}}$$

(Here $x_v = (x^1; \dots; x^n)$, $\lambda_v = (\lambda^1; \dots; \lambda^n)$)

Distributed Reformulation of Data-Driven DRO

Each agent i with own estimate x^i (and λ^i) of optimal solution

$$\min_{x_v, \lambda_v \geq \mathbf{0}_n} \quad \frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \frac{1}{N}\sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m}\Big(f(x^{v_k}, \xi) - \lambda^{v_k}\|\xi - \widehat{\xi}^k\|^2\Big) \qquad (\star)$$

$$\text{subject to} \quad \underbrace{\mathsf{L}\lambda_v = \mathbf{0}_n}_{\text{agreement on } \lambda^i\text{'s}} \quad \text{and} \quad \underbrace{(\mathsf{L} \otimes \mathsf{I}_d)x_v = \mathbf{0}_{nd}}_{\text{agreement on } x^i\text{'s}}$$

(Here $x_v = (x^1; \dots; x^n)$, $\lambda_v = (\lambda^1; \dots; \lambda^n)$)

Problems are equivalent (w/ f convex in x)

- Given $\widehat{x}_N$, there exists $\lambda^* \geq 0$ s.t. $(\mathbf{1}_n \otimes \widehat{x}_N, \lambda^* \mathbf{1}_n)$ is optimizer of $(\star)$
- If (x_v^*, λ_v^*) is optimizer of $(\star)$, then $x_v^* = \mathbf{1}_n \otimes \widehat{x}_N$

Same optimal value $\widehat{J}_N$

Optimization $(\star)$ has separable objective & locally computable constraints!

Modified Lagrangian

Getting rid of inner maximization in Lagrangian

$$\textbf{Lagrangian:} \quad L(x_v, \lambda_v, \nu, \eta) := \frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m} \Big(f(x^{v_k}, \xi) - \lambda^{v_k} \|\xi - \widehat{\xi}^k\|^2 \Big) + \nu^\top L \lambda_v + \eta^\top (L \otimes I_d) x_v$$

Modified Lagrangian

Getting rid of inner maximization in Lagrangian

Lagrangian: $$L(x_v, \lambda_v, \nu, \eta) := \frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m} \Big(f(x^{v_k}, \xi) - \lambda^{v_k} \|\xi - \widehat{\xi}^k\|^2 \Big) + \nu^\top \mathsf{L}\lambda_v + \eta^\top (\mathsf{L} \otimes \mathsf{I}_d) x_v$$

Augmented Lagrangian: (for better convergence properties)

$$L_{\text{aug}}(x_v, \lambda_v, \nu, \eta) := L(x_v, \lambda_v, \nu, \eta) + \frac{1}{2} x_v^\top (\mathsf{L} \otimes \mathsf{I}_d) x_v + \frac{1}{2} \lambda_v^\top \mathsf{L} \lambda_v$$

Modified Lagrangian

Getting rid of inner maximization in Lagrangian

Lagrangian: $$L(x_v, \lambda_v, \nu, \eta) := \frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \sum_{k=1}^{N} \max_{\xi \in \mathbb{R}^m} \Big(f(x^{v_k}, \xi) - \lambda^{v_k} \|\xi - \widehat{\xi}^k\|^2 \Big) + \nu^\top \mathsf{L}\lambda_v + \eta^\top (\mathsf{L} \otimes \mathsf{I}_d) x_v$$

Augmented Lagrangian: (for better convergence properties)

$$\begin{aligned} L_{\text{aug}}(x_v, \lambda_v, \nu, \eta) &:= L(x_v, \lambda_v, \nu, \eta) + \frac{1}{2} x_v^\top (\mathsf{L} \otimes \mathsf{I}_d) x_v + \frac{1}{2} \lambda_v^\top \mathsf{L} \lambda_v \\ &= \max_{\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(x_v, \lambda_v, \nu, \eta, \{\xi^{(k)}\}) \end{aligned}$$

$$\begin{aligned} \tilde{L}_{\text{aug}}(x_v, \lambda_v, \nu, \eta, \{\xi^{(k)}\}) := &\frac{\epsilon_N^2(\beta)\mathbf{1}_n^\top \lambda_v}{n} + \sum_{k=1}^{N} \Big(f(x^{v_k}, \xi) - \lambda^{v_k} \|\xi - \widehat{\xi}^k\|^2 \Big) \\ &+ \nu^\top \mathsf{L}\lambda_v + \eta^\top (\mathsf{L} \otimes \mathsf{I}_d) x_v + \frac{1}{2} x_v^\top (\mathsf{L} \otimes \mathsf{I}_d) x_v + \frac{1}{2} \lambda_v^\top \mathsf{L} \lambda_v \end{aligned}$$

