Learning-Assisted Data-Driven Optimization

Rohit Kannan

Center for Nonlinear Studies, Applied Mathematics & Plasma Physics
Los Alamos National Laboratory

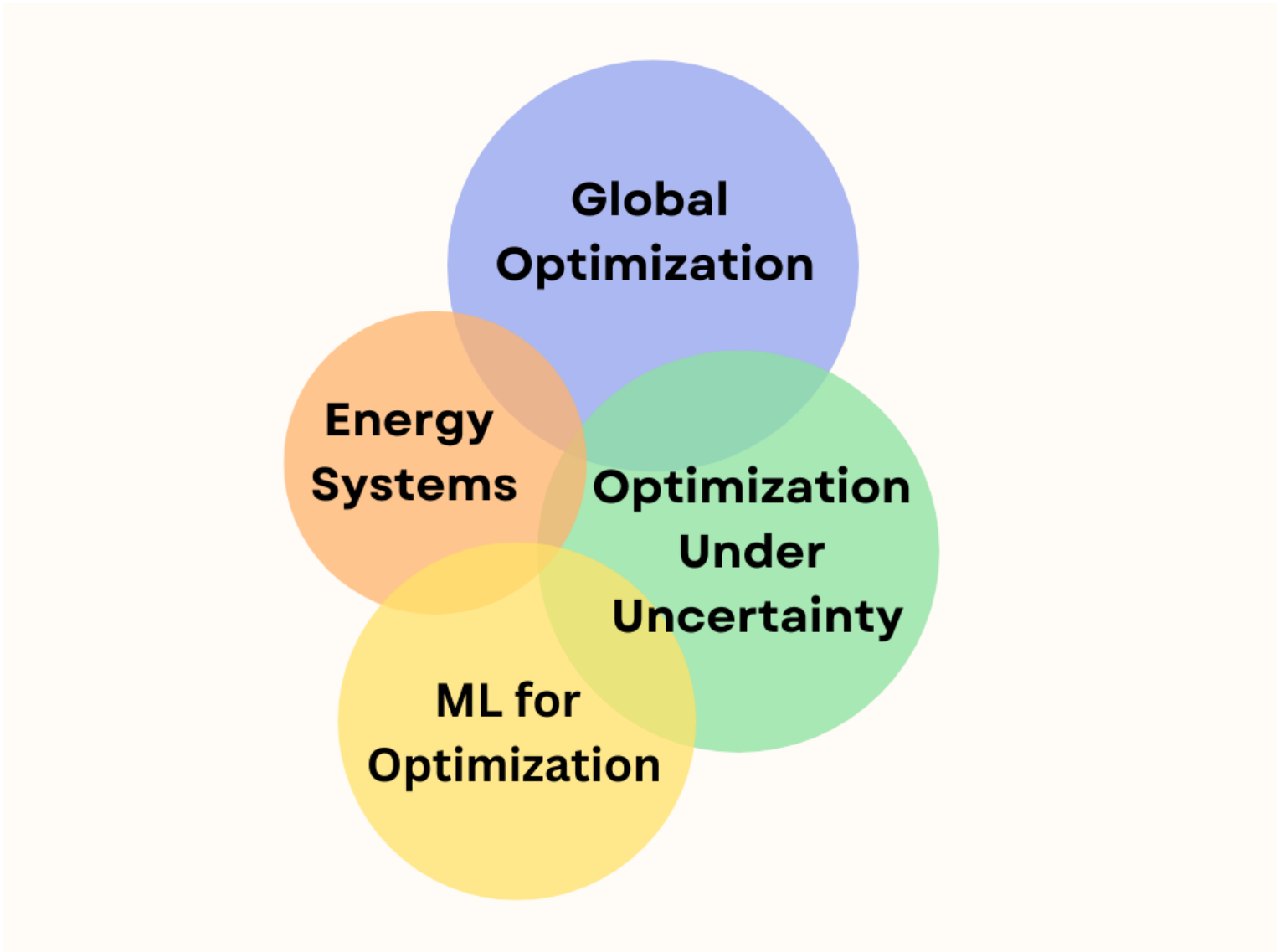
Grado Dept. of Industrial & Systems Engineering, Virginia Tech

February 6, 2023

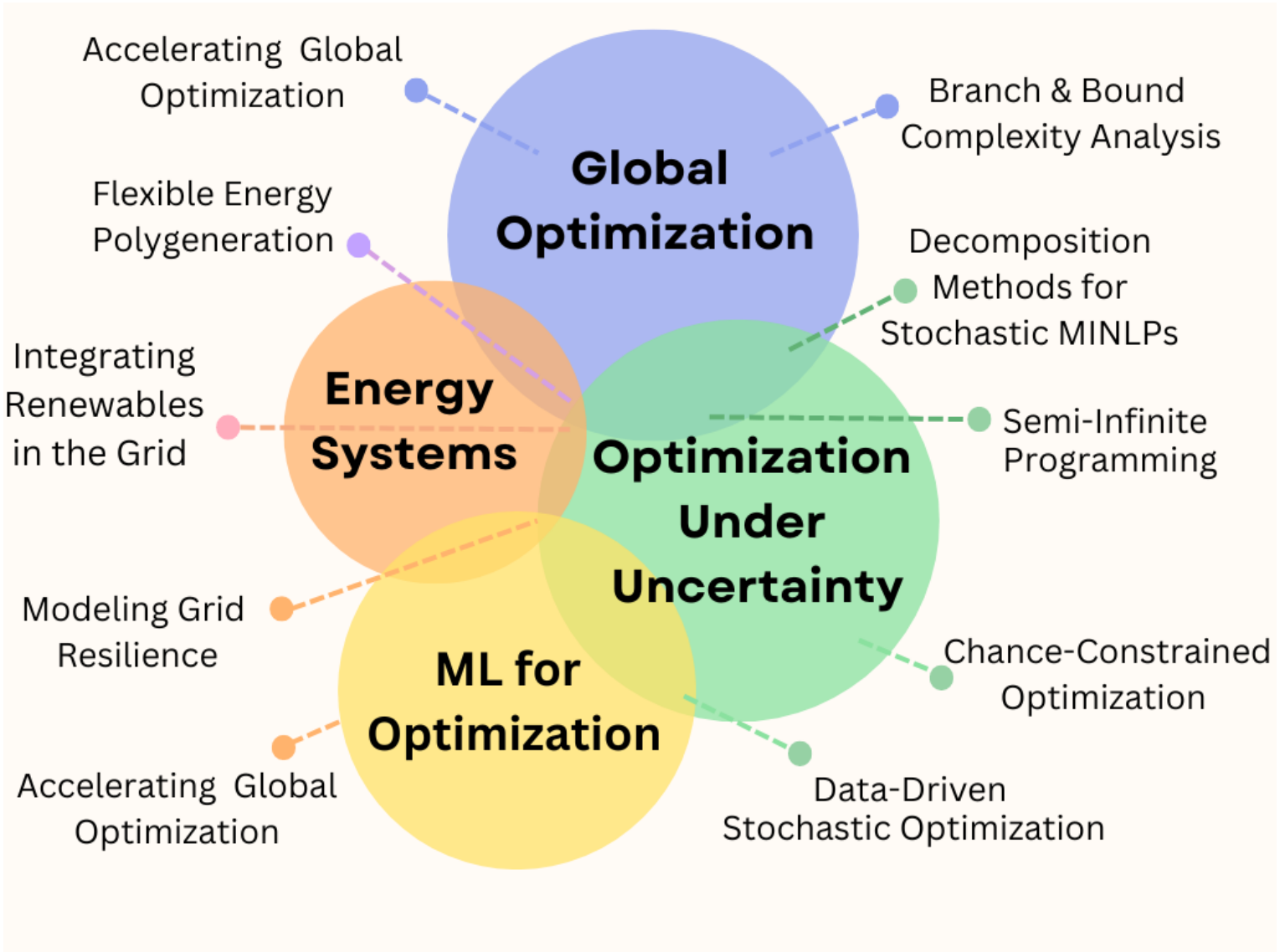
Funding: U.S. DOE, Center for Nonlinear Studies, LANL LDRD Program

Outline

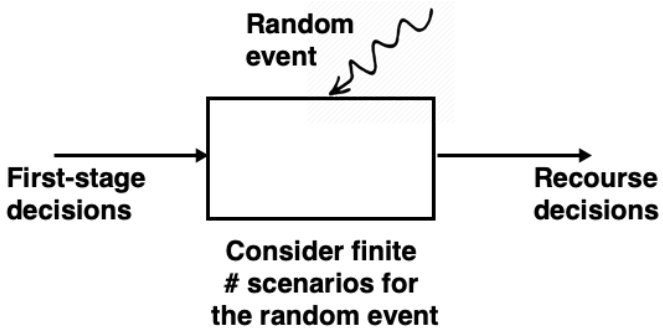
Research Overview

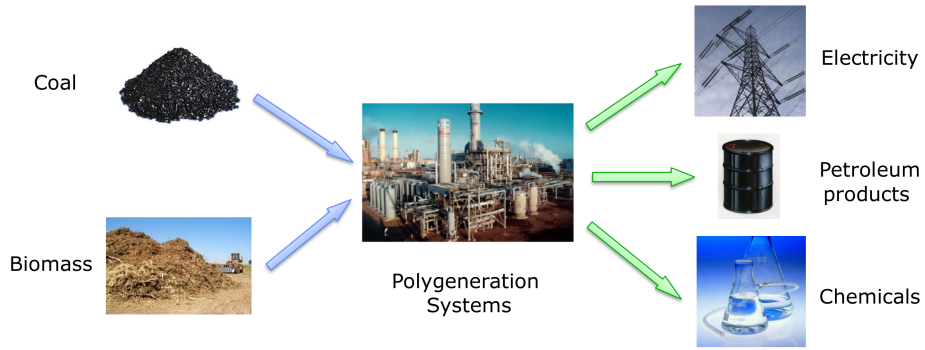


Research Overview



Global Optimization of Two-Stage Stochastic Programs



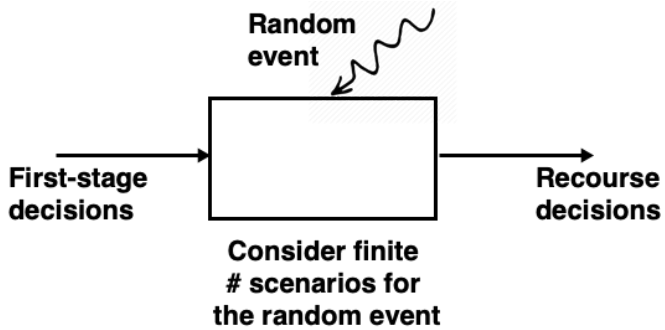


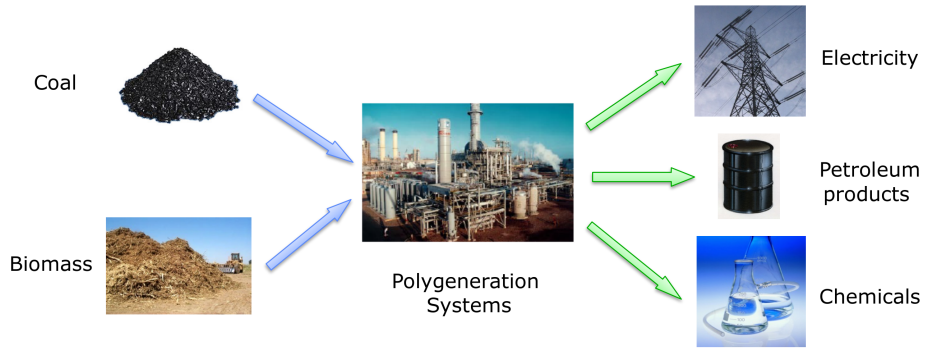
K. and Barton. Integrating Benders decomposition and Lagrangian relaxation for solving two-stage stochastic MINLPs

K. and Barton. GOSSIP: Decomposition software for the global optimization of two-stage stochastic MINLPs

Subramanian, K., et al. Optimization under uncertainty of a hybrid waste tire & natural gas flexible polygeneration system

Global Optimization of Two-Stage Stochastic Programs





- Complexity of generic B&B grows exponentially with number of scenarios
- Designed first fully-decomposable algorithm with provable convergence

Paul Barton
(MIT CHE)

Avinash Subramanian
(SINTEF)

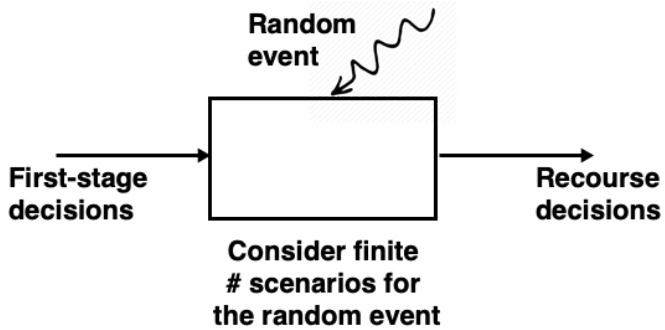
Truls Gundersen
(NTNU Energy)

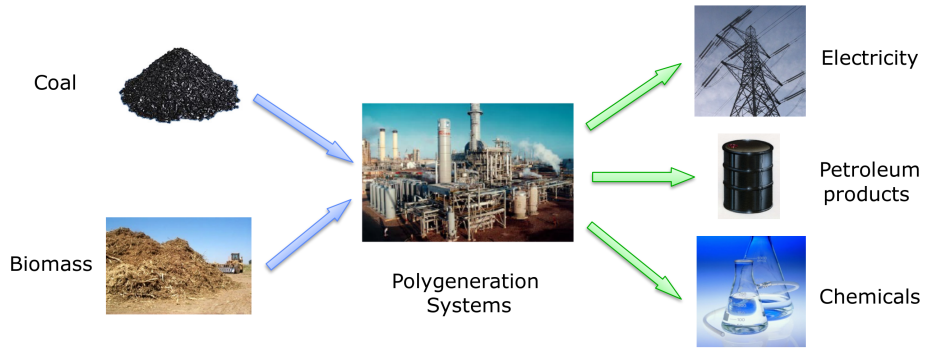
K. and Barton. Integrating Benders decomposition and Lagrangian relaxation for solving two-stage stochastic MINLPs

K. and Barton. GOSSIP: Decomposition software for the global optimization of two-stage stochastic MINLPs

Subramanian, K., et al. Optimization under uncertainty of a hybrid waste tire & natural gas flexible polygeneration system

Global Optimization of Two-Stage Stochastic Programs



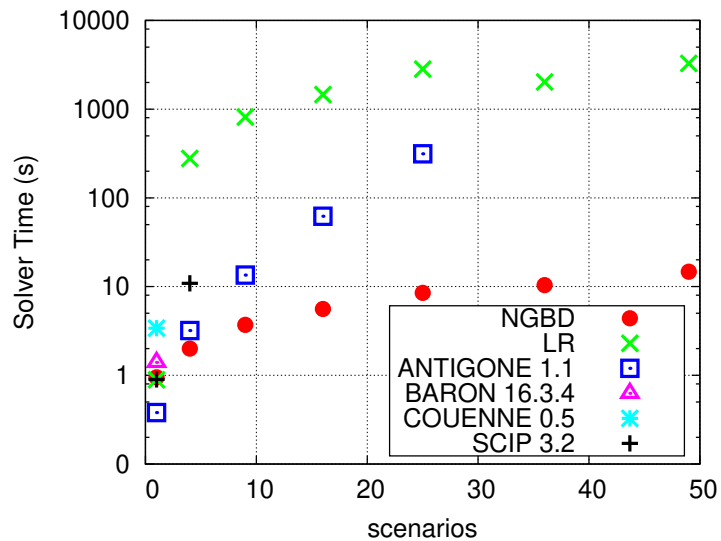


- Complexity of generic B&B grows exponentially with number of scenarios
- Designed first fully-decomposable algorithm with provable convergence

Paul Barton
(MIT CHE)

Avinash Subramanian
(SINTEF)

Truls Gundersen
(NTNU Energy)



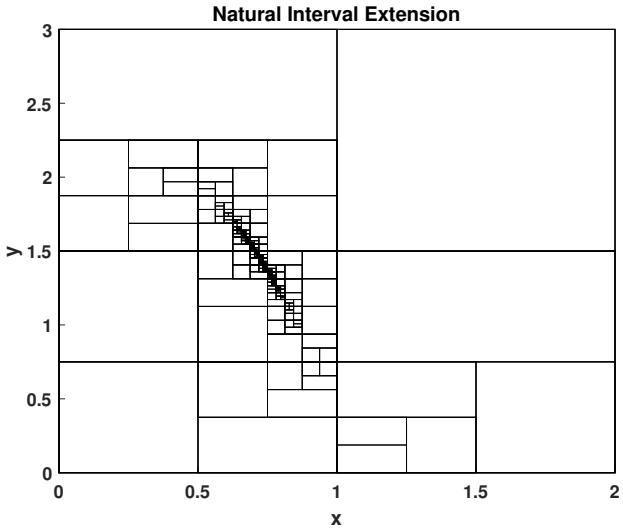
NGBD & LR: decomposition methods
Rest: State-of-the-art solvers

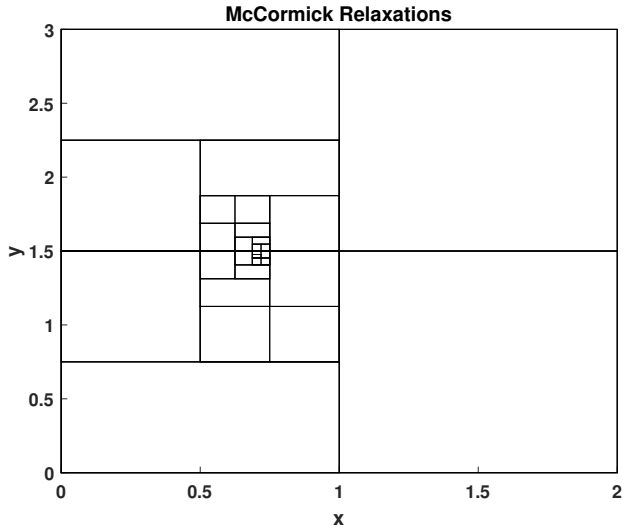
K. and Barton. Integrating Benders decomposition and Lagrangian relaxation for solving two-stage stochastic MINLPs

K. and Barton. GOSSIP: Decomposition software for the global optimization of two-stage stochastic MINLPs

Subramanian, K., et al. Optimization under uncertainty of a hybrid waste tire & natural gas flexible polygeneration system

Analysis of the Complexity of B&B Algorithms



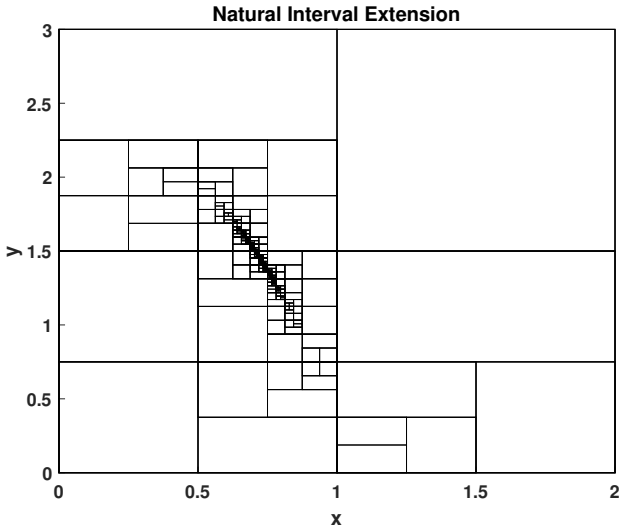


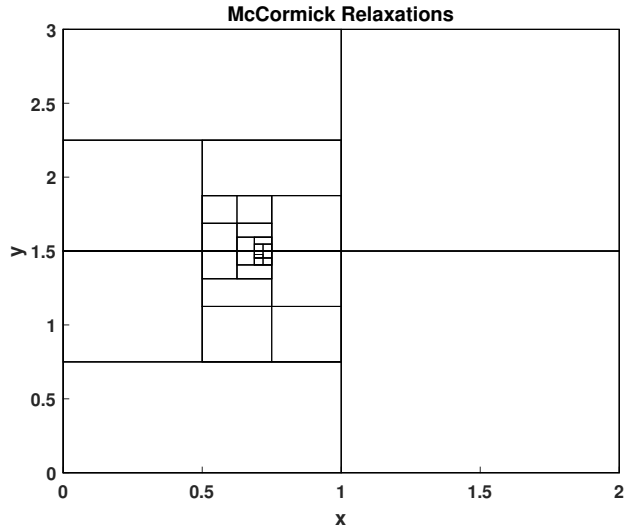
- B&B bounding methods may suffer from the "cluster problem"
- Built theory to understand which bounding methods can avoid this
  - Important implications for design of reduced-space B&B algorithms

K. and Barton (2018). The cluster problem in constrained global optimization. J. Global Optim.

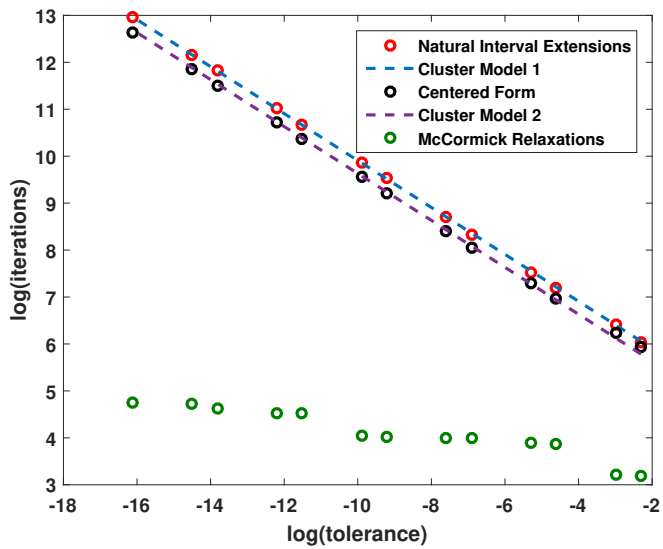
K. and Barton (2018). Convergence-order analysis of B&B algorithms for constrained problems. J. Global Optim.

