End-to-End Learning to Warm-Start for Real-Time Quadratic Optimization

Rajiv Sambharya (1), Georgina Hall (2), Brandon Amos (3), Bartolomeo Stellato (1)

(1) Princeton University (2) INSEAD (3) Meta

Motivation

- We often need to solve parametric quadratic programs (QPs) quickly.
- Standard algorithms are not designed to solve parametric problems.
- Can we use machine learning to accelerate parametric quadratic optimization by learning a good warm-start from data?

Robotics

Machine learning

Energy

Finance

Quadratic program

$$\begin{array}{ll} \text{minimize} & (1/2)x^TPx + c^Tx \\ \text{subject to} & Ax + s = b \\ & s \geq 0 \end{array}$$

with parameter
$\theta = (\mathbf{vec}(P), \mathbf{vec}(A), c, b)$

Contributions

- We propose a principled framework to **learn high quality warm-starts** for parametric QPs.
- We combine operator theory and Rademacher complexity theory to obtain novel **generalization bounds** for contractive operators.
- We benchmark our approach with various real-time applications.

Learning Framework

Linear complementarity problem

find u s.t. $\mathcal{C} \ni u \perp Mu + q \in \mathcal{C}^*$

$$M = \begin{bmatrix} P & A^T \\ -A & 0 \end{bmatrix}, q = (c, b), \mathcal{C} = \mathbf{R}^n \times \mathbf{R}^m_+$$

Monotone inclusion problem

find u s.t. $0 \in Mu + q + N_{\mathcal{C}}(u)$

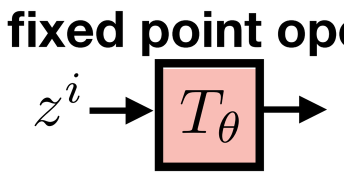
- solve with Douglas-Rachford (DR) splitting
- work with dual vector z

Algorithm 1 The DR Splitting algorithm for k iterations.

Inputs: initial point z^0, problem data (M, q), k number of iterations
Output: approximate solution z^k
For $i = 0, \dots, k-1$ **do**

$$u^{i+1} = (M + I)^{-1}(z^i - q)$$
$$\tilde{u}^{i+1} = \Pi_{\mathcal{C}}\left(2u^{i+1} - z^i\right)$$
$$z^{i+1} = z^i + \tilde{u}^{i+1} - u^{i+1}$$