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_{\text{v}},\lambda_{\text{v}}\geq \mathbf{0}_n} \max_{\nu,\eta} L_{\text{aug}}(x_{\text{v}},\lambda_{\text{v}},\nu,\eta) = \max_{\nu,\eta} \min_{x_{\text{v}},\lambda_{\text{v}}\geq \mathbf{0}_n} L_{\text{aug}}(x_{\text{v}},\lambda_{\text{v}},\nu,\eta)$$

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_{\text{v}},\lambda_{\text{v}}\geq \mathbf{0}_n} \max_{\nu,\eta} L_{\text{aug}}(x_{\text{v}},\lambda_{\text{v}},\nu,\eta) = \max_{\nu,\eta} \min_{x_{\text{v}},\lambda_{\text{v}}\geq \mathbf{0}_n} L_{\text{aug}}(x_{\text{v}},\lambda_{\text{v}},\nu,\eta)$$

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_{\text{v}}, \lambda_{\text{v}} \geq \mathbf{0}_n} \max_{\nu, \eta} \max_{\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot) = \max_{\nu, \eta} \min_{x_{\text{v}}, \lambda_{\text{v}} \geq \mathbf{0}_n} \max_{\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot)$$

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_{\text{v}},\lambda_{\text{v}}\geq\mathbf{0}_n}\ \max_{\nu,\eta,\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot) = \max_{\nu,\eta}\ \min_{x_{\text{v}},\lambda_{\text{v}}\geq\mathbf{0}_n}\ \max_{\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot)$$

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_v, \lambda_v \geq \mathbf{0}_n} \max_{\nu, \eta, \{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot) = \max_{\nu, \eta} \min_{x_v, \lambda_v \geq \mathbf{0}_n} \max_{\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot)$$

assuming **min-max** operator on the right can be **interchanged** – requires formal proof

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_{\text{v}},\lambda_{\text{v}}\geq \mathbf{0}_n} \max_{\nu,\eta,\{\xi^{(k)}\}} \tilde{L}_{\text{aug}}(\cdot) = \max_{\nu,\eta,\{\xi^{(k)}\}} \min_{x_{\text{v}},\lambda_{\text{v}}\geq \mathbf{0}_n} \tilde{L}_{\text{aug}}(\cdot)$$

Modified Lagrangian

Saddle points of L_{aug} exists implying

$$\min_{x_{\mathrm{v}},\lambda_{\mathrm{v}}\geq \mathbf{0}_n} \max_{\nu,\eta,\{\xi^{(k)}\}} \tilde{L}_{\mathrm{aug}}(\cdot) = \max_{\nu,\eta,\{\xi^{(k)}\}} \min_{x_{\mathrm{v}},\lambda_{\mathrm{v}}\geq \mathbf{0}_n} \tilde{L}_{\mathrm{aug}}(\cdot)$$

Correspondence between optima and saddle points

1. If $(x_{\mathrm{v}}^*, \lambda_{\mathrm{v}}^*, \nu^*, \eta^*)$ is saddle point of L, then $\exists \{(\xi^*)^{(k)}\}$ such that $((x_{\mathrm{v}}^*, \lambda_{\mathrm{v}}^*, \nu^*, \eta^*, \{(\xi^*)^{(k)}\})$ is saddle point of $\tilde{L}_{\mathrm{aug}}$ over $\lambda_{\mathrm{v}} \geq \mathbf{0}_n$
2. If $((x_{\mathrm{v}}^*, \lambda_{\mathrm{v}}^*, \nu^*, \eta^*, \{(\xi^*)^{(k)}\})$ is saddle point of $\tilde{L}_{\mathrm{aug}}$ over $\lambda_{\mathrm{v}} \geq \mathbf{0}_n$, then $(x_{\mathrm{v}}^*, \lambda_{\mathrm{v}}^*, \nu^*, \eta^*)$ is saddle point of L

$\tilde{L}_{\mathrm{aug}}$ is **convex-concave** in $((x_{\mathrm{v}}, \lambda_{\mathrm{v}}), (\nu, \eta, \{\xi^{(k)}\}))$ over domain $\lambda_{\mathrm{v}} \geq \mathbf{0}_n$

When Can Max-Min Operator Be Interchanged?

Theorem

Assuming f satisfies technical condition on directions of recession. Max-min operator can be interchanged under either

1. *convex-concave objective function f*
2. *convex-convex objective function f and*
   - *quadratic in ξ,*

     $$f(x, \xi) = \xi^\top Q \xi + x^\top R \xi + \ell(x)$$

   - *least-squares problem (w/ $d = m$),*

     $$f(x, \xi) = a(\xi_m - (\xi_{1:m-1}; 1)^\top x)^2$$

In either case, $\tilde{L}_{\text{aug}}$ is convex-concave in variables $((x_v, \lambda_v), \{\xi^{(k)}\})$