Analysis of the Complexity of B&B Algorithms





- B&B bounding methods may suffer from the "cluster problem"
- Built theory to understand which bounding methods can avoid this
  - Important implications for design of reduced-space B&B algorithms



K. and Barton (2018). The cluster problem in constrained global optimization. J. Global Optim.

K. and Barton (2018). Convergence-order analysis of B&B algorithms for constrained problems. J. Global Optim.

Stochastic Approximation for Chance Constraints

$$\nu_\alpha^* := \min_{x \in X} \; f(x)$$
$$\text{s.t. } \mathbb{P}\{g(x,\xi) \leq 0\} \geq 1 - \alpha$$

Jim Luedtke
(UW-Madison ISyE)

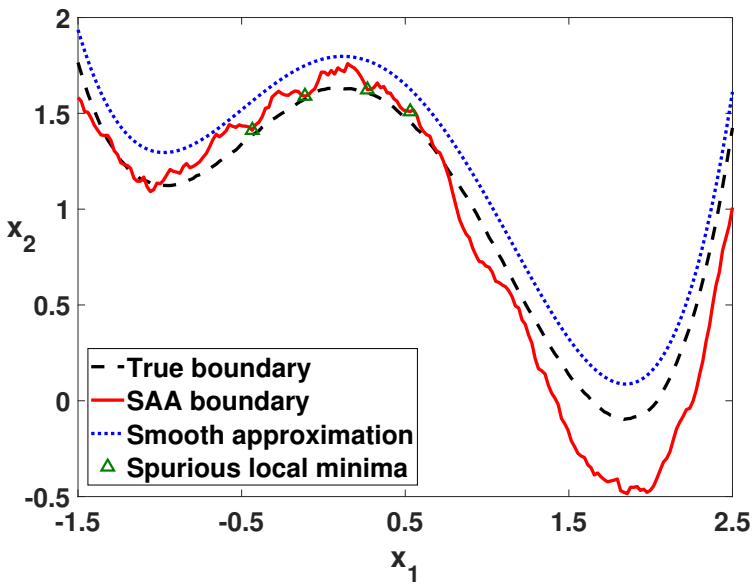
- Previous approaches are either suboptimal, or do not scale

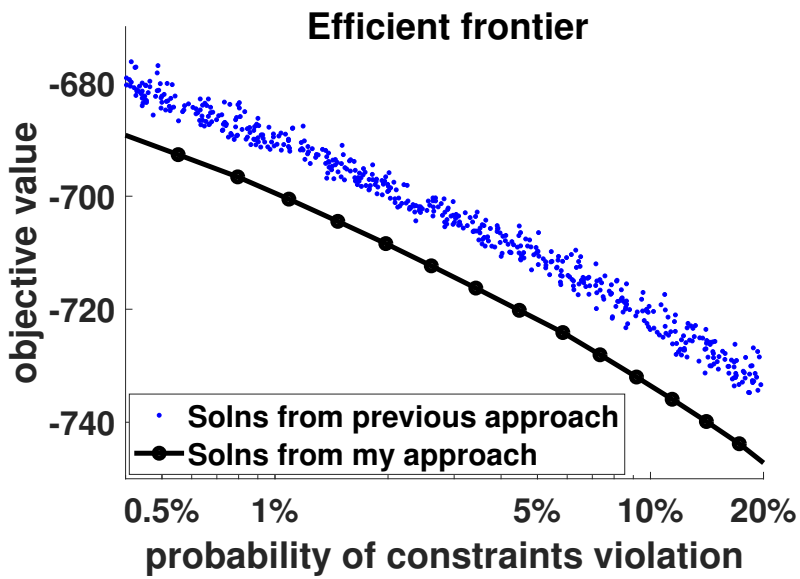
Stochastic Approximation for Chance Constraints

$$\nu_\alpha^* := \min_{x \in X} \; f(x)$$
$$\text{s.t. } \mathbb{P}\{g(x,\xi) \leq 0\} \geq 1 - \alpha$$

Jim Luedtke
(UW-Madison ISyE)

- Previous approaches are either suboptimal, or do not scale
- Designed a stochastic subgradient method for approximating the efficient frontier of cost versus risk (ν_α^* vs α)

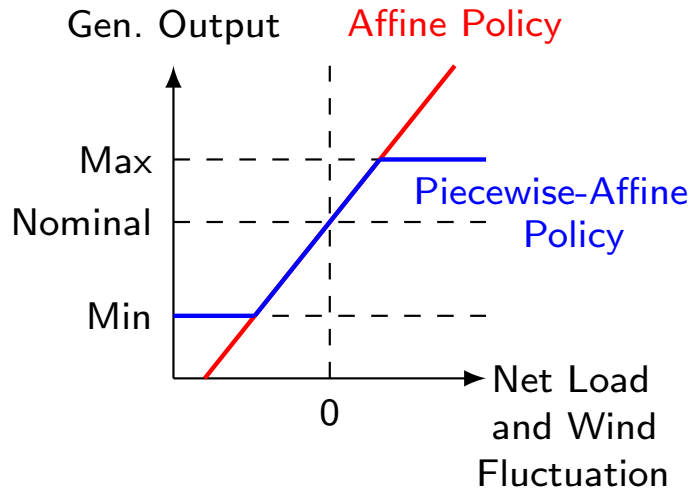




K. and Luedtke (2021). A stochastic approximation method for chance-constrained NLPs. Math. Prog. Comput.

Better Integration of Renewables in the Power Grid

- Generators balance renewables variability by activating reserves via piecewise-affine policy
  - Less conservative than forcing affine policy to be feasible with high probability

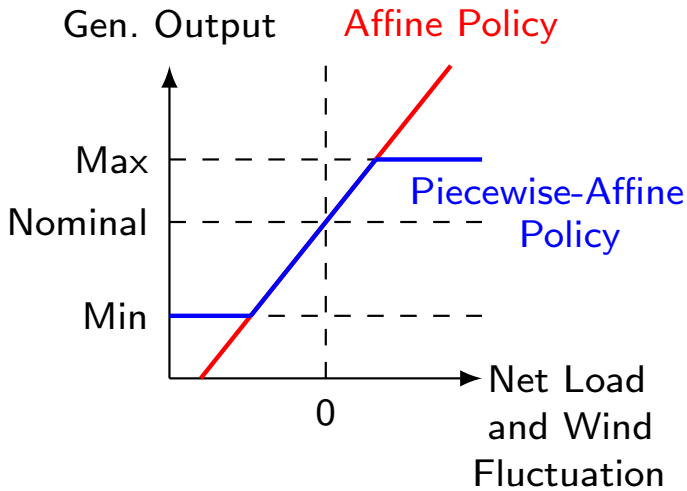


Line Roald
(UW-Madison ECE)

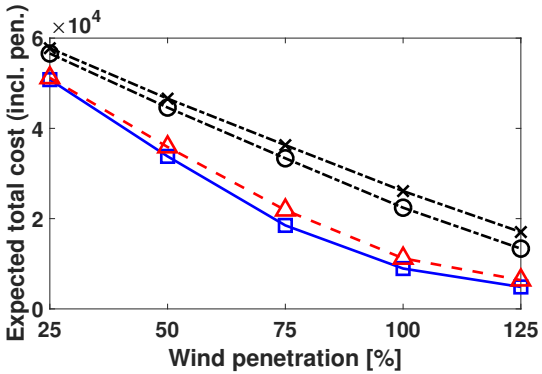
K., Luedtke, and Roald (2020). Stochastic DC-OPF with reserve saturation. Electric Power Systems Research

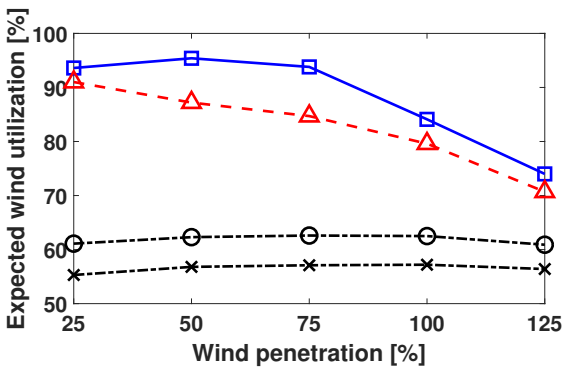
Better Integration of Renewables in the Power Grid

- Generators balance renewables variability by activating reserves via piecewise-affine policy
  - Less conservative than forcing affine policy to be feasible with high probability
- Tailored decomposition method for DC-OPF. Our approach yields solutions with



Lower total cost and Higher wind utilization





Line Roald
(UW-Madison ECE)

□: our approach. △: generator penalty. o and ×: chance constraints

K., Luedtke, and Roald (2020). Stochastic DC-OPF with reserve saturation. Electric Power Systems Research

Outline

Optimization Under Uncertainty

General optimization model with uncertain parameters Y:

$$\min_{z \in \mathcal{Z}} c(z, Y)$$

- $\mathcal{Z}$ is the feasible region (assume known) for decisions z
- Y is a vector of uncertain parameters $\Rightarrow$ ill-posed problem

Optimization Under Uncertainty

General optimization model with uncertain parameters Y:

$$\min_{z \in \mathcal{Z}} c(z, Y)$$

- $\mathcal{Z}$ is the feasible region (assume known) for decisions z
- Y is a vector of uncertain parameters $\Rightarrow$ ill-posed problem

Popular modeling approaches:

1. Stochastic: assuming distribution of Y known, minimize expected/average system cost

$$\min_{z \in \mathcal{Z}} \mathbb{E}_Y[c(z, Y)]$$

2. Robust: assuming support of Y known, minimize worst-case system cost

$$\min_{z \in \mathcal{Z}} \max_{y \in \mathcal{Y}} c(z, y)$$

Traditional Data-Driven Stochastic Programming

- Traditional SP: minimize expected system cost assuming feasible region $\mathcal{Z}$ and distribution of Y known

$$\min_{z \in \mathcal{Z}} \mathbb{E}_Y[c(z, Y)]$$

Traditional Data-Driven Stochastic Programming

- Traditional SP: minimize expected system cost assuming feasible region $\mathcal{Z}$ and distribution of Y known

$$\min_{z \in \mathcal{Z}} \mathbb{E}_Y[c(z, Y)]$$

- Data-driven SP: have access to samples $\{y^i\}_{i=1}^n$ of Y

$$\min_{z \in \mathcal{Z}} \mathbb{E}_Y[c(z, Y)] \approx \min_{z \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(z, y^i) \qquad \text{(SAA)}$$

- Sample Average Approximation theory: as sample size $n \to \infty$, optimal value and solutions converge at the rate $O_p(n^{-1/2})$

Traditional Data-Driven Stochastic Programming

- Traditional SP: minimize expected system cost assuming feasible region $\mathcal{Z}$ and distribution of Y known

$$\min_{z \in \mathcal{Z}} \mathbb{E}_Y[c(z, Y)]$$

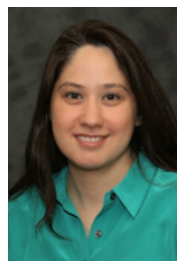
- Data-driven SP: have access to samples $\{y^i\}_{i=1}^n$ of Y

$$\min_{z \in \mathcal{Z}} \mathbb{E}_Y[c(z, Y)] \approx \min_{z \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(z, y^i) \qquad \text{(SAA)}$$

- Sample Average Approximation theory: as sample size $n \to \infty$, optimal value and solutions converge at the rate $O_p(n^{-1/2})$

How can we use covariates X to better predict the random vector Y?

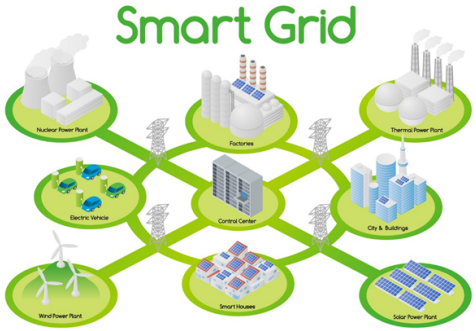
Jim Luedtke
(UW-Madison ISyE)



Güzin Bayraksan
(OSU ISE)

Nam Ho-Nguyen
(USYD Business)

Stochastic Programming with Covariate Information



Power Grid Scheduling

Y: Load; Renewable energy outputs

X: Weather observations; Time/Season

z: Generator scheduling decisions

Production Planning/Scheduling

Y: Product demands; Prices

X: Seasonality; Web search results

z: Production and inventory decisions

Portfolio Optimization

Y: Stock returns

X: Historical returns; Economic indicators

z: Investment decisions

Stochastic Programming with Covariate Information

- Assume we have uncertain parameter and covariate data pairs

$$\mathcal{D}_n := \{(y^i, x^i)\}_{i=1}^n$$

- When making decision z, we observe a *new* covariate $X = x$
- Goal: minimize expected cost given covariate observation x:

$$\min_{z \in \mathcal{Z}} \mathbb{E}\left[c(z, Y) \mid X = x\right]$$

Stochastic Programming with Covariate Information

- Assume we have uncertain parameter and covariate data pairs

$$\mathcal{D}_n := \{(y^i, x^i)\}_{i=1}^n$$

- When making decision z, we observe a *new* covariate $X = x$
- Goal: minimize expected cost given covariate observation x:

$$\min_{z \in \mathcal{Z}} \mathbb{E}\left[c(z, Y) \mid X = x\right]$$

- Challenge: $\mathcal{D}_n$ may not include covariate observation $X = x$
- How to construct data-driven approximation to conditional SP?

Stochastic Programming with Covariate Information

- Assume we have uncertain parameter and covariate data pairs

$$\mathcal{D}_n := \{(y^i, x^i)\}_{i=1}^n$$

- When making decision z, we observe a *new* covariate $X = x$
- Goal: minimize expected cost given covariate observation x:

$$\min_{z \in \mathcal{Z}} \mathbb{E}\left[c(z, Y) \mid X = x\right]$$

- Challenge: $\mathcal{D}_n$ may not include covariate observation $X = x$
- How to construct data-driven approximation to conditional SP?
  1. Learn: predict Y given $X = x$
  2. Optimize: integrate learning into optimization (with errors)

Stochastic Programming with Covariate Information

- Assume we have uncertain parameter and covariate data pairs

$$\mathcal{D}_n := \{(y^i, x^i)\}_{i=1}^n$$

- When making decision z, we observe a *new* covariate $X = x$
- Goal: minimize expected cost given covariate observation x:

$$\min_{z \in \mathcal{Z}} \mathbb{E}\left[c(z, Y) \mid X = x\right]$$

- Challenge: $\mathcal{D}_n$ may not include covariate observation $X = x$
- How to construct data-driven approximation to conditional SP?
  1. Learn: predict Y given $X = x$
  2. Optimize: integrate learning into optimization (with errors)
- Assume $Y = f^*(X) + Q^*(X)\varepsilon$ with X and ε *independent*

Traditional Integrated Learning and Optimization

1. Use data to train your favorite ML prediction model:

$$\hat{f}_n(\cdot) \in \arg\min_{f(\cdot)\in\mathcal{F}} \sum_{i=1}^{n} \ell(f(x^i), y^i) + \rho(f)$$

2. Given observed covariate $X = x$, use point prediction within deterministic optimization model

$$\min_{z\in\mathcal{Z}} c(z, \hat{f}_n(x))$$

Traditional Integrated Learning and Optimization

1. Use data to train your favorite ML prediction model:

$$\hat{f}_n(\cdot) \in \arg\min_{f(\cdot)\in\mathcal{F}} \sum_{i=1}^{n} \ell(f(x^i), y^i) + \rho(f)$$

2. Given observed covariate $X = x$, use point prediction within deterministic optimization model

$$\min_{z\in\mathcal{Z}} c(z, \hat{f}_n(x))$$

- Modular: separate learning and optimization steps
- Expect to work well only if prediction is highly accurate

Traditional Integrated Learning and Optimization

1. Use data to train your favorite ML prediction model:

$$\hat{f}_n(\cdot) \in \arg\min_{f(\cdot)\in\mathcal{F}} \sum_{i=1}^{n} \ell(f(x^i), y^i) + \rho(f)$$

2. Given observed covariate $X = x$, use point prediction within deterministic optimization model

$$\min_{z\in\mathcal{Z}} c(z, \hat{f}_n(x))$$

- Modular: separate learning and optimization steps
- Expect to work well only if prediction is highly accurate
- Many recently proposed improvements in the literature, e.g., Ban and Rudin (2019); Bertsimas and Kallus (2020); Deng and Sen (2022); Donti et al. (2017); Elmachtoub and Grigas (2022)

Empirical Residuals-based Sample Average Approximation

1. Estimate f^*, Q^* using your favorite ML method $\Rightarrow \hat{f}_n, \hat{Q}_n$

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

K., Ho-Nguyen, and Luedtke. Data-driven multistage stochastic optimization on time series. Working Paper

Empirical Residuals-based Sample Average Approximation

1. Estimate f^*, Q^* using your favorite ML method $\Rightarrow \hat{f}_n, \hat{Q}_n$

   Compute *empirical residuals* $\hat{\varepsilon}_n^i := [\hat{Q}_n(x^i)]^{-1}\left(y^i - \hat{f}_n(x^i)\right),\ i \in [n]$

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

K., Ho-Nguyen, and Luedtke. Data-driven multistage stochastic optimization on time series. Working Paper

Empirical Residuals-based Sample Average Approximation

1. Estimate f^*, Q^* using your favorite ML method $\Rightarrow \hat{f}_n, \hat{Q}_n$

   Compute *empirical residuals* $\hat{\varepsilon}^i_n := [\hat{Q}_n(x^i)]^{-1}\big(y^i - \hat{f}_n(x^i)\big)$, $i \in [n]$

2. Use $\{\hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}^i_n\}_{i=1}^n$ as proxy for samples of Y given $X = x$

$$\min_{z \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(z, \hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}^i_n) \qquad \text{(ER-SAA)}$$

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

K., Ho-Nguyen, and Luedtke. Data-driven multistage stochastic optimization on time series. Working Paper

Empirical Residuals-based Sample Average Approximation

1. Estimate f^*, Q^* using your favorite ML method $\Rightarrow \hat{f}_n, \hat{Q}_n$

   Compute *empirical residuals* $\hat{\varepsilon}_n^i := [\hat{Q}_n(x^i)]^{-1}\big(y^i - \hat{f}_n(x^i)\big)$, $i \in [n]$

2. Use $\{\hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}_n^i\}_{i=1}^n$ as proxy for samples of Y given $X = x$

$$\min_{z \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(z, \hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}_n^i) \qquad \text{(ER-SAA)}$$

   - Modular like traditional approach

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

K., Ho-Nguyen, and Luedtke. Data-driven multistage stochastic optimization on time series. Working Paper

