Recent Developments on Optimization Algorithms and Applications

JUNE 21, 2023

Yinyu Ye
Stanford University and CUHKSZ (Sabbatical Leave)

Today's Talk

I. **Homogeneous Second-Order Descent Direction: An Alternative to the Newton Step**

II. A Dimension-Reduced Trust-Region Method

III. Pre-Trained Statistical Cut Generation for Mixed-Integer Linear Programming Solvers

Early Complexity Analyses for Nonconvex Optimization

- $\min f(x), x \in X \; in \; \mathbb{R}^n$,

  where f is nonconvex and twice-differentiable,

  $g_k = \nabla f(x_k), H_k = \nabla^2 f(x_k)$

- Goal: find x_k such that:

  $\| \nabla f(x_k) \| \leq \epsilon$ (primary, first-order condition)

  $\lambda_{min}(H_k) \geq -\sqrt{\epsilon}$ (in active subspace, second-order condition)

- For the ball-constrained nonconvex QP: $\min \; c^T x + 0.5 x^T Q x \; s.t. \; \| x \|_2 \leq 1$

  O(loglog(ϵ^{-1})); see Vavasis&Zippel (1990), Y (1989,93).

- For nonconvex QP with polyhedral constraints: O(ϵ^{-1}); see Y (1998), Vavasis (2001)

Second-order Methods for General Optimization

SOM (Hessian-Type Methods) with M-Lipschitz cont. Hessian

- Trust-region (More 70, Sorenson 80). Fixed-radius TR $O\left(\epsilon^{-\frac{3}{2}}\right)$, see the lecture notes by Y since 2005
- Cubic regularization, $O(\epsilon^{-3/2})$,see Nesterov and Polyak (2006), Cartis, Gould, and Toint (2011)
- An adaptive trust-region framework, $O(\epsilon^{-3/2})$,Curtis, Robinson, and Samadi (2017)

SOM for convex functions

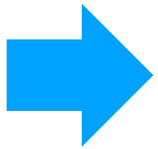
- Cubic regularization, $O(\epsilon^{-1/2})$,see Nesterov and Polyak (2006),
- Accelerated SOMs, $O(\epsilon^{-1/3})$, $O(\epsilon^{-1/3.5})$, see Monteiro and Svaitor (2013), Nesterov (2008), Doikov et al. (2022)
- Linearly convergent SOMs, self-concordance, see Nesterov and Nemirovskii (1994); scaled Lipschitz, see Kortanek and Zhu (1993), Anderson and Ye (1998); generalized concordance, see Sun (2019).

Disadvantage: each iteration requires $O(n^3)$ operations: **How to reduce it?**

An Integrated Descent Direction Using the SDP Homogeneous Model I (Zhang at al. SHUFE, 2022)

- **Recall the fixed-radius trust-region method minimizes the Taylor quadratic model**

$$\min_{d\in\mathbb{R}^n} m_k(d) := g_k^T d + \frac{1}{2} d^T H_k d \quad \text{s.t.} \|d\| \le \Delta_k .$$



$$\min_{[d,t]\in\mathbb{R}^{n+1}} m_k(d) := t \cdot g_k^T d + \frac{1}{2} d^T H_k d + \frac{1}{2}\delta \cdot (1-t^2) \quad \text{s.t.} \ \|d\|^2 + t^2 = \Delta_k^2 + 1$$

where Δ_k =c ... /M is the trust ball radius.

- **-g_k is the first-order steepest descent direction but ignores Hessian;**
- **the most-left eigenvector of H_k-would be a descent direction for the second order term**
- **Could we construct a direction integrating both?**

Answer: Use the most-left eigenvector of the SDP homogenized quadratic function!

(In Rojas 2001, a specialized Lanczos method is proposed for solving the Trust-region Subproblem with a given radius; and in Adachi 2017 for solving more Generalized Trust-region Subproblems)

An Integrated Descent Direction Using the SDP Homogeneous Model II (Zhang at al. SHUFE, 2022)

$$\psi_k(\xi_0, t; \delta) := \frac{1}{2}\begin{bmatrix}\xi_0 \\ t\end{bmatrix}^T \begin{bmatrix}H_k & g_k \\ g_k^T & -\delta\end{bmatrix}\begin{bmatrix}\xi_0 \\ t\end{bmatrix} = \frac{t^2}{2}\begin{bmatrix}\xi_0/t \\ 1\end{bmatrix}^T \begin{bmatrix}H_k & g_k \\ g_k^T & -\delta\end{bmatrix}\begin{bmatrix}\xi_0/t \\ 1\end{bmatrix}$$

- **Find the direction $\xi = \xi_0/t$ (if *t = 0* then set *t=1*) by the leftmost eigenvector:**

$$\min_{||[\xi_0;t]||\leq 1} \psi_k(\xi_0, t; \delta)$$

with a suitable $\boldsymbol{\delta}_k$ and use ξ as the direction to go – a single loop algorithm to solve the original problem and ignore the trust-region subproblem.