Distributed Algorithm for Network Optimization

Primal-dual dynamics for $\tilde{L}_{\text{aug}}$ is distributed

$$\begin{aligned}
\frac{dx_v}{dt} &= -\Pr_{\mathcal{X}}(\nabla_{x_v}\tilde{L}_{\text{aug}}(x_v,\lambda_v,\nu,\eta,\{\xi^{(k)}\})) \\
\frac{d\lambda_v}{dt} &= [-\nabla_{\lambda_v}\tilde{L}_{\text{aug}}(x_v,\lambda_v,\nu,\eta,\{\xi^{(k)}\})]^+_{\lambda_v} \\
\frac{d\nu}{dt} &= \nabla_{\nu}\tilde{L}_{\text{aug}}(x_v,\lambda_v,\nu,\eta,\{\xi^{(k)}\}) \\
\frac{d\eta}{dt} &= \nabla_{\eta}\tilde{L}_{\text{aug}}(x_v,\lambda_v,\nu,\eta,\{\xi^{(k)}\}) \\
\frac{d\xi^{(k)}}{dt} &= \nabla_{\xi^{(k)}}\tilde{L}_{\text{aug}}(x_v,\lambda_v,\nu,\eta,\{\xi^{(k)}\}),\ \forall k \in \{1,\dots,N\}
\end{aligned}$$

$\tilde{L}_{\text{aug}}$ not necessarily strictly convex in (x_v,λ_v), not linear in $\{\xi^k\}$

Distributed Algorithm for Network Optimization

Primal-dual dynamics for $\tilde{L}_{\text{aug}}$ is distributed

$$\frac{dx^i}{dt} = \frac{1}{N}\sum_{k\in\mathcal{K}_i} \nabla_{x^i} g_k(x^i,\lambda^i,\xi^k) + \sum_{j\in\mathcal{N}_i}\Big((\eta^i-\eta^j)+(x^i-x^j)\Big)$$

$$\frac{d\lambda^i}{dt} = \Big[\frac{\epsilon_N^2(\beta)}{n} + \frac{1}{N}\sum_{k\in\mathcal{K}_i} \nabla_{\lambda^i} g_k(x^i,\lambda^i,\xi^k) + \sum_{j\in\mathcal{N}_i}\Big((\nu^i-\nu^j)+(\lambda^i-\lambda^j)\Big)\Big]^+_{\lambda^i}$$

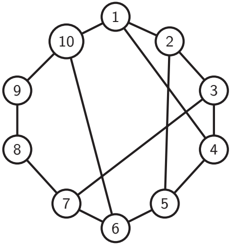
$$\frac{d\nu^i}{dt} = \sum_{j\in\mathcal{N}_i} a_{ij}(\lambda^i-\lambda^j)$$

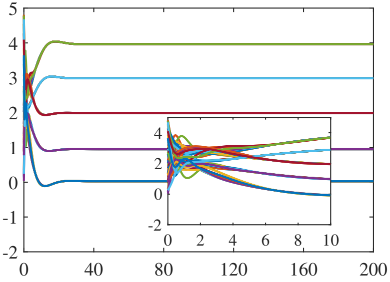
$$\frac{d\eta^i}{dt} = \sum_{j\in\mathcal{N}_i} a_{ij}(x^i-x^j)$$

$$\frac{d\xi^k}{dt} = \frac{1}{N}\nabla_\xi g_k(x^i,\lambda^i,\xi^k),\ \forall k\in\mathcal{K}_i \qquad [g_k(x,\lambda,\xi) := f(x,\xi) - \lambda\|\xi-\boldsymbol{k}\|^2]$$

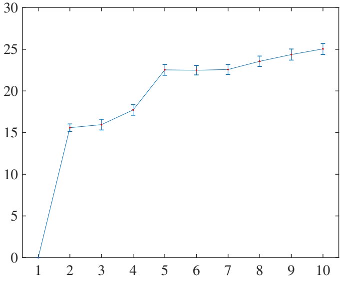
Illustration

- data $\widehat{\xi}^k = (\widehat{w}^k, \widehat{y}^k) \in \mathbb{R}^4 \times \mathbb{R}$: input-output pairs
- goal: find predictor $x \in \mathbb{R}^5$ such that $x^\top(w;1) \sim y$
- quadratic loss $f(x,\xi) = (x^\top(w;1) - y)^2$
- dataset: $w \sim \mathcal{N}(0, I_4)$, $y = (1,4,3,2) * w + v$, v uniformly distributed over $[-1,1]$
- each agent 30 i.i.d samples (300 network samples)





evolution of optimizer estimates



relative benefit of cooperation

Summary

Conclusions

- network optimization via primal-dual dynamics
- Lyapunov function: distance to saddle-point set + magnitude of vector field
- robustness against disturbances, real-time state-triggered implementation, time-varying, data-driven formulations

Current&Future work

- distributed regularization for strongly convex-concave formulations and impact on saddle points
- robust stability via ISS for general convex optimization
- nonconvex scenarios via sequential convex approx.
- dynamic ambiguity sets and online data-driven distributionally robust optimization
- trigger design for accelerated convergence