Empirical Residuals-based Sample Average Approximation

1. Estimate f^*, Q^* using your favorite ML method $\Rightarrow \hat{f}_n, \hat{Q}_n$

   Compute *empirical residuals* $\hat{\varepsilon}_n^i := [\hat{Q}_n(x^i)]^{-1}\big(y^i - \hat{f}_n(x^i)\big),\ i \in [n]$

2. Use $\{\hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}_n^i\}_{i=1}^n$ as proxy for samples of Y given $X = x$

$$\min_{z \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(z, \hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}_n^i) \qquad \text{(ER-SAA)}$$

- Modular like traditional approach

Contributions:

- General convergence analysis
- Improvements when sample size is small
- Extension to dynamic/sequential decision-making

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

K., Ho-Nguyen, and Luedtke. Data-driven multistage stochastic optimization on time series. Working Paper

New Small Sample Variant of ER-SAA

Mitigate effects of overfitting by using *leave-one-out residuals*

1. Estimate f^*, Q^* separately with each data point i left out (leave-one-out regression) $\Rightarrow \hat{f}_{-i}(\cdot), \hat{Q}_{-i}(\cdot)$ for $i \in [n]$

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

New Small Sample Variant of ER-SAA

Mitigate effects of overfitting by using *leave-one-out residuals*

1. Estimate f^*, Q^* separately with each data point i left out (leave-one-out regression) $\Rightarrow \hat{f}_{-i}(\cdot), \hat{Q}_{-i}(\cdot)$ for $i \in [n]$

Compute *leave-one-out residuals* $\hat{\varepsilon}^i_n := [\hat{Q}_{-i}(x^i)]^{-1}\big(y^i - \hat{f}_{-i}(x^i)\big),\ i \in [n]$

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

New Small Sample Variant of ER-SAA

Mitigate effects of overfitting by using *leave-one-out residuals*

1. Estimate f^*, Q^* separately with each data point i left out (leave-one-out regression) $\Rightarrow \hat{f}_{-i}(\cdot), \hat{Q}_{-i}(\cdot)$ for $i \in [n]$

Compute *leave-one-out residuals* $\hat{\varepsilon}^i_n := [\hat{Q}_{-i}(x^i)]^{-1}\big(y^i - \hat{f}_{-i}(x^i)\big)$, $i \in [n]$

2. Use $\{\hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}^i_n\}_{i=1}^n$ or $\{\hat{f}_{-i}(x) + \hat{Q}_{-i}(x)\hat{\varepsilon}^i_n\}_{i=1}^n$ as proxy for samples of Y given $X = x$

$$\min_{z \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(z, \hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}^i_n) \qquad \text{(J-SAA)}$$

Inspired by Jackknife methods (Barber et al., 2021)

K., Bayraksan, and Luedtke. Data-driven SAA with covariate information. arXiv:2207.13554. Under Revision

Distributionally robust optimization (ER-DRO)

- Minimize worst-case expected cost over a set of distributions

$$\hat{z}_n^{DRO}(x) \in \arg\min_{z \in \mathcal{Z}} \max_{Q \in \hat{\mathcal{P}}_n(x)} \mathbb{E}_{Y \sim Q}[c(z, Y)]$$

$$\hat{\mathcal{P}}_n(x) = \text{“confidence region” for distribution of } Y \text{ given } X = x$$

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

Distributionally robust optimization (ER-DRO)

- Minimize worst-case expected cost over a set of distributions

$$\hat{z}_n^{DRO}(x) \in \arg\min_{z \in \mathcal{Z}} \max_{Q \in \hat{\mathcal{P}}_n(x)} \mathbb{E}_{Y \sim Q}[c(z, Y)]$$

$\hat{\mathcal{P}}_n(x) =$ "confidence region" for distribution of Y given $X = x$

- $\hat{\mathcal{P}}_n(x) := \left\{ \frac{1}{n} \sum_{i=1}^{n} \delta_{\hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}_n^i} \right\} \implies$ ER-SAA

- Motivation: DRO regularizes small sample ER-SAA, yielding solutions with better out-of-sample performance

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

Distributionally robust optimization (ER-DRO)

- Minimize worst-case expected cost over a set of distributions

$$\hat{z}_n^{DRO}(x) \in \arg\min_{z \in \mathcal{Z}} \max_{Q \in \hat{\mathcal{P}}_n(x)} \mathbb{E}_{Y \sim Q}[c(z, Y)]$$

$\hat{\mathcal{P}}_n(x) =$ "confidence region" for distribution of Y given $X = x$

- $\hat{\mathcal{P}}_n(x) := \left\{ \frac{1}{n} \sum_{i=1}^{n} \delta_{\hat{f}_n(x) + \hat{Q}_n(x)\hat{\varepsilon}_n^i} \right\} \implies$ ER-SAA
- Motivation: DRO regularizes small sample ER-SAA, yielding solutions with better out-of-sample performance
- Example: Wasserstein ambiguity sets of order $p \in [1, +\infty)$:

$$\hat{\mathcal{P}}_n(x) := \{\text{distributions } Q \text{ such that the } p\text{-Wasserstein distance between } Q \text{ and } \hat{P}_n^{ER}(x) \leq \zeta_n(x)\}$$

K., Bayraksan, and Luedtke. Residuals-based DRO with covariate information. arXiv:2012.01088. Under Review

Toward Convergence Theory: Definitions

Recall

- $v^*(x) = \min_{z \in \mathcal{Z}} \mathbb{E}_\varepsilon[c(z, f^*(x) + Q^*(x)\varepsilon)]$
  $= \text{optimal value of true conditional SP}$
- $\hat{z}_n^{ER}(x) = \text{ER-SAA solution}$

Asymptotic optimality: the out-of-sample cost of data-driven solutions approaches the optimal value of the true conditional SP as the sample size increases

$$\mathbb{E}_\varepsilon\left[c(\hat{z}_n^{ER}(x), f^*(x) + Q^*(x)\varepsilon)\right] \xrightarrow{p} v^*(x)$$

Toward Convergence Theory: Definitions

Recall

- $v^*(x) = \min_{z \in \mathcal{Z}} \mathbb{E}_\varepsilon[c(z, f^*(x) + Q^*(x)\varepsilon)]$
  $= \text{optimal value of true conditional SP}$
- $\hat{z}_n^{ER}(x) = \text{ER-SAA solution}$

Asymptotic optimality: the out-of-sample cost of data-driven solutions approaches the optimal value of the true conditional SP as the sample size increases

$$\mathbb{E}_\varepsilon\left[c(\hat{z}_n^{ER}(x), f^*(x) + Q^*(x)\varepsilon)\right] \xrightarrow{p} v^*(x)$$

Setting: two-stage stochastic mixed-integer linear programs with continuous recourse and r.h.s. uncertainty

From hereon, assume for simplicity that $Q^* \equiv I$

Asymptotic Optimality of ER-SAA Solutions

Asymptotic Optimality of ER-SAA Solutions

Assumption: The regression procedure satisfies

- Pointwise error consistency: $\hat{f}_n(x) \xrightarrow{p} f^*(x)$ for a.e. x
- Mean-squared estimation error consistency:

$$\frac{1}{n}\sum_{i=1}^{n}\|f^*(x^i) - \hat{f}_n(x^i)\|^2 \xrightarrow{p} 0.$$

Asymptotic Optimality of ER-SAA Solutions

Assumption: The regression procedure satisfies

- Pointwise error consistency: $\hat{f}_n(x) \xrightarrow{p} f^*(x)$ for a.e. x
- Mean-squared estimation error consistency:

$$\frac{1}{n}\sum_{i=1}^{n}\|f^*(x^i) - \hat{f}_n(x^i)\|^2 \xrightarrow{p} 0.$$

Informal Theorem (Asymptotic Optimality)

Under the above assumptions†, the ER-SAA solution $\hat{z}_n^{ER}(x)$ is asymptotically optimal for a.e. x, i.e.,

$$\mathbb{E}_\varepsilon\left[c(\hat{z}_n^{ER}(x), f^*(x) + \varepsilon)\right] \xrightarrow{p} v^*(x)$$

†Plus some mild standard assumptions on the true conditional SP, see arXiv:2207.13554

Finite-Sample Guarantees for ER-SAA Solutions

Estimate sample size n required for optimal solutions of ER-SAA to be κ-optimal to the true conditional SP with probability $\geq 1 - \delta$

Finite-Sample Guarantees for ER-SAA Solutions

Estimate sample size n required for optimal solutions of ER-SAA to be κ-optimal to the true conditional SP with probability $\geq 1 - \delta$

- If f^* is linear and we use OLS regression, then require
- If f^* is s-sparse linear and we use the Lasso, then require
- If f^* is Lipschitz and we use kNN regression, then require

Finite-Sample Guarantees for ER-SAA Solutions

Estimate sample size n required for optimal solutions of ER-SAA to be κ-optimal to the true conditional SP with probability $\geq 1 - \delta$

- If f^* is linear and we use OLS regression, then require

$$n \geq \frac{O(1)}{\kappa^2}\left[d_z \log\left(\frac{O(1)}{\kappa}\right) + d_y \log\left(\frac{O(1)}{\delta}\right) + d_x d_y\right]$$

- If f^* is s-sparse linear and we use the Lasso, then require

$$n \geq \frac{O(1)}{\kappa^2}\left[d_z \log\left(\frac{O(1)}{\kappa}\right) + s d_y \log\left(\frac{O(1)}{\delta}\right) + s \log(d_x) d_y\right]$$

- If f^* is Lipschitz and we use kNN regression, then require

$$n \geq \frac{O(1)d_z}{\kappa^2} \log\left(\frac{O(1)}{\kappa}\right) + \left(\frac{O(1)d_y}{\kappa^2}\right)^{d_x}\left[d_x \log\left(\frac{O(1)d_x d_y}{\kappa^2}\right) + \log\left(\frac{O(1)}{\delta}\right)\right]$$

Choosing the Ambiguity Set Radius for Wasserstein DRO

Choosing the Ambiguity Set Radius for Wasserstein DRO

Assumption: For any risk level $\alpha \in (0, 1)$, there exists a constant $\kappa_{p,n}(\alpha, x) > 0$ such that the regression procedure satisfies

$$\mathbb{P}\left\{\|f^*(x) - \hat{f}_n(x)\|^p > \kappa^p_{p,n}(\alpha, x)\right\} \leq \alpha, \quad \text{and}$$

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}\|f^*(x^i) - \hat{f}_n(x^i)\|^p > \kappa^p_{p,n}(\alpha, x)\right\} \leq \alpha.$$

Choosing the Ambiguity Set Radius for Wasserstein DRO

Assumption: For any risk level $\alpha \in (0,1)$, there exists a constant $\kappa_{p,n}(\alpha, x) > 0$ such that the regression procedure satisfies

$$\mathbb{P}\{\|f^*(x) - \hat{f}_n(x)\|^p > \kappa^p_{p,n}(\alpha, x)\} \leq \alpha, \quad \text{and}$$

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}\|f^*(x^i) - \hat{f}_n(x^i)\|^p > \kappa^p_{p,n}(\alpha, x)\right\} \leq \alpha.$$

Example: Finite-sample guarantee on regression step holds for $p = 2$ and

- OLS, Lasso with $\kappa^2_{2,n}(\alpha, x) = O(n^{-1}\log(\alpha^{-1}))$
- CART, RF with $\kappa^2_{2,n}(\alpha, x) = O(n^{-1}\log(\alpha^{-1}))^{O(1)/d_x}$

Choosing the Ambiguity Set Radius for Wasserstein DRO

Assumption: For any risk level $\alpha \in (0,1)$, there exists a constant $\kappa_{p,n}(\alpha, x) > 0$ such that the regression procedure satisfies

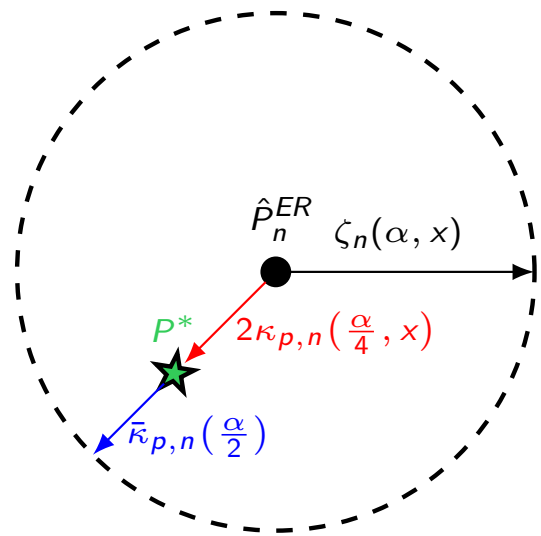
$$\mathbb{P}\{\|f^*(x) - \hat{f}_n(x)\|^p > \kappa^p_{p,n}(\alpha, x)\} \leq \alpha, \quad \text{and}$$

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}\|f^*(x^i) - \hat{f}_n(x^i)\|^p > \kappa^p_{p,n}(\alpha, x)\right\} \leq \alpha.$$

Given covariate realization x and risk level $\alpha \in (0,1)$, use radius

$$\zeta_n(\alpha, x) := 2\kappa_{p,n}\left(\tfrac{\alpha}{4}, x\right) + \bar{\kappa}_{p,n}\left(\tfrac{\alpha}{2}\right)$$

$\bar{\kappa}_{p,n}\left(\frac{\alpha}{2}\right)$:= traditional Wasserstein radius used if we know f^* (Kuhn et al., 2019)



Guarantees $\mathbb{P}\{d_W(\hat{P}_n^{ER}(x), P_{Y|X=x}) > \zeta_n(\alpha, x)\} \leq \alpha$

Flavor of Wasserstein ER-DRO Results

Informal Theorem (Finite Sample Certificate)

For the above choice of the Wasserstein radius $\zeta_n(\alpha, x)$, the solution $\hat{z}_n^{DRO}(x)$ and the optimal value $\hat{v}_n^{DRO}(x)$ satisfy

$$\mathbb{P}\left\{\mathbb{E}_\varepsilon\left[c(\hat{z}_n^{DRO}(x), f^*(x)+\varepsilon)\right] \leq \hat{v}_n^{DRO}(x)\right\} \geq 1-\alpha$$

Flavor of Wasserstein ER-DRO Results

Informal Theorem (Finite Sample Certificate)

For the above choice of the Wasserstein radius $\zeta_n(\alpha, x)$, the solution $\hat{z}_n^{DRO}(x)$ and the optimal value $\hat{v}_n^{DRO}(x)$ satisfy

$$\mathbb{P}\left\{\mathbb{E}_\varepsilon\left[c(\hat{z}_n^{DRO}(x), f^*(x)+\varepsilon)\right] \leq \hat{v}_n^{DRO}(x)\right\} \geq 1-\alpha$$

Informal Theorem (Rate of Convergence)

Suppose there is a sequence of risk levels $\{\alpha_n\} \subset (0,1)$ such that $\sum_n \alpha_n < +\infty$ and the radius satisfies $\lim_{n\to\infty} \zeta_n(\alpha_n, x) = 0$. Then the sequence $\{\hat{z}_n^{DRO}(x)\}$ of solutions satisfies

$$\mathbb{E}_\varepsilon\left[c(\hat{z}_n^{DRO}(x), f^*(x)+\varepsilon)\right] = v^*(x) + O_p(\zeta_n(\alpha_n, x))$$

Numerical Study: Optimal Resource Allocation

- Meet demands of 30 customer types for 20 resources (two-stage stochastic LP with r.h.s. uncertainty)
- Uncertain demands Y generated according to

$$Y_j = \alpha_j^* + \sum_{l=1}^{3} \beta_{jl}^* (X_l)^{\theta} + \varepsilon_j, \quad \forall j \in \{1, \cdots, 30\},$$

where $\varepsilon_j \sim \mathcal{N}(0, \sigma_j^2)$, $\theta \in \{0.5, 1, 2\}$, $\dim(X) \in \{10, 100\}$

Numerical Study: Optimal Resource Allocation

- Meet demands of 30 customer types for 20 resources (two-stage stochastic LP with r.h.s. uncertainty)
- Uncertain demands Y generated according to

$$Y_j = \alpha_j^* + \sum_{l=1}^{3} \beta_{jl}^* (X_l)^{\theta} + \varepsilon_j, \quad \forall j \in \{1, \cdots, 30\},$$

  where $\varepsilon_j \sim \mathcal{N}(0, \sigma_j^2)$, $\theta \in \{0.5, 1, 2\}$, $\dim(X) \in \{10, 100\}$
- Fit linear model with OLS/Lasso regression (even when $\theta \neq 1$)

$$Y_j = \alpha_j + \sum_{l=1}^{\dim(X)} \beta_{jl} X_l + \eta_j, \quad \forall j \in \{1, \cdots, 30\},$$

  where η_j are zero-mean errors
- Estimate optimality gap of solutions $\hat{z}_n^{ER}(x)$ and $\hat{z}_n^{J}(x)$

Results with Correct Model Class ($\theta = 1$)

Green (k): ER-SAA+kNN

Blue (O): ER-SAA+OLS

Black (R): Reweighted SAA with kNN (Bertsimas and Kallus, 2020)

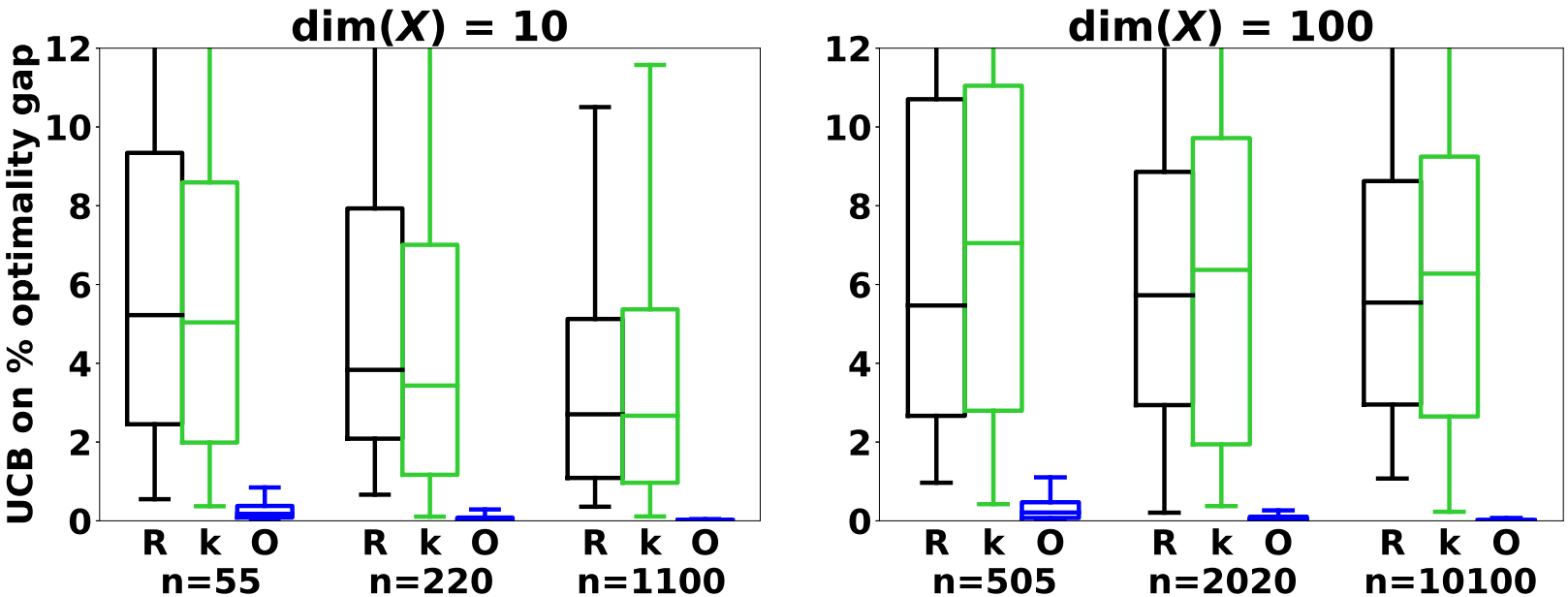
Results with Correct Model Class ($\theta = 1$)

Green (k): ER-SAA+kNN

Blue (O): ER-SAA+OLS

Black (R): Reweighted SAA with kNN (Bertsimas and Kallus, 2020)