- **Accessible at the cost of $O\left(n^2\epsilon^{-1/4}\right)$ via the randomized Lanczos method and needs only Hessian-Vector-Product (HVP).**

How to Set δ : Theoretical Guarantees of HSODM

- **Consider using the second-order homogenized direction, and let the length of each step $\|\eta\xi\|$ be fixed: $\|\eta\xi\| \leq \Delta_k = \frac{2\sqrt{\epsilon}}{M}$, where $f(x)$ has L-Lipschitz gradient and M-Lipschitz Hessian.**

- **Theorem 1 (Global convergence rate) : Let $f(x)$ satisfy the Lipchitz Assumption and fix $\delta = \sqrt{\varepsilon}$, and let $x_{k+1} = x_k + \eta_k\xi$ where $\eta_k = \Delta_k/\|\xi\|$, then algorithm has $O(\epsilon^{-3/2})$ iteration complexity to second-order stationarity, where each iteration compute the most-left eigenvector of the homogenized matrix to ϵ accuracy.**

- **Theorem 2 (Local convergence rate): If the iterate x_k of HSODM converges to a strict local optimum x^* , HSODM possesses a local superlinear (quadratic) speed of convergence: $\| x_{k+1} - x^* \| = O(\| x_k - x^* \|^2)$.**

HSODM with Line-Search Methods

- **Fixed step length η_k may be too conservative.**
- **Observation I: homogenized direction ξ can be used with any Line-search (e.g., Hager-Zhang)**
- **Theorem 3 (Global convergence with Line-search, informal) : If we apply the backtrack to compute η_k with parameter $\beta \in (0,1)$ then the algorithm converges in $O\left(\epsilon^{-\frac{3}{2}} |\log_\beta(\epsilon)|\right)$ iterations.**

Application: HSODM for Policy Optimization in Reinforcement Learning

- Consider policy optimization of linearized objective in reinforcement learning

$$\max_{\theta \in \mathbb{R}^d} L(\theta) := L(\pi_\theta),$$

$$\theta_{k+1} = \theta_k + \alpha_k \cdot M_k \nabla \eta(\theta_k),$$

- The Natural Policy Gradient (NPG) method (Kakade, 2001) uses the Fisher information matrix where M_k is the inverse of

$$F_k(\theta) = \mathbb{E}_{\rho_{\theta_k}, \pi_{\theta_k}} \left[\nabla \log \pi_{\theta_k}(s,a) \nabla \log \pi_{\theta_k}(s,a)^T \right]$$

- Based on KL divergence, TRPO (Schulman et al. 2015) uses KL divergence in the constraint:

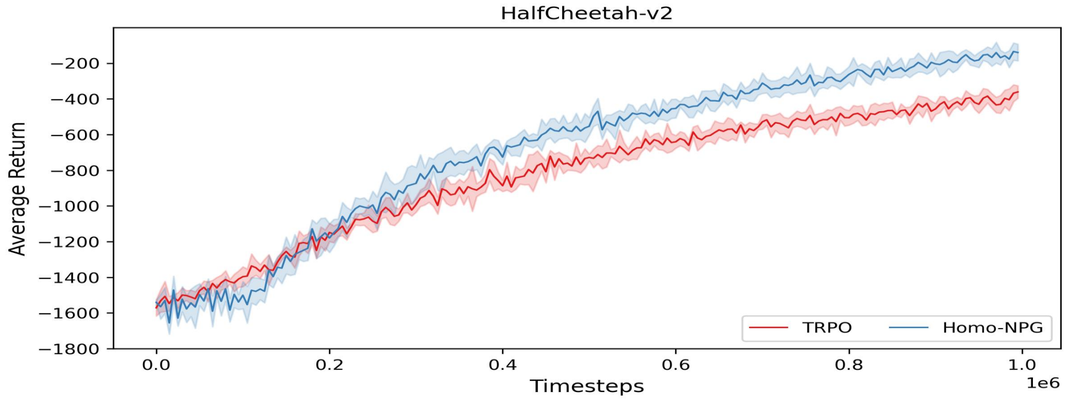
$$\max_{\theta} \nabla L_{\theta_k}(\theta_k)^T (\theta - \theta_k)$$
$$\text{s.t. } \mathbb{E}_{s \sim \rho_{\theta_k}} [D_{KL}(\pi_{\theta_k}(\cdot \mid s); \pi_\theta(\cdot \mid s))] \leq \delta.$$