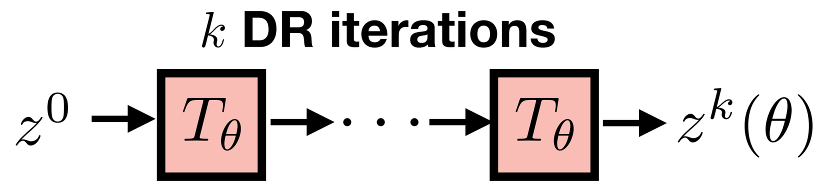
fixed point operator



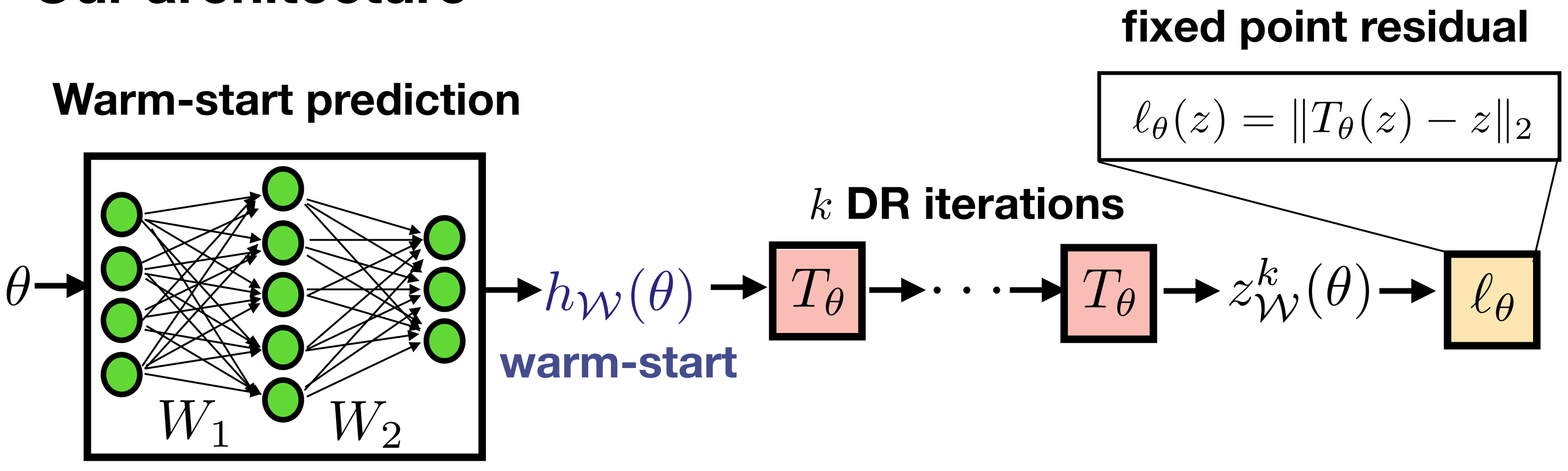
Standard DR splitting

- Starts with a random point, z^0
- We learn the warm-start instead

k DR iterations



Our architecture



- Neural network with weights $\mathcal{W} = \{W_i\}_{i=1}^L$
- $h_{\mathcal{W}}(\theta) = W_L\phi(W_{L-1}\phi(\dots\phi(W_1\theta)))$
- Let $\mathcal{H}$ be all the mappings, $h_{\mathcal{W}}$, considered

Learning task

- Training objective: to minimize the empirical risk

$$\text{minimize } \hat{R}^k(h_{\mathcal{W}}) = \frac{1}{N}\sum_{i=1}^N \ell_{\theta_i}(T^k_{\theta_i}(h_{\mathcal{W}}(\theta_i)))$$

- The ultimate goal is to reduce the generalization error and minimize the risk, $R^k(h_{\mathcal{W}}) = \mathbf{E}[\ell_\theta(T^k_\theta(h_{\mathcal{W}}(\theta)))]$.

Generalization Guarantees

Contractive operator *(Assumption)* : T_θ is β-contractive for $\beta \in (0,1)$ if

$$\|T_\theta(z) - T_\theta(w)\|_2 \leq \beta\|z - w\|_2 \quad \forall z, w \in \mathbf{dom}(T_\theta).$$

Rademacher complexity: The empirical Rademacher complexity of a function class $\mathcal{F}$ is

$$\mathbf{erad}(\mathcal{F}) = \frac{1}{N}\mathbf{E}_\sigma\left[\sup_{f \in \mathcal{F}} \sum_{i=1}^N \sigma_i f(\theta_i)\right].$$

Theorem 1. *Suppose all operators T_θ are β-contractive for $\beta \in (0,1)$. Let $\mathcal{H}$ be the set of ReLU neural networks such that for any $h_{\mathcal{W}} \in \mathcal{H}$, $\mathbf{dist}_{\mathbf{fix}\, T_\theta}(h_{\mathcal{W}}(\theta)) \leq B$ for some $B > 0$ and any $\theta \in \Theta$. Then, with probability at least $1 - \delta$ over the draw of i.i.d samples,*

$$R(h_{\mathcal{W}}) \leq \hat{R}(h_{\mathcal{W}}) + 2\sqrt{2}\beta^k\left(2\mathbf{erad}(\mathcal{H}) + B\log(1/\delta)/(2N)\right), \quad \forall h_{\mathcal{W}} \in \mathcal{H},$$

where k is the number of DR iterations and N is the number of training samples.

Corollary 1. *Let $\mathcal{H}$ be the set of linear functions with bounded norm, i.e., $\mathcal{H} = \{h \mid h(\theta) = W\theta\}$ where $\theta \in \mathbf{R}^d$, $W \in \mathbf{R}^{(m+n)\times d}$ and $(1/2)\|W\|_F^2 \leq B$ for some $B > 0$. Then, with probability at least $1 - \delta$ over the draw of i.i.d samples,*

$$R(h_{\mathcal{W}}) \leq \hat{R}(h_{\mathcal{W}}) + 2\sqrt{2}\beta^k\left(2\rho_2(\theta)\sqrt{2d/N} + B\log(1/\delta)/(2N)\right), \quad \forall h_{\mathcal{W}} \in \mathcal{H},$$

where k is the number of DR iterations, N is the number of training samples, and $\rho_2(\theta) = \max_{\theta\in\Theta}\|\theta\|_2$.