Lower y-axis value $\implies$ closer to optimal



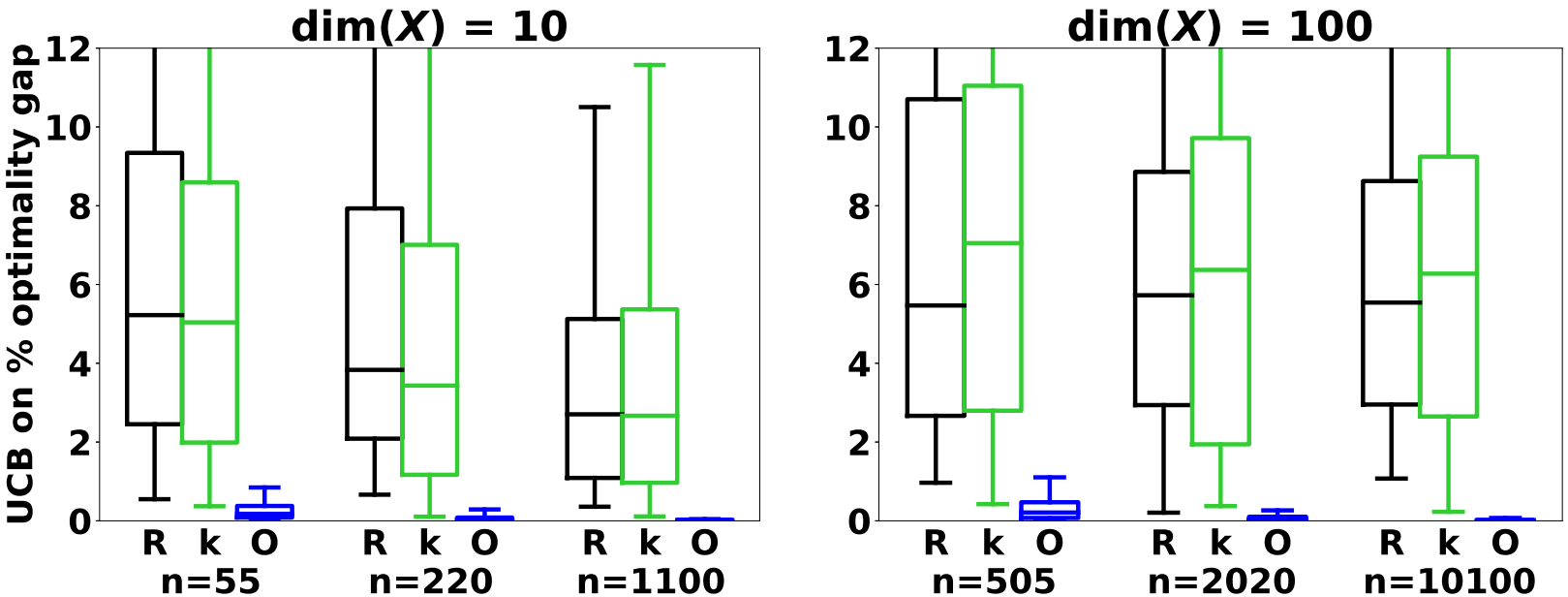
Results with Correct Model Class ($\theta = 1$)

Green (k): ER-SAA+kNN

Blue (O): ER-SAA+OLS

Black (R): Reweighted SAA with kNN (Bertsimas and Kallus, 2020)

Lower y-axis value $\implies$ closer to optimal



Boxes: 25, 50, and 75 percentiles of 99% upper confidence bounds

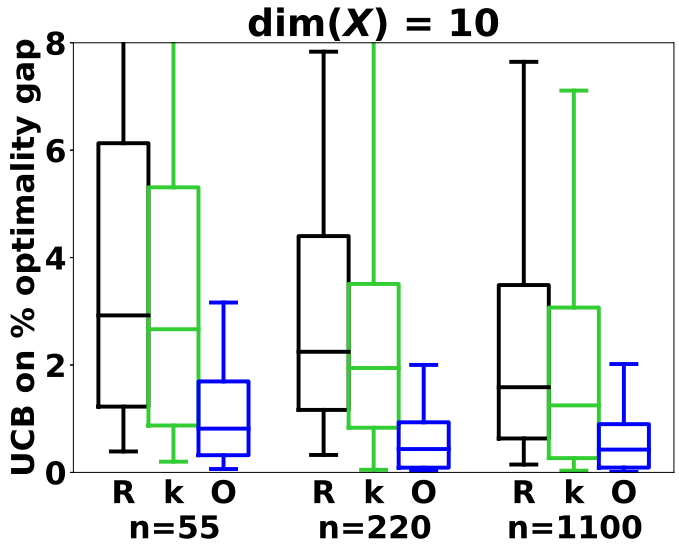
Whiskers: 5 and 95 percentiles

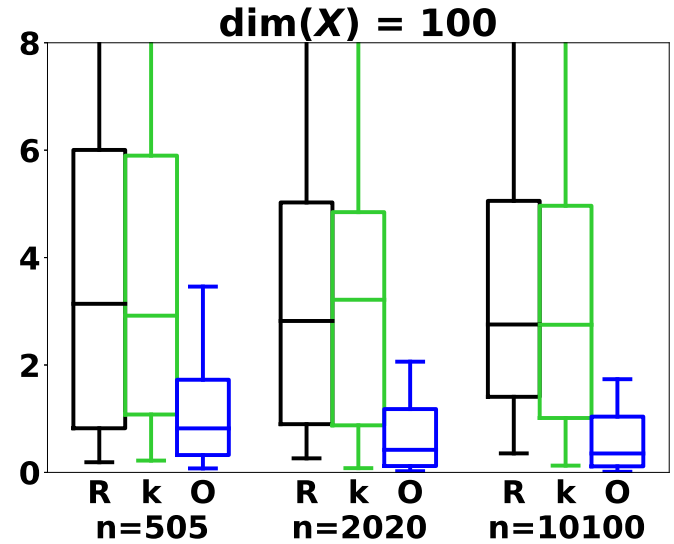
Sample sizes: $\{5, 20, 100\} \times (\dim(X) + 1)$

Results with Misspecified Model Class ($\theta \neq 1$)

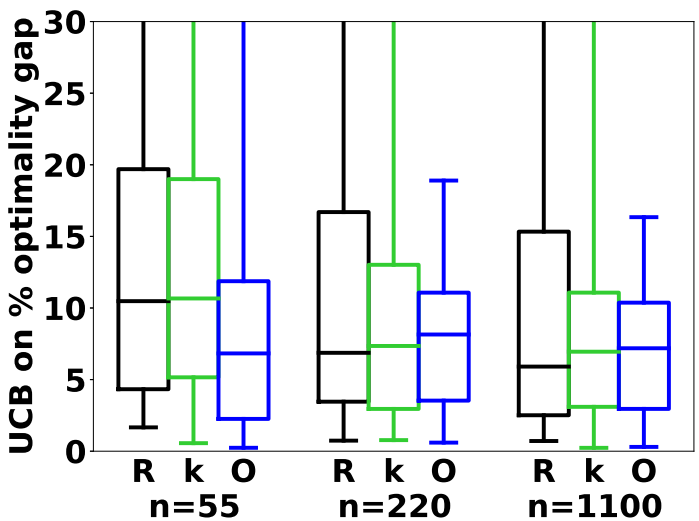
O: ER-SAA+OLS, k: ER-SAA+kNN, R: Reweighted SAA with kNN

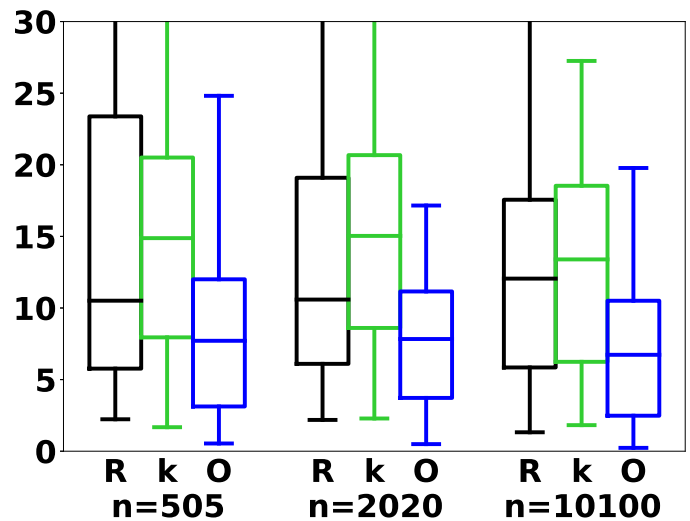
$\theta = 0.5$





$\theta = 2$

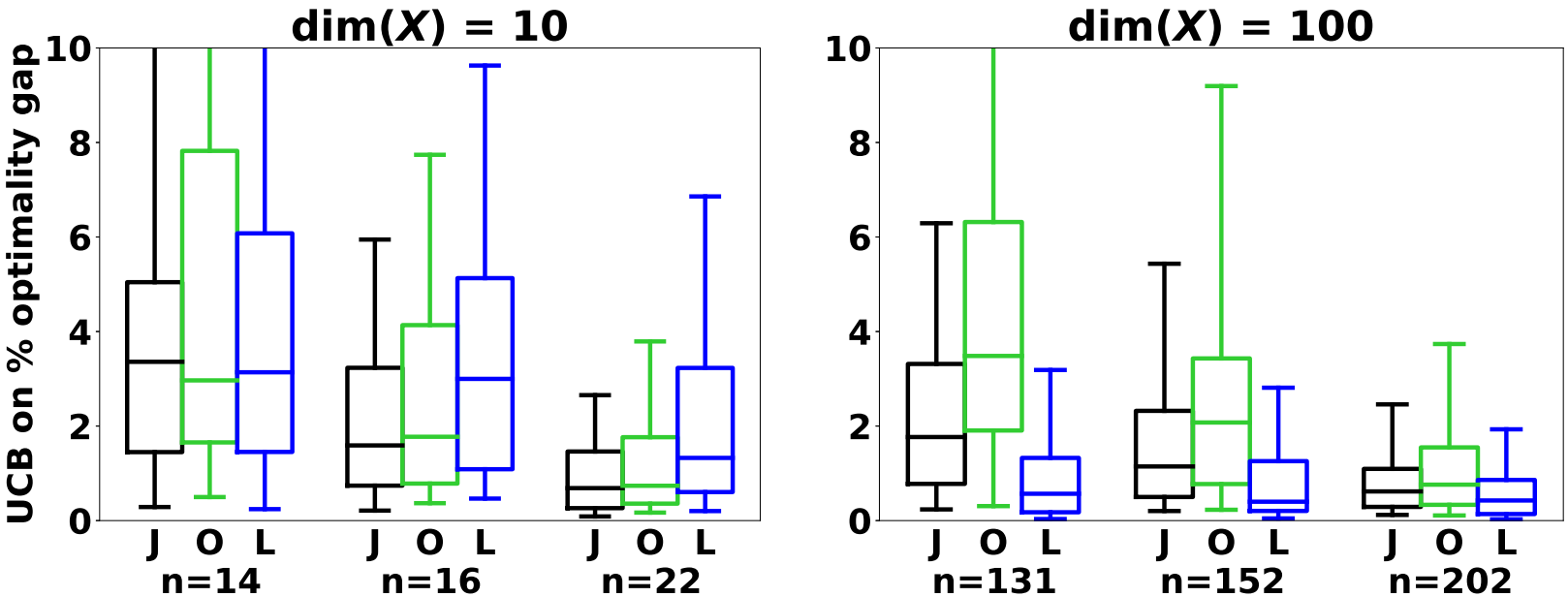




Advantage of J-SAA, Modularity with Limited Data ($\theta = 1$)

Black (J): J-SAA+OLS, Green (O): ER-SAA+OLS, Blue (L): ER-SAA+Lasso

Lower y-axis value $\implies$ closer to optimal



Boxes: 25, 50, and 75 percentiles of 99% upper confidence bounds

Whiskers: 5 and 95 percentiles

Sample sizes: $\{1.3, 1.5, 2\} \times (\dim(X) + 1)$

Part 1: Concluding Remarks

Empirical residuals formulations: A modular approach to using covariate information in optimization

- Converges under appropriate assumptions on prediction and optimization models
- Trade-off in choosing prediction model class: using a misspecified model can lead to better results with limited data
- Preprints: arXiv:2207.13554 and arXiv:2012.01088 with lots of additional theory and experiments
- Ongoing: multistage stochastic opt. for time series data

Part 1: Concluding Remarks

Empirical residuals formulations: A modular approach to using covariate information in optimization

- Converges under appropriate assumptions on prediction and optimization models
- Trade-off in choosing prediction model class: using a misspecified model can lead to better results with limited data
- Preprints: arXiv:2207.13554 and arXiv:2012.01088 with lots of additional theory and experiments
- Ongoing: multistage stochastic opt. for time series data

Future work

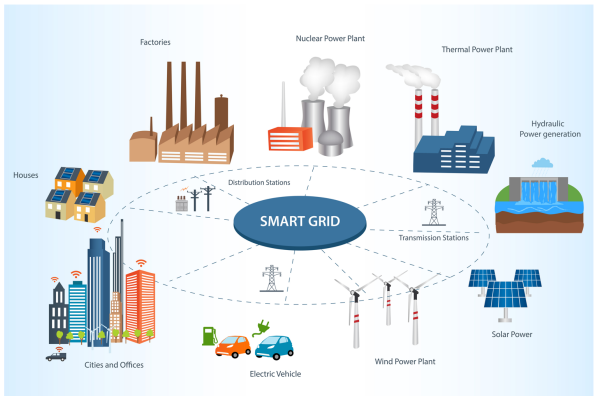
- Formulations with stochastic constraints, discrete recourse decisions; robust multistage optimization
- Application to energy systems optimization

Outline

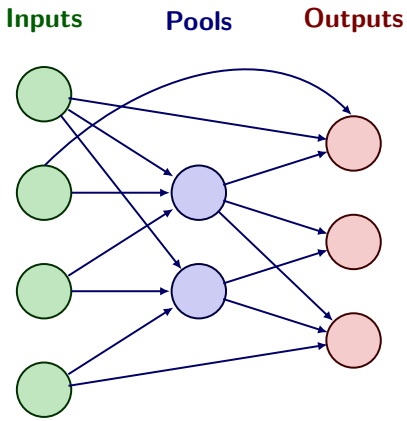
Motivation

Many important applications can be formulated as nonconvex QCQPs

AC Optimal Power Flow



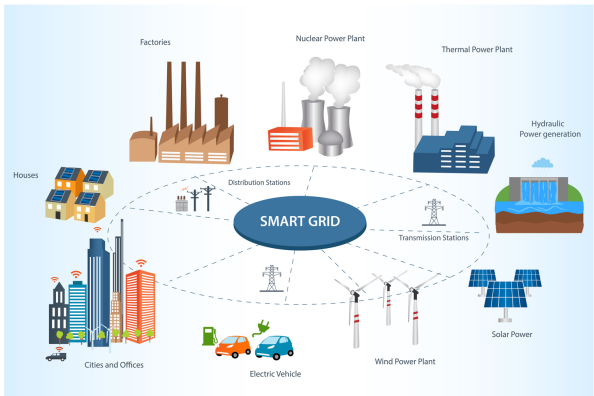
The Pooling Problem



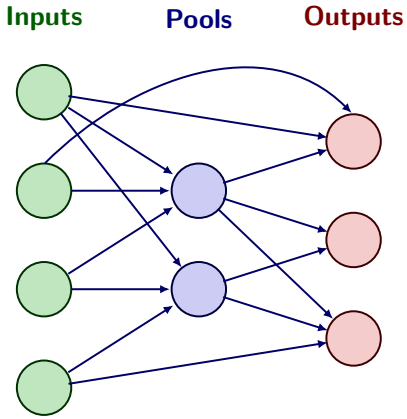
Motivation

Many important applications can be formulated as nonconvex QCQPs

AC Optimal Power Flow



The Pooling Problem

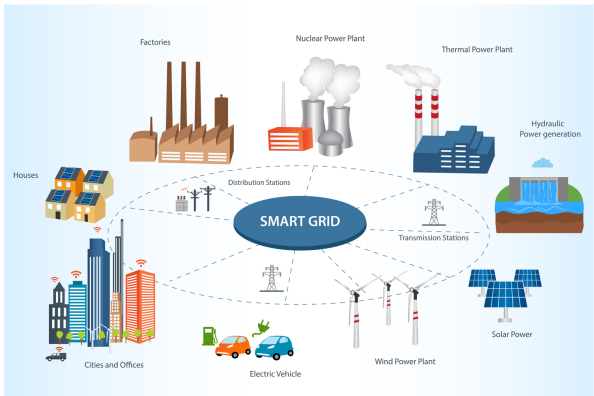


Often, wish to *repeatedly* solve instances of the same nonconvex problem with different data, e.g., loads, wind, qualities, prices

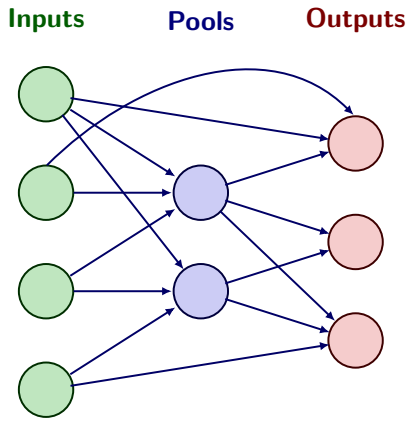
Motivation

Many important applications can be formulated as nonconvex QCQPs

AC Optimal Power Flow



The Pooling Problem



Often, wish to *repeatedly* solve instances of the same nonconvex problem with different data, e.g., loads, wind, qualities, prices

Can we exploit *shared structure* to accelerate global solution?