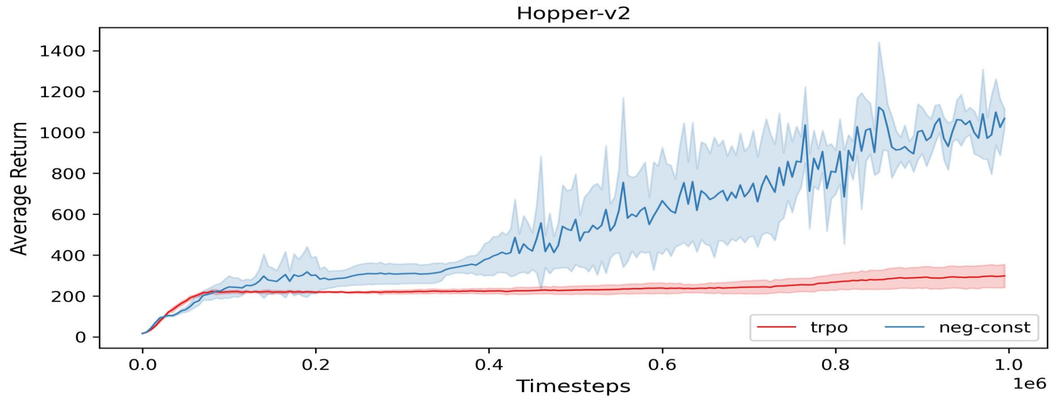
$$\min_{\|[v;t]\| \leq 1} \begin{bmatrix} v \\ t \end{bmatrix}^T \begin{bmatrix} F_k & g_k \\ g_k^T & -\delta \end{bmatrix} \begin{bmatrix} v \\ t \end{bmatrix}$$

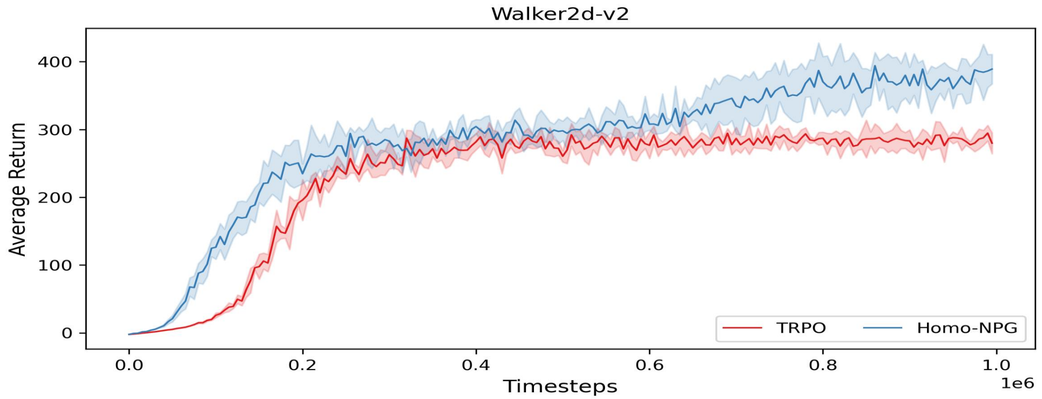
Homogeneous Natural Policy Gradient:
Apply the Homogenized Direction with Line Search!