Numerical Experiments

Vehicle tracking

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^T (y_t - y_t^{\text{ref}})^T Q_t (y_t - y_t^{\text{ref}}) + u_t^T R u_t \\ \text{subject to} & x_{t+1} = A(v)x_t + B(v)u_t + E(v)\delta_t \\ & |u_t| \leq \bar{u} \\ & |u_t - u_{t-1}| \leq \overline{\Delta u} \\ & y_t = Cx_t \\ & x_0 = x_{\text{init}} \end{array}$$

$\theta = (v, x_{\text{init}}, u_{-1}, y_t^{\text{ref}}, \delta_t)$

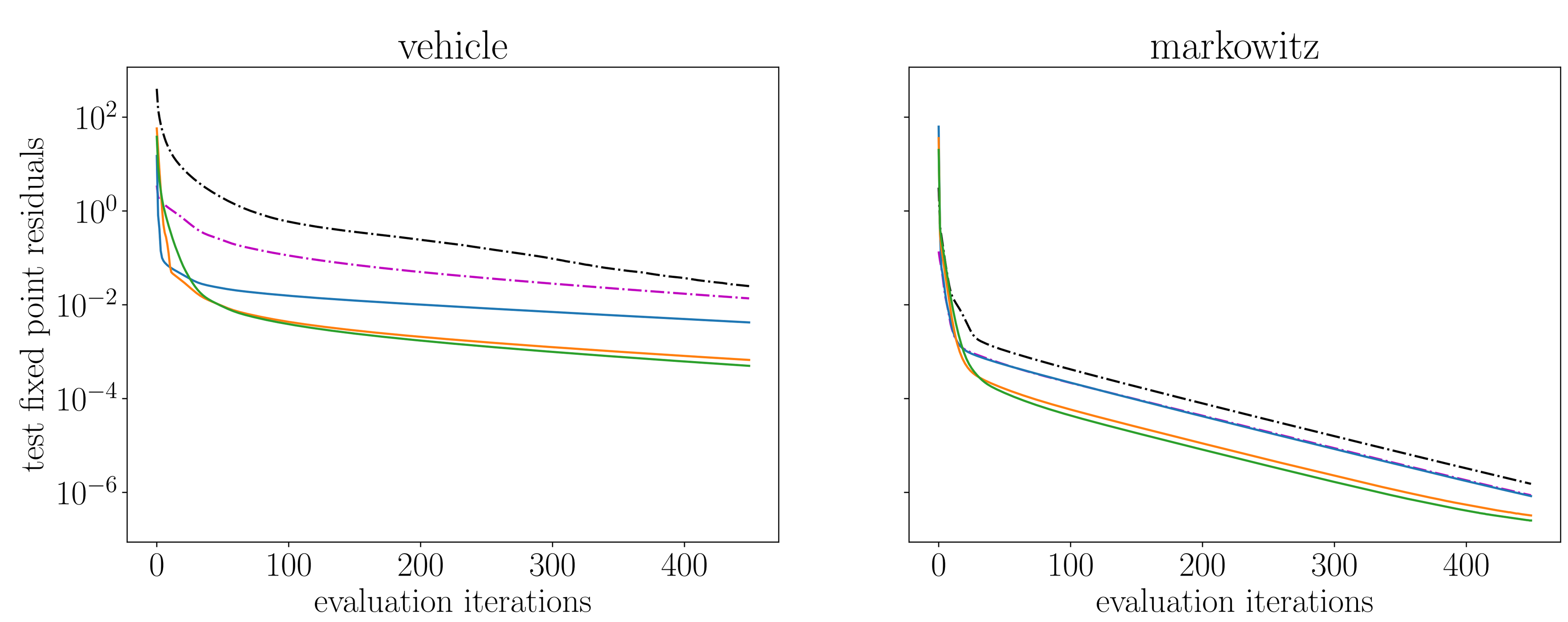
- $A(v), B(v), E(v)$: dynamics
- $\bar{u}, \overline{\Delta u}$: control limits
- C: observation matrix
- Q_t, R: costs

Markowitz portfolio

$$\begin{array}{ll} \text{maximize} & \rho\mu^T x - x^T\Sigma x \\ \text{subject to} & \mathbf{1}^T x = 1 \\ & x \geq 0 \end{array}$$

$\theta = \mu$

- μ: returns
- Σ: covariance
- ρ: hyperparameter



■ no warm-start ■ nearest neighbor warm-start
learned warm-start k = {■ 5 ■ 15 ■ 50}

- The learned warm-start reduces the number of DR iterations to reach a given accuracy by at least 30% and as much as 90%.

Conclusions

- We accelerate quadratic optimization by learning a good warm-start.
- We prove generalization bounds in the contractive case.
- We provide numerical results for a vehicle tracking problem and a portfolio optimization problem.