Harsha Nagarajan
(LANL)

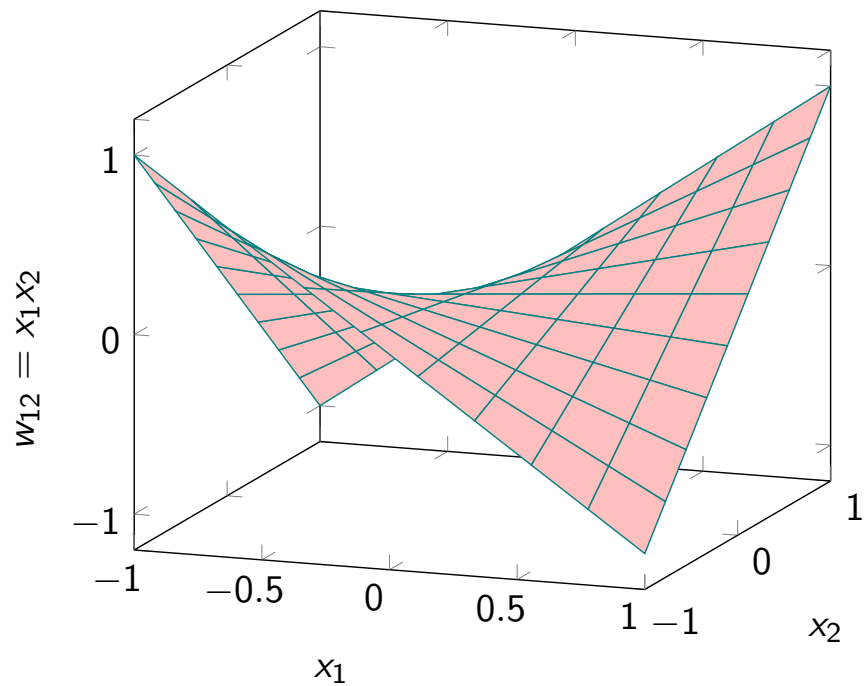
Deepjyoti Deka
(LANL)

Global Optimization of QCQPs

Consider the following class of QCQPs:

$$\begin{aligned} \nu^* := \min_{x,w} \; & c^\mathsf{T} x + d^\mathsf{T} w \\ \text{s.t.} \; & w_{ij} = x_i x_j, \quad \forall (i,j) \in \mathcal{B}, \\ & Ax + Bw \leq b, \;\; x \in [-1,1]^{d_x} \end{aligned}$$

- The bilinear constraints are what make the problem hard



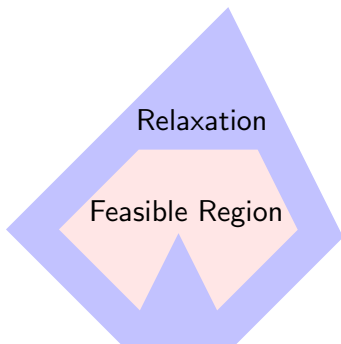
K., Nagarajan, and Deka. Learning to Accelerate the Global Optimization of QCQPs. arXiv:2301.00306. Under Review

Global Optimization of QCQPs

Consider the following class of QCQPs:

$$\begin{aligned} \nu^* := \min_{x,w} \; & c^\mathsf{T} x + d^\mathsf{T} w \\ \text{s.t.} \; & w_{ij} = x_i x_j, \quad \forall (i,j) \in \mathcal{B}, \\ & Ax + Bw \leq b, \;\; x \in [-1,1]^{d_x} \end{aligned}$$

- The bilinear constraints are what make the problem hard
- Get feasible solutions/upper bounds using local optimization
- Obtain lower bounds on ν^* using relaxations



K., Nagarajan, and Deka. Learning to Accelerate the Global Optimization of QCQPs. arXiv:2301.00306. Under Review

Relaxing Bilinear Terms

The feasible region of the hard *bilinear* constraints

$$w_{ij} = x_i x_j, \quad x_i, x_j \in [-1, 1] \tag{1}$$

is a subset of the feasible region of the easy *linear* constraints

$$\begin{aligned} -x_i - x_j - 1 \leq w_{ij} \leq x_i - x_j + 1, \\ x_i + x_j - 1 \leq w_{ij} \leq x_j - x_i + 1, \\ x_i, x_j \in [-1, 1] \end{aligned} \tag{2}$$

Relaxing Bilinear Terms

The feasible region of the hard *bilinear* constraints

$$w_{ij} = x_i x_j, \quad x_i, x_j \in [-1, 1] \tag{1}$$

is a subset of the feasible region of the easy *linear* constraints

$$\begin{aligned} -x_i - x_j - 1 \leq w_{ij} \leq x_i - x_j + 1, \\ x_i + x_j - 1 \leq w_{ij} \leq x_j - x_i + 1, \\ x_i, x_j \in [-1, 1] \end{aligned} \tag{2}$$

Replace bilinear constraints (1) in the QCQP with McCormick Relaxations (2) to determine a valid lower bound

$$\begin{aligned} \nu^* \geq \nu^M := \min_{x,w} \ & c^\mathsf{T} x + d^\mathsf{T} w \\ \text{s.t.} \ & Ax + Bw \leq b, \\ & -x_i - x_j - 1 \leq w_{ij} \leq x_i - x_j + 1, \quad \forall (i,j) \in \mathcal{B}, \\ & x_i + x_j - 1 \leq w_{ij} \leq x_j - x_i + 1, \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1, 1]^{d_x} \end{aligned}$$

Relaxing Bilinear Terms

The feasible region of the hard *bilinear* constraints

$$w_{ij} = x_i x_j, \quad x_i, x_j \in [-1, 1] \tag{1}$$

is a subset of the feasible region of the easy *linear* constraints

$$\begin{aligned} -x_i - x_j - 1 \leq w_{ij} \leq x_i - x_j + 1, \\ x_i + x_j - 1 \leq w_{ij} \leq x_j - x_i + 1, \\ x_i, x_j \in [-1, 1] \end{aligned} \tag{2}$$

Replace bilinear constraints (1) in the QCQP with McCormick Relaxations (2) to determine a valid lower bound

$$\begin{aligned} \nu^* \geq \nu^M := \min_{x,w} \ & c^T x + d^T w \\ \text{s.t.} \ & Ax + Bw \leq b, \\ & -x_i - x_j - 1 \leq w_{ij} \leq x_i - x_j + 1, \quad \forall (i,j) \in \mathcal{B}, \\ & x_i + x_j - 1 \leq w_{ij} \leq x_j - x_i + 1, \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1, 1]^{d_x} \end{aligned}$$

Typically $\nu^M \ll \nu^*$, and the gap is closed using continuous B&B

Tighten Relaxations By Partitioning Variable Domains

- Partition variable domains into "disjoint" subintervals, e.g.,

$$x_1 \in [-1, 0] \text{ OR } [0, 1]$$
$$x_2 \in [-1, 0] \text{ OR } [0, 1]$$

Tighten Relaxations By Partitioning Variable Domains

- Partition variable domains into "disjoint" subintervals, e.g.,

$$x_1 \in [-1, 0] \text{ OR } [0, 1]$$
$$x_2 \in [-1, 0] \text{ OR } [0, 1]$$

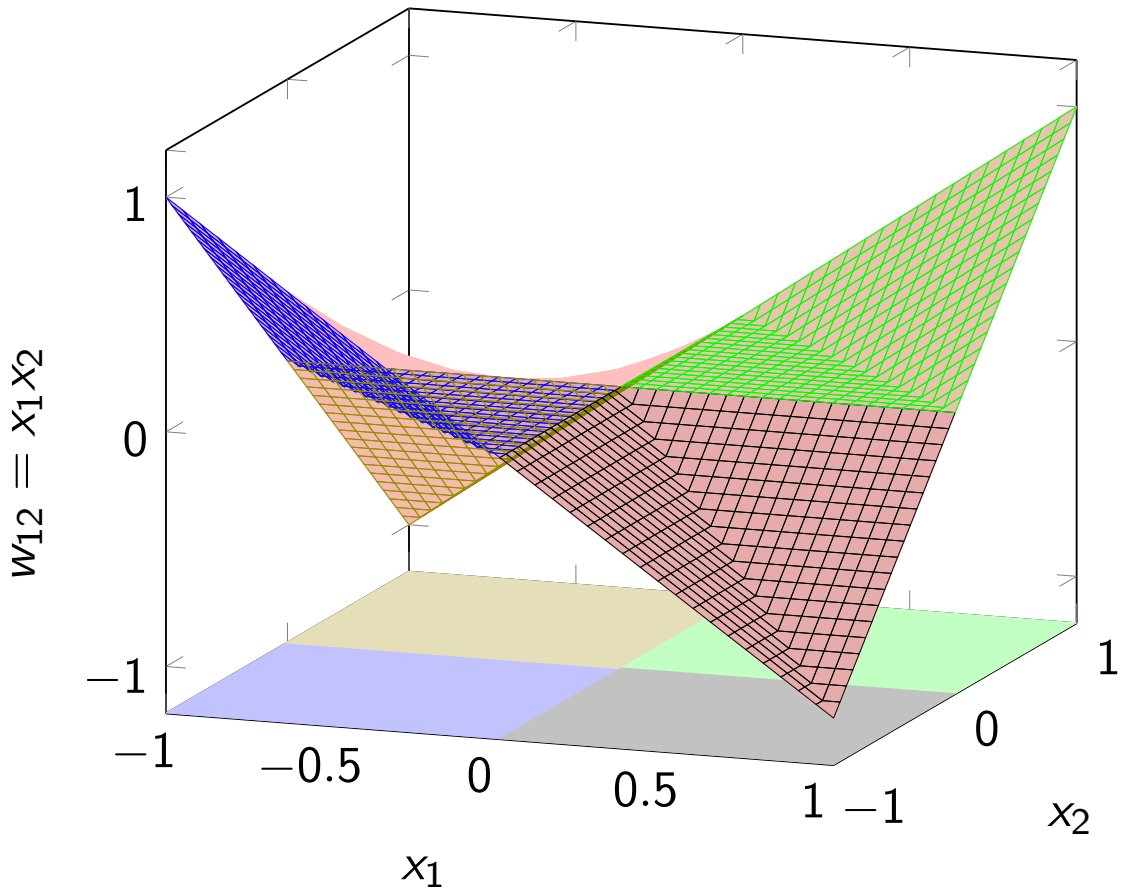
- Construct Piecewise McCormick Relaxations on the variable partitions and solve a MIP to obtain lower bound

$$\begin{aligned} \nu^* \geq \nu^{PMR} := \min_{x,w} \; & c^T x + d^T w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1, 1]^{d_x}, \end{aligned}$$

where p_i is the vector of partitioning points for x_i

The Lower Part of the Piecewise McCormick Relaxations

Partitions: $x_1 \in [-1, 0]$ OR $[0, 1]$, $\quad x_2 \in [-1, 0]$ OR $[0, 1]$



Refine Variable Partitions for Convergence

- Partition variable domains into "disjoint" subintervals, e.g.,

$$x_1 \in [-1, 0] \text{ OR } [0, 1]$$
$$x_2 \in [-1, 0] \text{ OR } [0, 1]$$

- Construct Piecewise McCormick Relaxations on the variable partitions and solve a MIP to obtain lower bound

$$\begin{aligned} \nu^* \geq \nu^{PMR} := \min_{x,w} \; & c^\mathsf{T} x + d^\mathsf{T} w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1, 1]^{d_x}, \end{aligned}$$

where p_i is the vector of partitioning points for x_i

Refine Variable Partitions for Convergence

- Partition variable domains into "disjoint" subintervals, e.g.,

$$x_1 \in [-1, 0] \text{ OR } [0, 1]$$
$$x_2 \in [-1, 0] \text{ OR } [0, 1]$$

- Construct Piecewise McCormick Relaxations on the variable partitions and solve a MIP to obtain lower bound

$$\begin{aligned} \nu^* \geq \nu^{PMR} := \min_{x,w} \; & c^T x + d^T w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1, 1]^{d_x}, \end{aligned}$$

  where p_i is the vector of partitioning points for x_i

- Refine variable partitions to close gap between ν^{PMR} and ν^*

$$\begin{aligned} \text{e.g.} \quad & x_1 \in [-1, -0.5] \text{ OR } [-0.5, 0] \text{ OR } [0, 1] \\ & x_2 \in [-1, 0] \quad \text{OR } [0, 0.2] \quad \text{OR } [0.2, 1] \end{aligned}$$

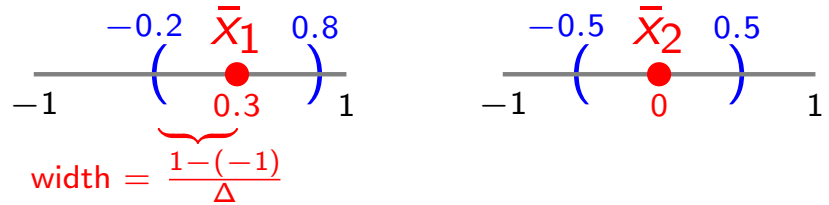
How to Pick Partitioning Points?

Adaptive strategy in the solver Alpine (Nagarajan et al., 2019): refine partitions around a reference point $\bar{x}$ (e.g., around a feasible point or solution to McCormick relaxation)

How to Pick Partitioning Points?

Adaptive strategy in the solver Alpine (Nagarajan et al., 2019): refine partitions around a reference point $\bar{x}$ (e.g., around a feasible point or solution to McCormick relaxation)

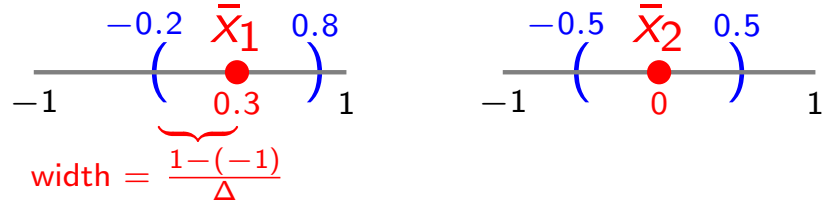
- Example: if $\bar{x} = (0.3, 0)$ and parameter $\Delta = 4$



How to Pick Partitioning Points?

Adaptive strategy in the solver Alpine (Nagarajan et al., 2019): refine partitions around a reference point $\bar{x}$ (e.g., around a feasible point or solution to McCormick relaxation)

- Example: if $\bar{x} = (0.3, 0)$ and parameter $\Delta = 4$



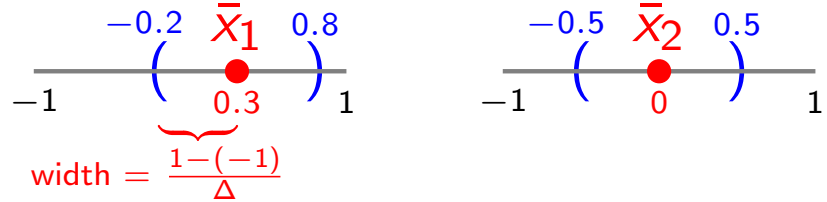
Best choice of Δ can vary depending on instance (illustration on 3 random QCQPs)

Δ	**4**	**10**	**15**
Time for Ex1:	5087s	**704s**	1551s
Time for Ex2:	**2632s**	5023s	6642s
Time for Ex3:	3000s	4540s	**1433s**

How to Pick Partitioning Points?

Adaptive strategy in the solver Alpine (Nagarajan et al., 2019): refine partitions around a reference point $\bar{x}$ (e.g., around a feasible point or solution to McCormick relaxation)

- Example: if $\bar{x} = (0.3, 0)$ and parameter $\Delta = 4$



Best choice of Δ can vary depending on instance (illustration on 3 random QCQPs)

Δ	**4**	**10**	**15**
Time for Ex1:	5087s	**704s**	1551s
Time for Ex2:	**2632s**	5023s	6642s
Time for Ex3:	3000s	4540s	**1433s**

Can we choose better partitioning points for faster convergence?

More partitioning points $\implies$ tighter lower bounds at the expense of harder MIPs

Strong Partitioning (SP) to Improve Choice of Partitions

New Approach: Choose partitioning points to maximize the lower bound

$$p^* \in \arg\max_{p \in P} \nu^{PMR}(p),$$

- p_i is the vector of partitioning points for x_i

$$\begin{aligned} \nu^{PMR}(p) := \min_{x,w} \; & c^T x + d^T w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1,1]^{d_x}, \end{aligned}$$

Strong Partitioning (SP) to Improve Choice of Partitions

New Approach: Choose partitioning points to maximize the lower bound

$$p^* \in \arg\max_{p \in P} \nu^{PMR}(p),$$

- p_i is the vector of partitioning points for x_i

$$\begin{aligned} \nu^{PMR}(p) := \min_{x,w} \; & c^\mathsf{T} x + d^\mathsf{T} w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1, 1]^{d_x}, \end{aligned}$$

- From iteration 2, use aforementioned partitioning strategy (guaranteed to converge irrespective of points chosen by SP)

Strong Partitioning (SP) to Improve Choice of Partitions

New Approach: Choose partitioning points to maximize the lower bound

$$p^* \in \arg\max_{p \in P} \nu^{PMR}(p),$$

- p_i is the vector of partitioning points for x_i

$$\begin{aligned} \nu^{PMR}(p) := \min_{x,w} \; & c^T x + d^T w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1,1]^{d_x}, \end{aligned}$$

- From iteration 2, use aforementioned partitioning strategy (guaranteed to converge irrespective of points chosen by SP)

How to solve this max-min problem (locally)?

Using generalized gradients of value function ν^{PMR} within a bundle solver

Strong Partitioning (SP) to Improve Choice of Partitions

New Approach: Choose partitioning points to maximize the lower bound

$$p^* \in \arg\max_{p \in P} \nu^{PMR}(p),$$

- p_i is the vector of partitioning points for x_i

$$\begin{aligned} \nu^{PMR}(p) := \min_{x,w} \; & c^T x + d^T w \\ \text{s.t.} \; & Ax + Bw \leq b, \\ & (x_i, x_j, w_{ij}) \in \mathcal{PMR}_{ij}(p_i, p_j), \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [-1,1]^{d_x}, \end{aligned}$$

- From iteration 2, use aforementioned partitioning strategy (guaranteed to converge irrespective of points chosen by SP)

How to solve this max-min problem (locally)?

Using generalized gradients of value function ν^{PMR} within a bundle solver

Solving this max-min problem may be as hard as solving the QCQP!

Using ML to Accelerate Partitioning (Within Alpine)

Given family of random QCQPs of the form (Bao et al., 2009)

$$\begin{aligned} \nu^*(\theta) := \min_{x,w} \; & c(\theta)^\mathsf{T} x + d(\theta)^\mathsf{T} w \\ \text{s.t.} \; & A(\theta)x + B(\theta)w \leq b, \\ & w_{ij} = x_i x_j, \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [0,1]^{d_x} \end{aligned}$$

Test instances

- $d_x \in \{10, 20, 50\}$
- $5d_x$ bilinear terms
- d_x bilinear inequalities
- $d_x/5$ linear equalities

Parameters θ vary from one instance to the next

Using ML to Accelerate Partitioning (Within Alpine)

Given family of random QCQPs of the form (Bao et al., 2009)

$$\begin{aligned} \nu^*(\theta) := \min_{x,w} \; & c(\theta)^\mathsf{T} x + d(\theta)^\mathsf{T} w \\ \text{s.t.} \; & A(\theta)x + B(\theta)w \leq b, \\ & w_{ij} = x_i x_j, \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [0,1]^{d_x} \end{aligned}$$

Test instances

$d_x \in \{10, 20, 50\}$

$5d_x$ bilinear terms

d_x bilinear inequalities

$d_x/5$ linear equalities

Parameters θ vary from one instance to the next

Input: underlying problem, distribution of parameters θ

Output: ML model that predicts partitioning points given $\bar{\theta}$

Using ML to Accelerate Partitioning (Within Alpine)

Given family of random QCQPs of the form (Bao et al., 2009)

$$\begin{aligned} \nu^*(\theta) := \min_{x,w} \; & c(\theta)^\mathsf{T} x + d(\theta)^\mathsf{T} w \\ \text{s.t.} \; & A(\theta)x + B(\theta)w \leq b, \\ & w_{ij} = x_i x_j, \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [0,1]^{d_x} \end{aligned}$$

Test instances

$d_x \in \{10, 20, 50\}$

$5d_x$ bilinear terms

d_x bilinear inequalities

$d_x/5$ linear equalities

Parameters θ vary from one instance to the next

Input: underlying problem, distribution of parameters θ
Output: ML model that predicts partitioning points given $\bar{\theta}$

- Generate N training samples $\{\theta^i\}$ of the problem parameters θ
- Solve max-min problem to determine "optimal" partitioning points for each training instance

Using ML to Accelerate Partitioning (Within Alpine)

Given family of random QCQPs of the form (Bao et al., 2009)

$$\begin{aligned} \nu^*(\theta) := \min_{x,w} \; & c(\theta)^\mathsf{T} x + d(\theta)^\mathsf{T} w \\ \text{s.t.} \; & A(\theta)x + B(\theta)w \leq b, \\ & w_{ij} = x_i x_j, \quad \forall (i,j) \in \mathcal{B}, \\ & x \in [0,1]^{d_x} \end{aligned}$$

Test instances

$d_x \in \{10, 20, 50\}$

$5d_x$ bilinear terms

d_x bilinear inequalities

$d_x/5$ linear equalities

Parameters θ vary from one instance to the next

Input: underlying problem, distribution of parameters θ
Output: ML model that predicts partitioning points given $\bar{\theta}$

- Generate N training samples $\{\theta^i\}$ of the problem parameters θ
- Solve max-min problem to determine "optimal" partitioning points for each training instance
- Learn an ML model $\theta^i \mapsto$ `optimal partitioning points` (use scikit-learn's AdaBoostRegressor with 10-fold CV)
- Use ML model to predict partitioning points for new instance $\bar{\theta}$

Numerical Results for Random QCQPs

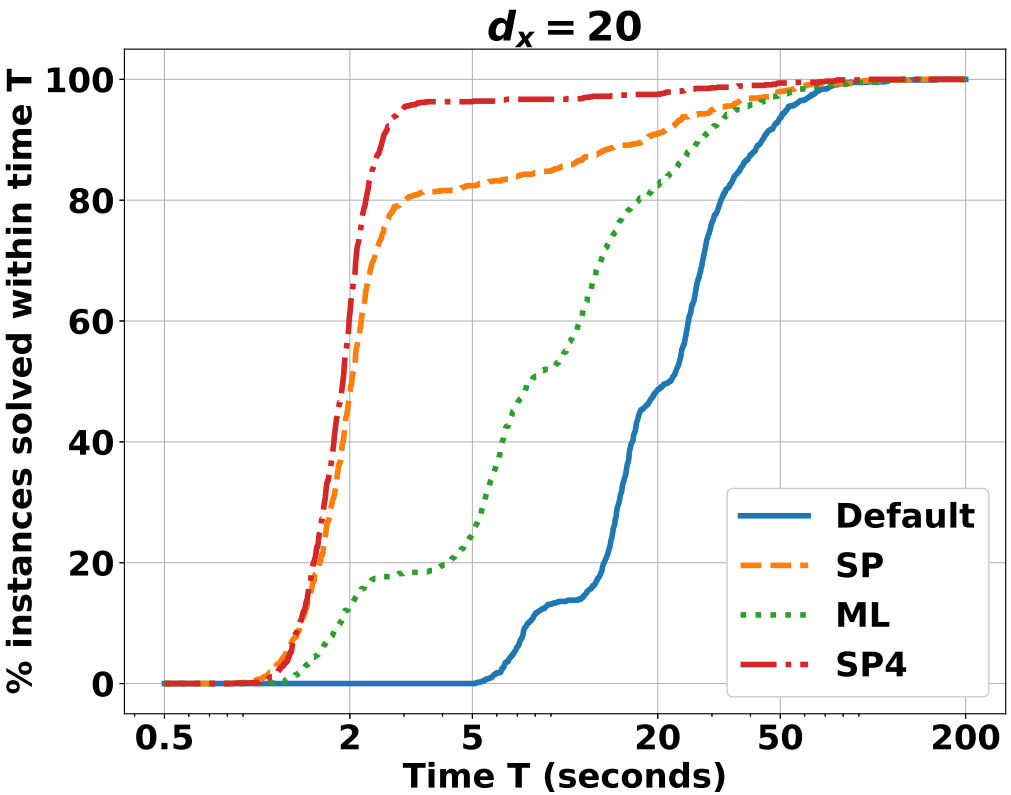
Results for $d_x = 20$ variables

- Generate 1000 random QCQPs with varying parameters θ
- determine 2/4 SP points per variable for each instance
- Eliminate partitioning points that aren't useful

Numerical Results for Random QCQPs

Results for $d_x = 20$ variables

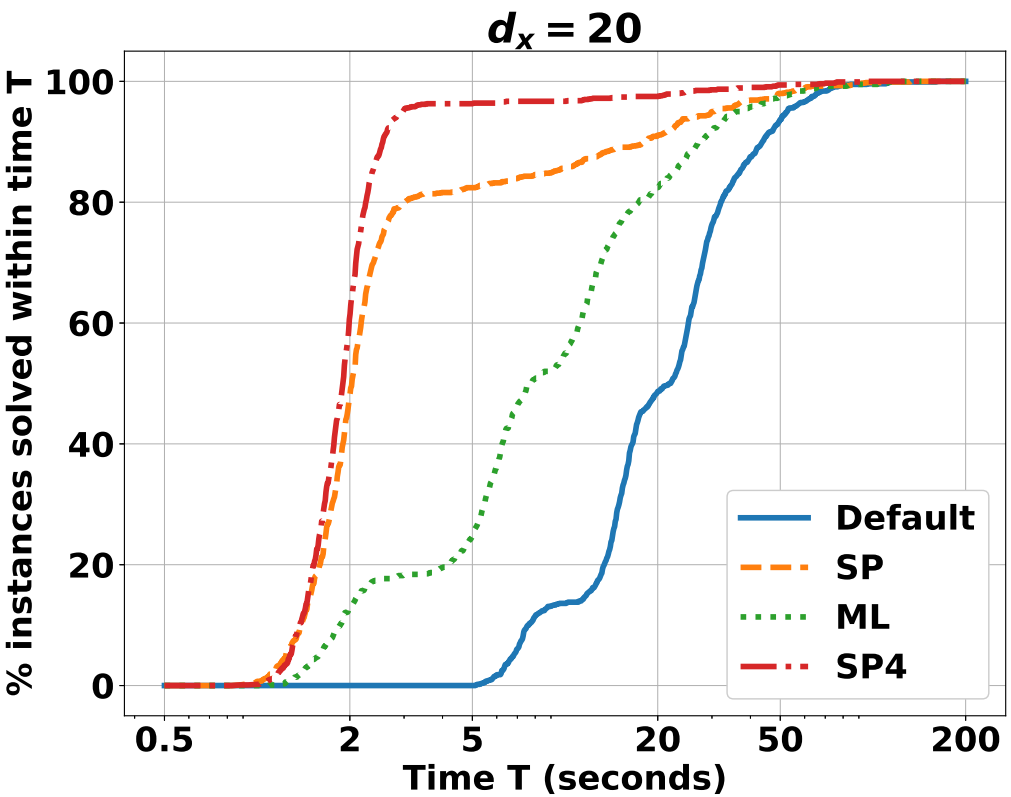
- Generate 1000 random QCQPs with varying parameters θ
- determine 2/4 SP points per variable for each instance
- Eliminate partitioning points that aren't useful



Numerical Results for Random QCQPs

Results for $d_x = 20$ variables

- Generate 1000 random QCQPs with varying parameters θ
- determine 2/4 SP points per variable for each instance
- Eliminate partitioning points that aren't useful

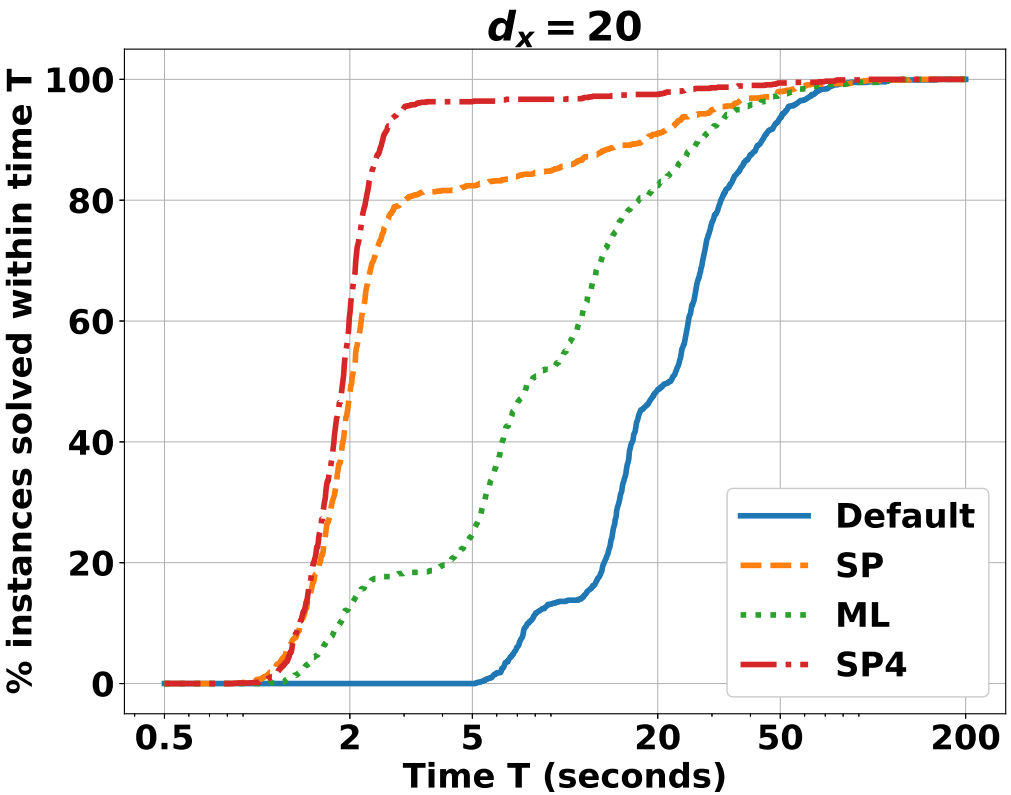


Speedup/ Slowdown	% SP Inst.	% ML Inst.
$1x - 3x$	13.1	48.7
$3x - 5x$	12.3	16.0
$5x - 10x$	31.2	15.3
$10x - 20x$	29.9	6.0
$> 20x$	10.0	0.9
$0.5x - 1x$	3.3	9.8
$< 0.5x$	0.2	3.3

Numerical Results for Random QCQPs

Results for $d_x = 20$ variables

- Generate 1000 random QCQPs with varying parameters θ
- determine 2/4 SP points per variable for each instance
- Eliminate partitioning points that aren't useful



Speedup/ Slowdown	% SP Inst.	% ML Inst.
$1x - 3x$	13.1	48.7
$3x - 5x$	12.3	16.0
$5x - 10x$	31.2	15.3
$10x - 20x$	29.9	6.0
$> 20x$	10.0	0.9
$0.5x - 1x$	3.3	9.8
$< 0.5x$	0.2	3.3

Average Speedup (Shifted GM):

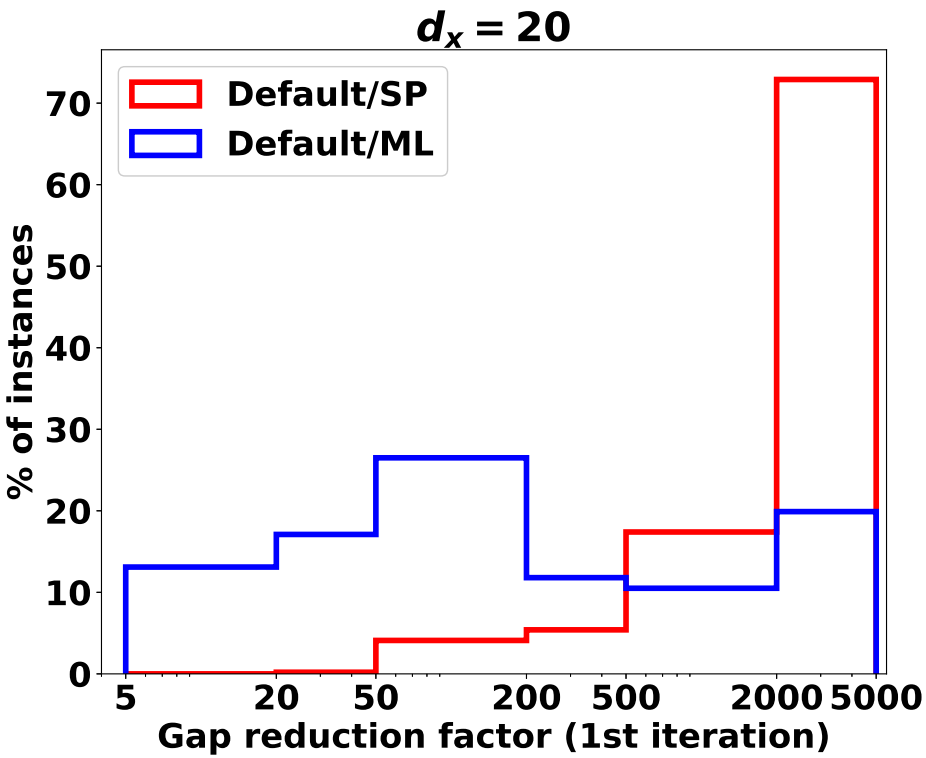
Alpine+SP: $5.1x$, Alpine+ML: $2.1x$

Alpine+SP4: $9x$, Alpine+ML4: $2.3x$

Numerical Results for Random QCQPs

Results for $d_x = 20$ variables

- Generate 1000 random QCQPs with varying parameters θ
- determine 2/4 SP points per variable for each instance
- Eliminate partitioning points that aren't useful



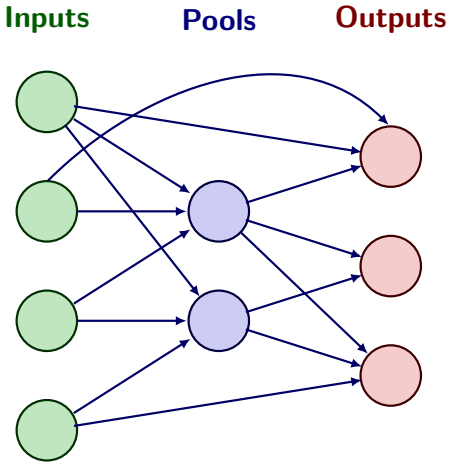
Speedup/ Slowdown	% SP Inst.	% ML Inst.
$1x - 3x$	13.1	48.7
$3x - 5x$	12.3	16.0
$5x - 10x$	31.2	15.3
$10x - 20x$	29.9	6.0
$> 20x$	10.0	0.9
$0.5x - 1x$	3.3	9.8
$< 0.5x$	0.2	3.3

Average Speedup (Shifted GM):

Alpine+SP: $5.1x$, Alpine+ML: $2.1x$

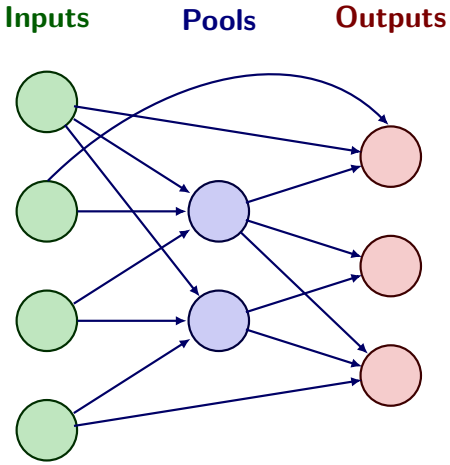
Alpine+SP4: $9x$, Alpine+ML4: $2.3x$

Numerical Results for the Pooling Problem



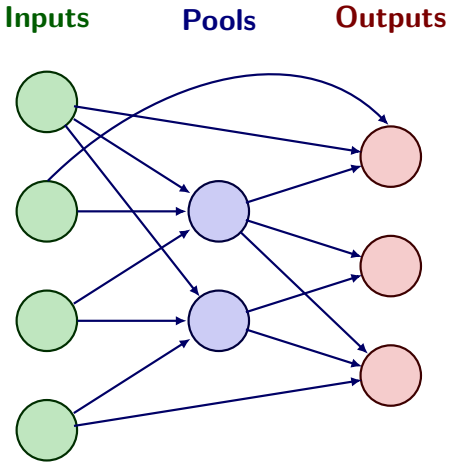
- 45 sources, 15 pools, 30 terminals, 1 quality (124/572 variables part. in 261 bilinear terms)
- 1000 random instances with θ = input qualities
- 2 SP points per variable (total 124×2)

Numerical Results for the Pooling Problem

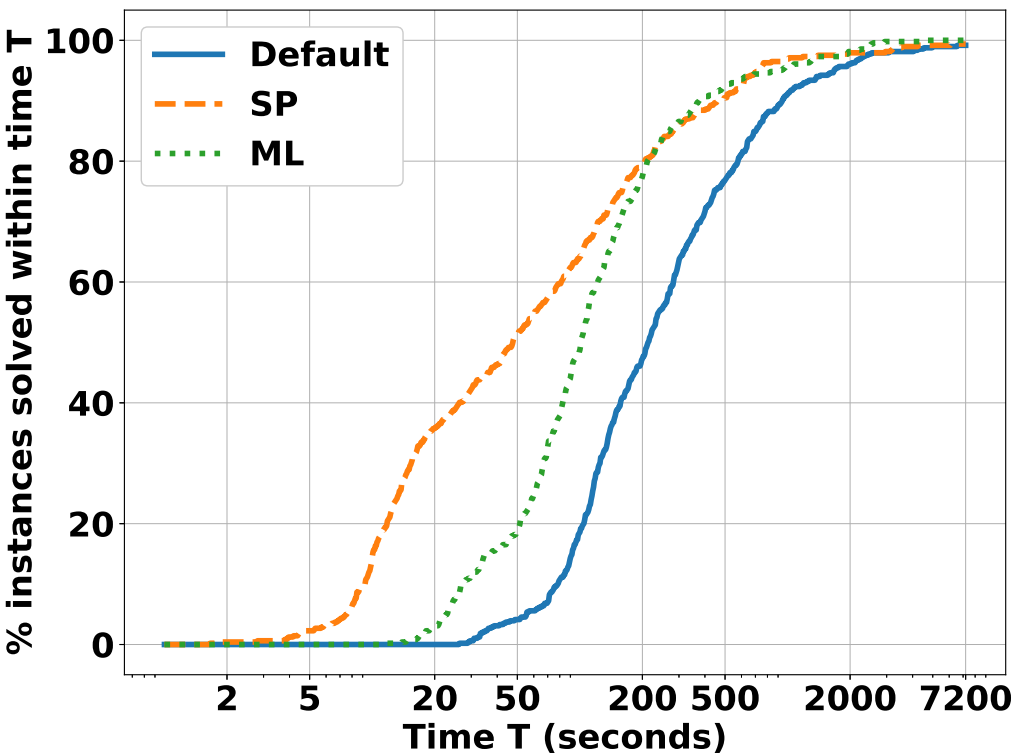


- 45 sources, 15 pools, 30 terminals, 1 quality (124/572 variables part. in 261 bilinear terms)
- 1000 random instances with θ = input qualities
- 2 SP points per variable (total 124×2)
- Feature dimension: 667, Output dimension: 248

Numerical Results for the Pooling Problem



- 45 sources, 15 pools, 30 terminals, 1 quality (124/572 variables part. in 261 bilinear terms)
- 1000 random instances with θ = input qualities
- 2 SP points per variable (total 124×2)
- Feature dimension: 667, Output dimension: 248



Speedup/ Slowdown	% SP Inst.	% ML Inst.
$1x - 3x$	29.1	53.9
$3x - 5x$	16.1	21.5
$5x - 10x$	21.7	10.4
$10x - 20x$	20.3	1.6
$> 20x$	6.2	0.1
$0.5x - 1x$	4.5	1.7
$< 0.5x$	2.1	10.8

Average Speedup (Shifted GM):

Alpine+SP: $3.6x$, Alpine+ML: $2.1x$

Part 2: Concluding Remarks

Strong Partitioning provides an excellent benchmark for ML to accelerate partitioning algorithms for global optimization

- SP reduces Alpine's solution time by $4x - 16x$ on average (max. speedups of $15x - 700x$)
- SP can reduce Alpine's first iteration gap by more than $2000x$!
- Off-the-shelf ML model improves Alpine's run time by $2x - 4.5x$ on average (max. speedups of $10x - 200x$)

Part 2: Concluding Remarks

Strong Partitioning provides an excellent benchmark for ML to accelerate partitioning algorithms for global optimization

- SP reduces Alpine's solution time by $4x - 16x$ on average (max. speedups of $15x - 700x$)
- SP can reduce Alpine's first iteration gap by more than $2000x$!
- Off-the-shelf ML model improves Alpine's run time by $2x - 4.5x$ on average (max. speedups of $10x - 200x$)

Ongoing and future work

- Techniques for adaptive strong partitioning
- Investigate tailored ML models to imitate SP
- Extend SP to broader optimization classes, including MINLPs
- Explore application to AC optimal power flow

References

G.-Y. Ban and C. Rudin. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019.

X. Bao, N. V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods & Software*, 24(4-5):485–504, 2009.

R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486 – 507, 2021.

D. Bertsimas and N. Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

Y. Deng and S. Sen. Predictive stochastic programming. *Computational Management Science*, pages 1–34, 2022.

P. Donti, B. Amos, and J. Z. Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 5484–5494, 2017.

A. N. Elmachtoub and P. Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations Research & Management Science in the Age of Analytics*, pages 130–166. INFORMS, 2019.

H. Nagarajan, M. Lu, S. Wang, R. Bent, and K. Sundar. An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs. *Journal of Global Optimization*, 74(4):639–675, 2019.

ER-SAA

Numerical Study: Optimal Resource Allocation

$$\min_{z \geq 0} \; c^{\mathsf{T}} z + \mathbb{E}_Y[Q(z, Y)]$$

- z_i: quantity of resource $i \in \mathcal{I}$ (order before demands realized)
- Y_j: uncertain demand of customer type $j \in \mathcal{J}$

$$\begin{aligned} Q(z, Y) := \min_{w, v \geq 0} \quad & d^{\mathsf{T}} w \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}} v_{ij} \leq z_i, \quad \forall i \in \mathcal{I}, \\ & \sum_{i \in \mathcal{I}} \mu_{ij} v_{ij} + w_j \geq Y_j, \quad \forall j \in \mathcal{J}. \end{aligned}$$

- v_{ij}: amount of resource i allocated to customer type j
- w_j: amount of customer type j demand that is not met
- $\mu_{ij} \geq 0$: service rate of resource i for customer type j

Wasserstein ER-DRO

Choosing the Radius for Wasserstein ER-DRO in Practice

- Theoretical Wasserstein radius: involves unknown constants and is typically conservative
- Use cross-validation to specify the radius $\zeta_n(x)$
  - Approach 1: Ignore covariate information altogether while choosing ζ_n
  - Approach 2: Use the data $\mathcal{D}_n$ to choose ζ_n independently of the covariate realization $X = x$
  - Approach 3: Use both the data $\mathcal{D}_n$ and the covariate realization $X = x$ to choose the radius $\zeta_n(x)$
- Approach 3 is more data intensive than Approaches 1 & 2

Numerical Study: Mean-CVaR Portfolio Optimization

$$\min_{z \in \mathcal{Z}} \ \mathbb{E}_Y[-Y^\mathsf{T} z] + \rho \, \mathrm{CVaR}_\beta(-Y^\mathsf{T} z),$$

where $\mathcal{Z} := \left\{ z \in \mathbb{R}^{d_z}_+ : \sum_i z_i = 1 \right\}$.

- z_i: fraction of capital invested in asset i
- Y_i: uncertain net return of asset i
- $\mathrm{CVaR}_\beta \approx$ average of the $100(1-\beta)\%$ worst return outcomes
- $\rho \geq 0$ and $\beta \in [0, 1)$: risk parameters (e.g., $\rho = 10$, $\beta = 0.8$)

Numerical Study: Mean-CVaR Portfolio Optimization

- Consider instance with 10 assets
- Uncertain returns Y generated according to

$$Y_j = \nu_j^* + \sum_{l=1}^{3} \mu_{jl}^* (X_l)^{\theta} + \bar{\varepsilon}_j + \omega, \quad \forall j \in \{1, \ldots, 10\},$$

where $\bar{\varepsilon}_j \sim \mathcal{N}(0, 0.025j)$, $\omega \sim \mathcal{N}(0, 0.02)$, $\theta \in \{0.5, 1, 2\}$, $\dim(X) \in \{10, 100\}$

- Fit linear model with OLS/Lasso regression (even when $\theta \neq 1$)

$$Y_j = \nu_j + \sum_{l=1}^{\dim(X)} \mu_{jl} X_l + \eta_j, \quad \forall j \in \{1, \ldots, 10\},$$

where η_j are zero-mean errors

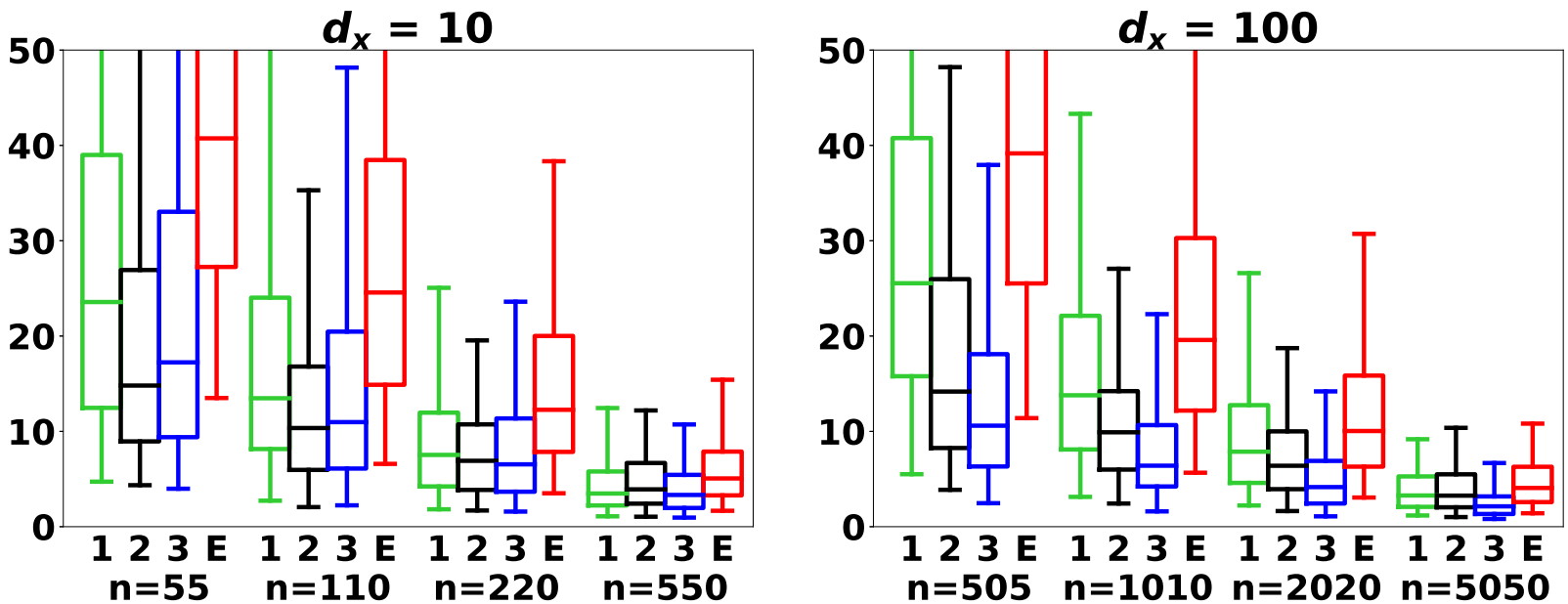
- Estimate optimality gap of solutions $\hat{z}_n^{ER}(x)$ and $\hat{z}_n^{DRO}(x)$

Results with OLS and Correct Model Class ($\theta = 1$)

E: ER-SAA + OLS

1, **2** & **3**: Wasserstein radius specified using Approaches 1, 2 & 3

Lower y-axis value $\implies$ closer to optimal



Boxes: 25, 50, and 75 percentiles of 99% upper confidence bounds

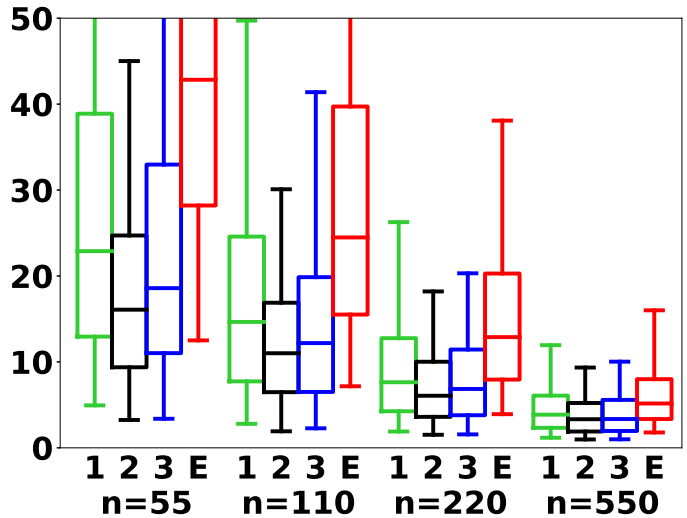
Whiskers: 5 and 95 percentiles

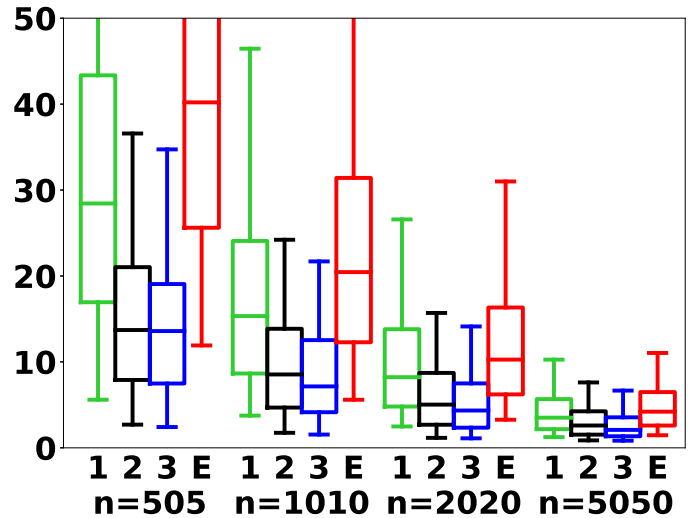
Sample sizes: $\{5, 10, 20, 50\} \times (\dim(X) + 1)$

Results with OLS and Misspecified Model Class ($\theta \neq 1$)

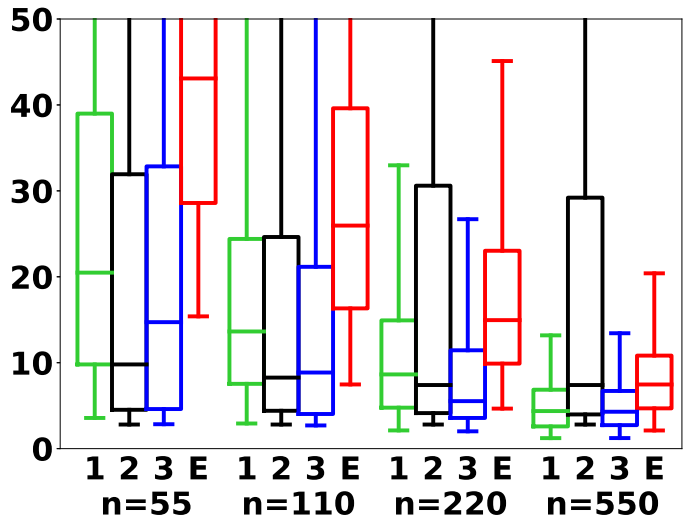
$\mathbf{d_x = 10}$ $\mathbf{d_x = 100}$

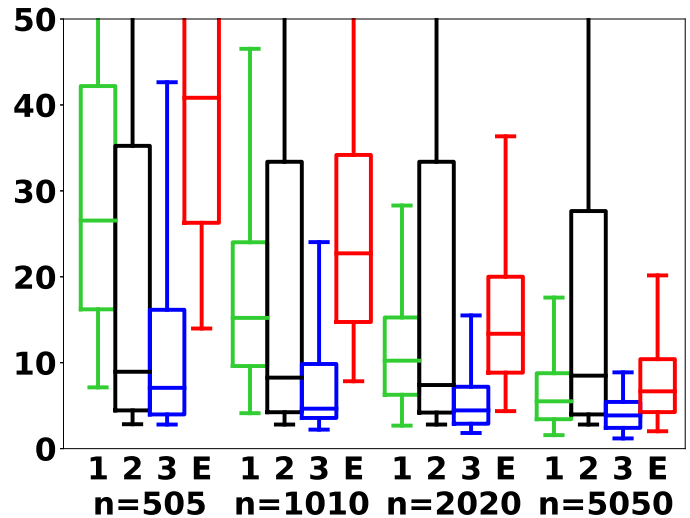
$\theta = 0.5$





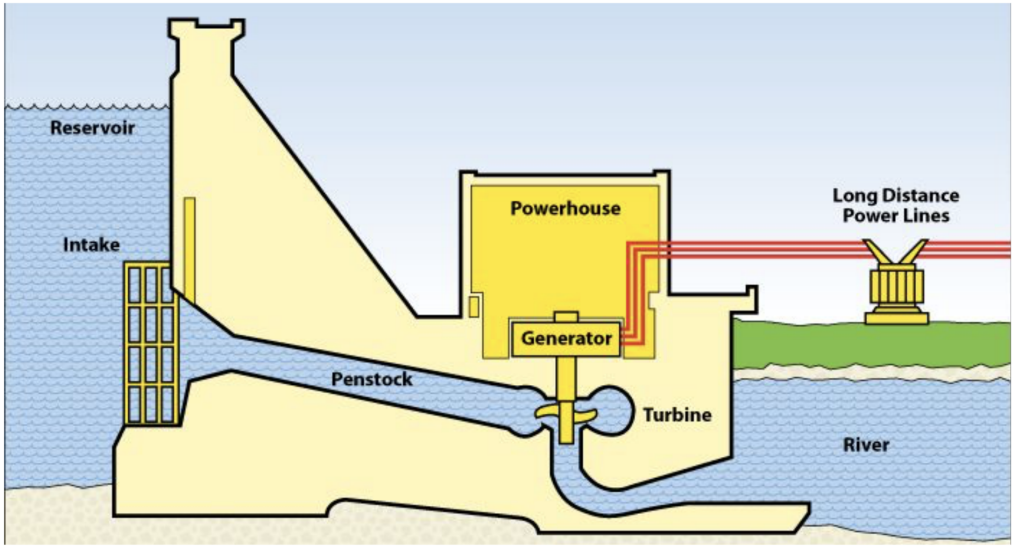
$\theta = 2$





Data-Driven Multistage Stochastic Optimization on Time Series

Numerical Study: Hydrothermal Scheduling



$$\min \sum_{t} \text{generation \& spillage costs at time } t$$

s.t. at each time stage t :

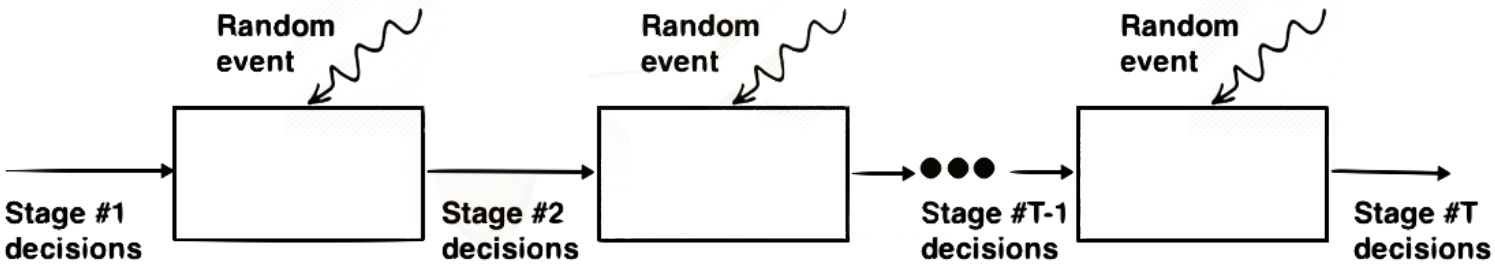
reservoir volume increase = rainfall - generation

thermal + hydro generation = demand

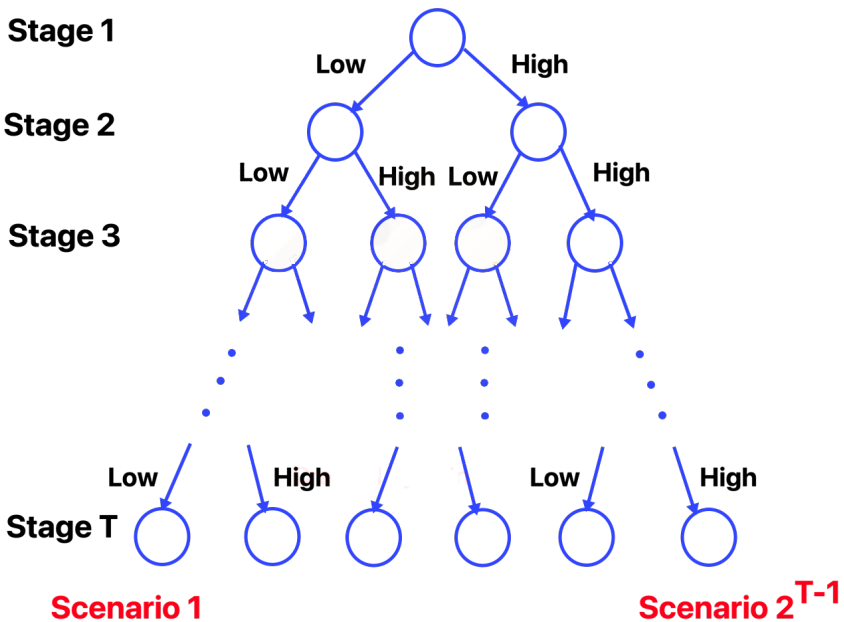
bounds on reservoir height, generation amounts

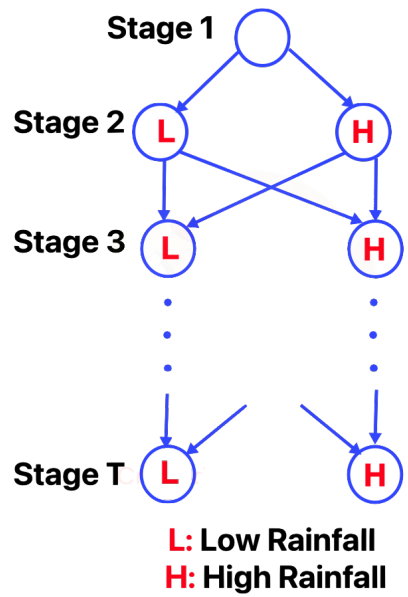
- Uncertain rainfall at each time stage t

Multistage Stochastic Optimization



Complexity of multi-stage stochastic programs can grow significantly with the number of stages T!





Multistage Stochastic Optimization

Consider the multistage stochastic program

$$V_t(x_{t-1}, \xi_{[t]}) := \min_{x_t \in X_t(x_{t-1}, \xi_t)} f_t(x_t, \xi_t) + \mathbb{E}\left[V_{t+1}(x_t, \xi_{[t+1]}) \mid \xi_{[t]}\right], \quad t \in [T-1],$$
$$V_T(x_{T-1}, \xi_{[T]}) := \min_{x_T \in X_T(x_{T-1}, \xi_T)} f_T(x_T, \xi_T) \qquad \text{(MSSP)}$$

- Decision Process: $\xi_1 \rightsquigarrow x_1 \rightsquigarrow \xi_2 \rightsquigarrow x_2 \rightsquigarrow \cdots \xi_T \rightsquigarrow x_T$
- Time Series: $\xi_{[t]} := (\xi_1, \xi_2, \ldots, \xi_t)$, where $\{\xi_t\}$ is a stochastic process satisfying

$$\xi_t = m_t^*(\xi_{t-1}, \varepsilon_t), \quad \forall t \in \mathbb{Z}$$

We deal with multi-stage stochastic LPs, where

- $f_t(x_t, \xi_t) := c_t^\top x_t$
- $X_t(x_{t-1}, \xi_t) := \left\{x_t \in \mathbb{R}_+^{n_t} : B_t(\xi_t)x_{t-1} + A_t x_t = h_t(\xi_t)\right\}$

K., Ho-Nguyen, and Luedtke. Data-driven multistage stochastic optimization on time series. Working Paper

Problem Setup

- Given historical data from a *single trajectory* of $\{\xi_t\}$

$$\mathcal{D}_n := \left\{\tilde{\xi}_s, \tilde{\xi}_{s+1}, \cdots, \tilde{\xi}_{s+n}\right\}$$

- Want to solve

$$V_1(x_0, \xi_1) := \min_{x_1 \in X_1(x_0, \xi_1)} f_1(x_1, \xi_1) + \mathbb{E}\left[V_2(x_1, \xi_2) \mid \xi_1\right],$$

where

$$V_t(x_{t-1}, \xi_t) := \min_{x_t \in X_t(x_{t-1}, \xi_t)} f_t(x_t, \xi_t) + \mathbb{E}\left[V_{t+1}(x_t, \xi_{t+1}) \mid \xi_t\right], \quad t \in [T-1],$$
$$V_T(x_{T-1}, \xi_T) := \min_{x_T \in X_T(x_{T-1}, \xi_T)} f_T(x_T, \xi_T).$$

- Assume
  - True model: $\xi_t = f^*(\xi_{t-1}) + Q^*(\xi_{t-1})\varepsilon_t$ with i.i.d. errors $\{\varepsilon_t\}$
  - We know function classes $\mathcal{F}$, $\mathcal{Q}$ such that $f^* \in \mathcal{F}$, $Q^* \in \mathcal{Q}$

Empirical Residuals-based Sample Average Approximation

Extension of the two-stage approach

1. Estimate f^*, Q^* using our favorite ML method $\Rightarrow \hat{f}_n, \hat{Q}_n$

   Compute *empirical residuals*

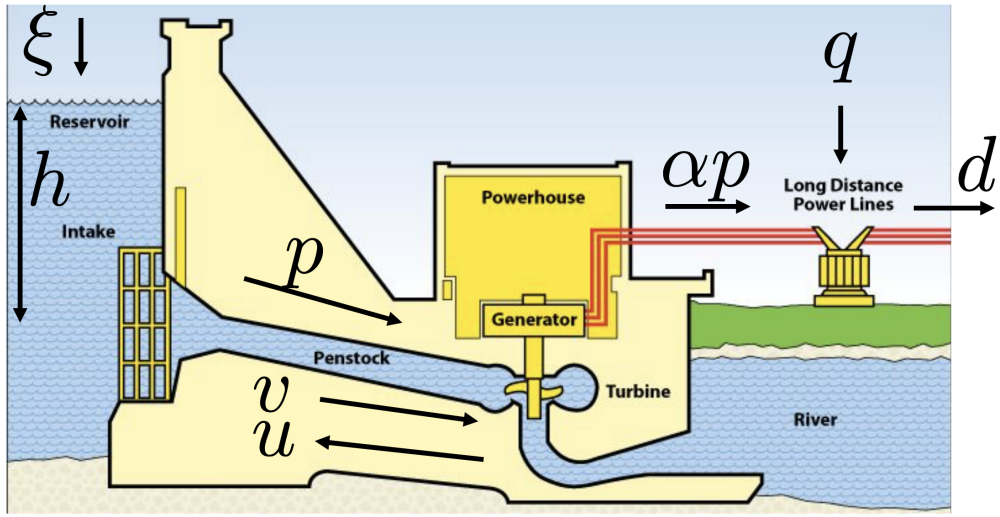
$$\hat{\varepsilon}_n^i := [\hat{Q}_n(\tilde{\xi}_{s+i-1})]^{-1}\big(\tilde{\xi}_{s+i} - \hat{f}_n(\tilde{\xi}_{s+i-1})\big), \quad i \in [n]$$

2. Use $\{\hat{f}_n(\xi_t) + \hat{Q}_n(\xi_t)\hat{\varepsilon}_n^i\}_{i=1}^n$ as proxy for samples of ξ_{t+1} given ξ_t

$$\hat{V}_{t,n}^{ER}(x_{t-1}, \xi_t) := \min_{x_t \in X_t(x_{t-1}, \xi_t)} f_t(x_t, \xi_t) + \frac{1}{n}\sum_{j \in [n]} \hat{V}_{t+1,n}^{ER}(x_t, \hat{f}_n(\xi_t) + \hat{Q}_n(\xi_t)\hat{\varepsilon}_n^i)$$

- Modular like traditional approach
- Only require a single trajectory of $\{\xi_t\}$
- Tailored convergence analysis required since *same empirical errors used* in each time stage

Numerical Experiments: Hydrothermal Scheduling



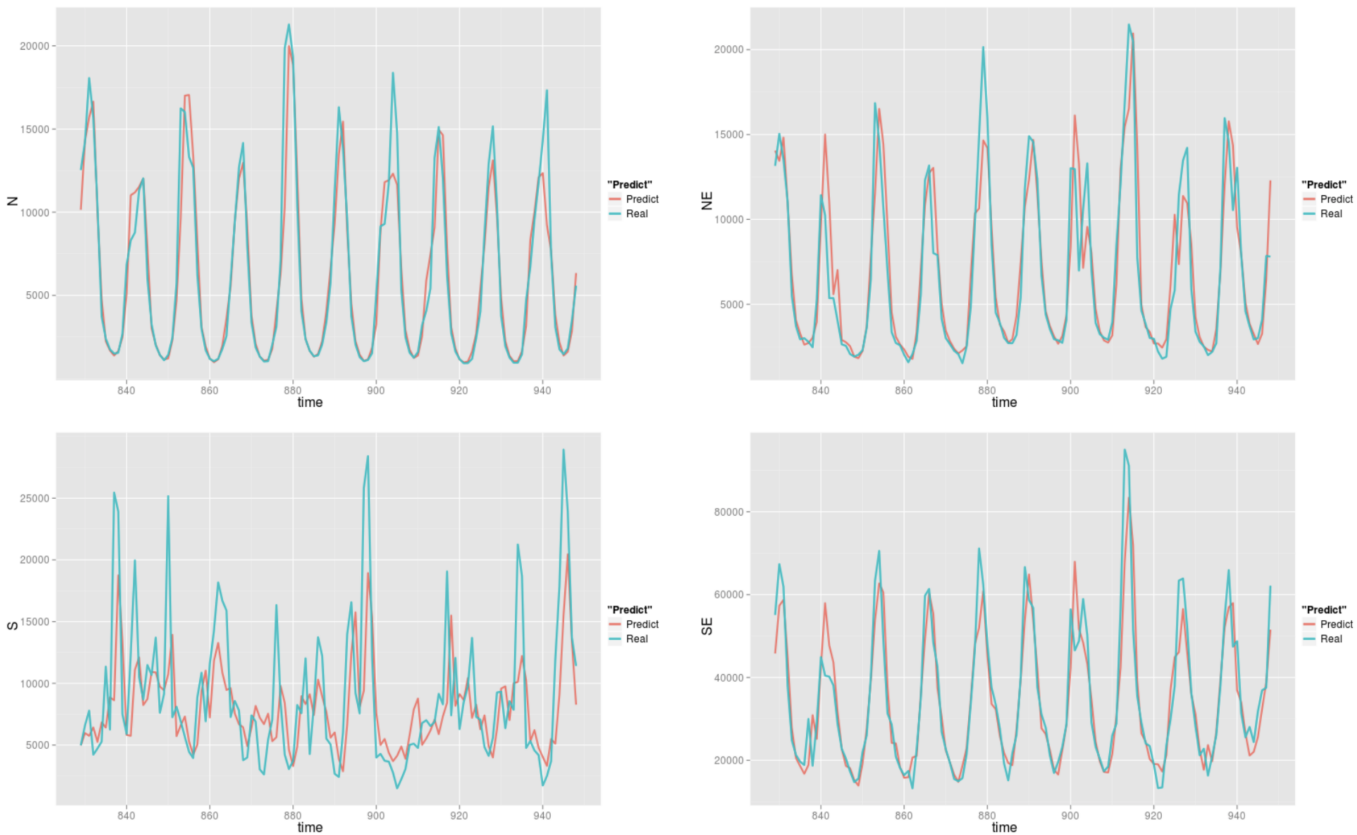
- Decisions z_t: Hydrothermal & natural gas generation, spillage
- Random vector ξ: Amount of rainfall

Numerical Experiments: Hydrothermal Scheduling

Assume true time series model for rainfall is of the form

$$\xi_t = (\alpha_t^* + \beta_t^* \xi_{t-1}) \exp(\varepsilon_t),$$

where $\alpha_t^* = \alpha_{t+12}^*$, $\beta_t^* = \beta_{t+12}^*$, $\varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \Sigma)$



Good fit to historical data over 8 decades!

Numerical Experiments: Hydrothermal Scheduling

- Consider the Brazilian interconnected power system with four hydrothermal reservoirs
- Generate a sample trajectory of $\{\xi_t\}$ using time series model

$$\xi_t = (\alpha_t^* + \beta_t^* \xi_{t-1}) \exp(\varepsilon_t),$$

where $\alpha_t^* = \alpha_{t+12}^*, \quad \beta_t^* = \beta_{t+12}^*, \quad \varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \Sigma)$

- Estimate coefficients $(\hat{\alpha}_t, \hat{\beta}_t)$ such that

$$\hat{\alpha}_t = \hat{\alpha}_{t+12}, \quad \hat{\beta}_t = \hat{\beta}_{t+12}$$

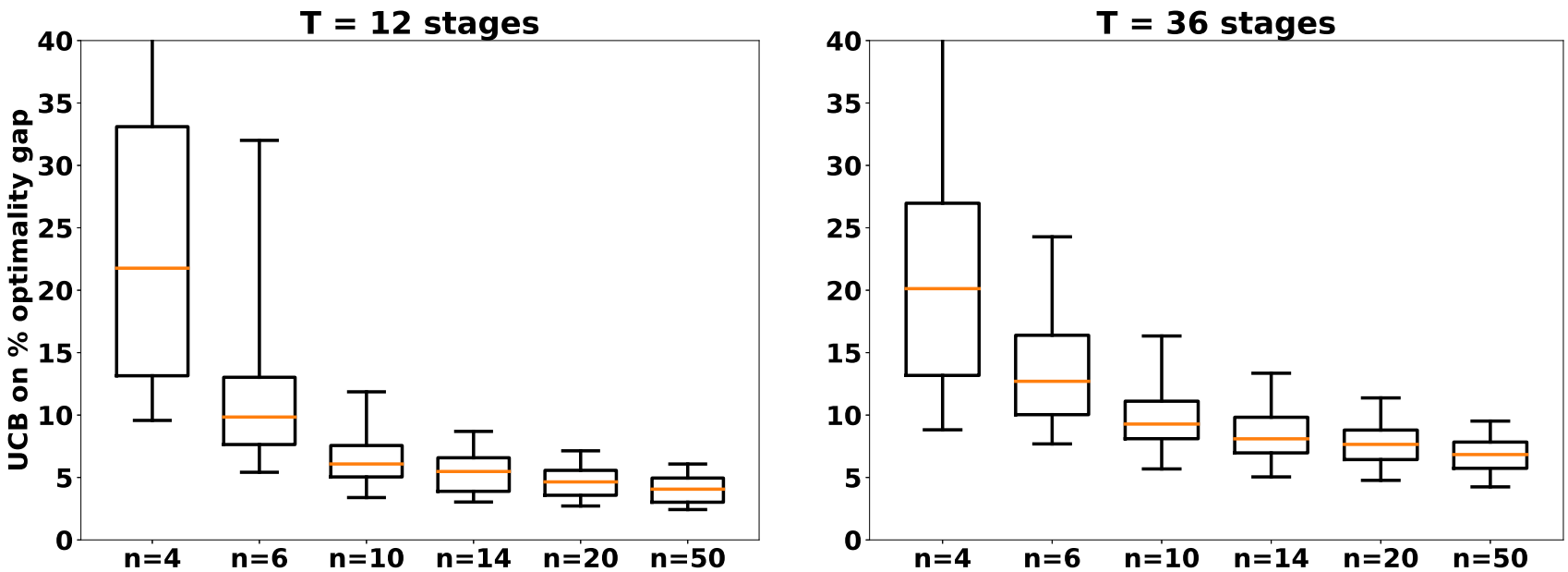
Use these to estimate samples of the errors ε_t

- Solve the ER-SAA model using `SDDP.jl`.
Estimate sub-optimality of ER-SAA solutions

Results when the time series model is correctly specified

Estimate true heteroscedastic model: $\xi_t = (\alpha_t^* + \beta_t^* \xi_{t-1}) \exp(\varepsilon_t)$

Lower y-axis value $\Longrightarrow$ closer to optimal



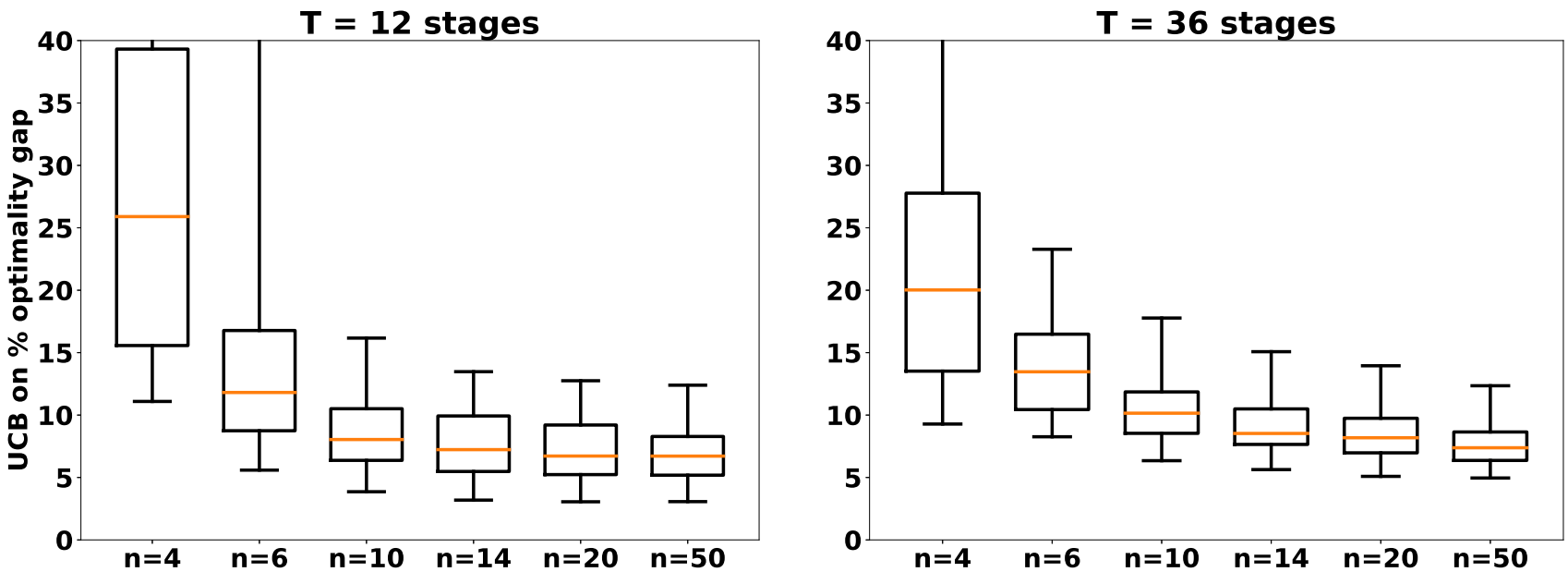
n: number of historical samples *per month*

Boxes: 25, 50, and 75 percentiles of optimality gap estimates;
Whiskers: 5 and 95 percentiles

Results when the time series model is misspecified

Estimate seasonal additive error model: $\xi_t = \alpha_t^* + \beta_t^* \xi_{t-1} + \varepsilon_t$

Lower y-axis value $\Longrightarrow$ closer to optimal



n: number of historical samples *per month*

Boxes: 25, 50, and 75 percentiles of optimality gap estimates;
Whiskers: 5 and 95 percentiles

Using ML to Accelerate Global Optimization

Using ML to Accelerate Partitioning Algorithms

Input: underlying problem, distribution of parameters θ

Output: ML model that predicts partitioning points given $\bar{\theta}$

- Generate 1000 training samples $\{\theta^i\}$ of problem parameters θ
- Solve max-min problem to determine "optimal" partitioning points for each training instance
- Learn an ML model $\theta^i \mapsto$ `optimal partitioning points`
- Use ML model to predict partitioning points for new instance $\bar{\theta}$

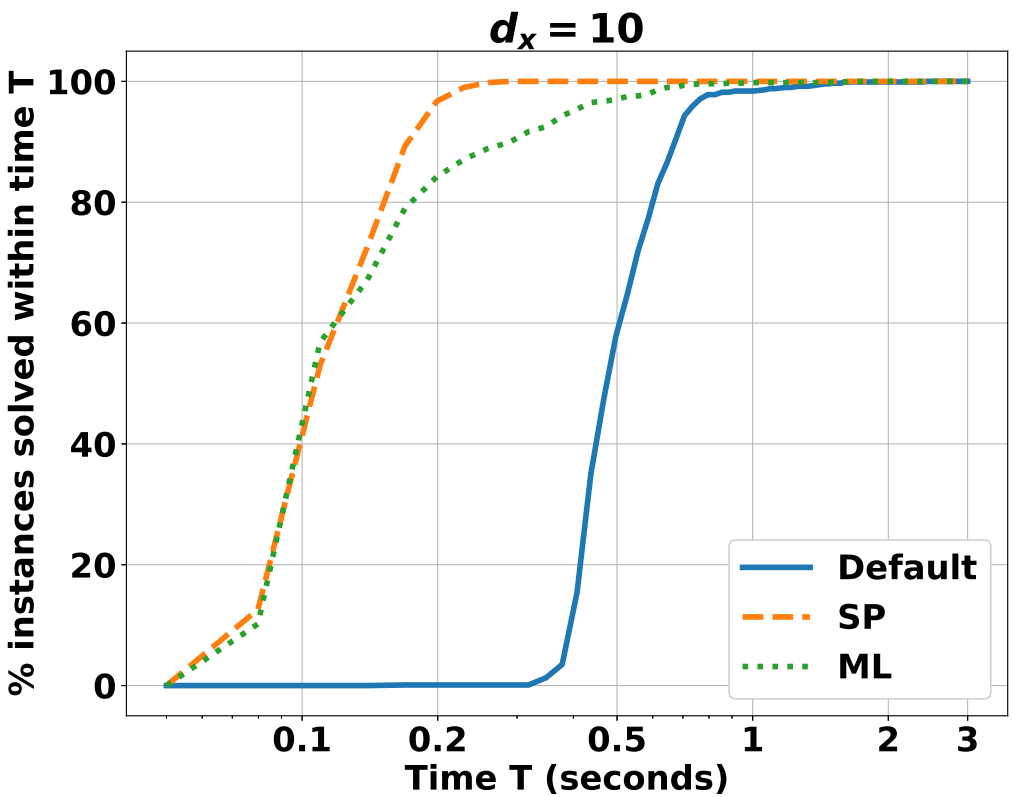
Use Scikit-learn's AdaBoostRegressor to train Regression Trees with max_depth $= 25$, num_estimators $= 1000$ (no tuning!)

- Features for training and prediction:
  - Parameter θ
  - Best found feasible solution during presolve (one local solve)
  - McCormick lower bounding solution (no partitioning)
- Use 10-fold cross validation to generate predictions for $\{\theta^i\}$

Numerical Results for Random QCQPs

Results for $d_x = 10$ variables

- Generate 1000 random QCQPs with varying parameters θ
- For each instance, determine 2 optimal partitioning points per variable by solving a max-min problem
- Eliminate optimal partitioning points that aren't useful



Speedup/ Slowdown	% SP Inst.	% ML Inst.
$1x - 2x$	1.1	7.7
$2x - 3x$	10.2	11.4
$3x - 5x$	47.4	38.5
$5x - 10x$	40.1	40.0
$> 10x$	1.2	0.1
$0.5x - 1x$	—	2.1
$< 0.5x$	—	0.2

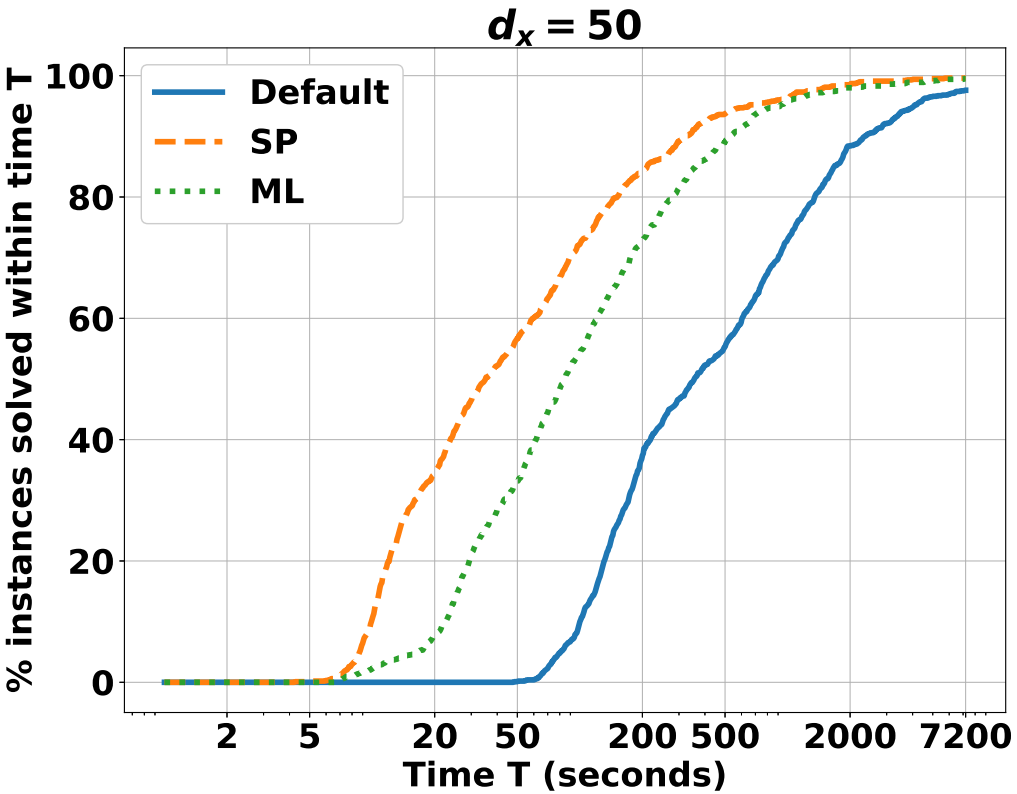
Average Speedup (Shifted GM):

Alpine+SP: $4.5x$, Alpine+ML: $3.5x$

Numerical Results for Random QCQPs

Results for $d_x = 50$ variables

- Generate 1000 random QCQPs with varying parameters θ
- 2 partitioning points per variable for each instance
- Eliminate partitioning points that aren't useful



Speedup/ Slowdown	% SP Inst.	% ML Inst.
$1x - 5x$	25.7	49.3
$5x - 10x$	26.3	25.3
$10x - 20x$	24.3	13.7
$20x - 50x$	14.9	5.4
$> 50x$	6.9	0.8
$0.5x - 1x$	1.5	4.8
$< 0.5x$	0.4	0.7

Average Speedup (Shifted GM):

Alpine+SP: $8.1x$, Alpine+ML: $4.2x$