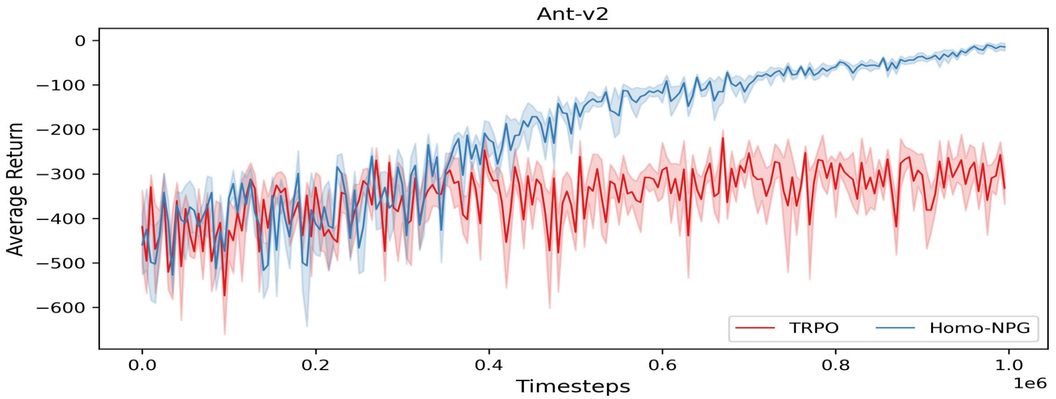
HSODM for Policy Optimization in RL II

- A comparison of Homogeneous NPG and Trust-region Policy Optimization (Schultz, 2015)



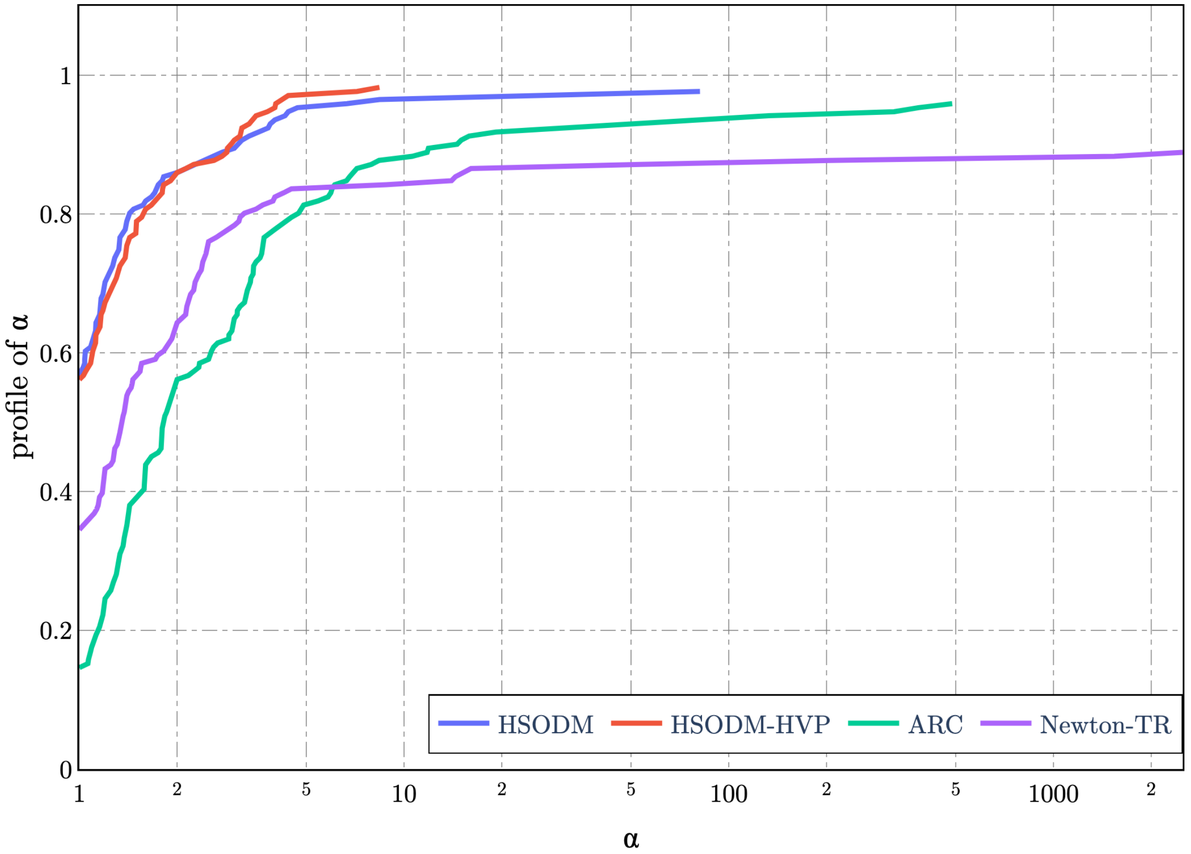






- **Homogeneous model provides significant improvements over TRPO (public open-source solver)**

Application: HSODM for CUTEst Benchmark



Performance Profile of iteration #

α – iteration # compared to the best

$profile(\alpha)$ – percentage of solved instances within α

- Compare HSODM (with Hessian), HSODM-HVP (with HVP), Newton TR and ARC
- Compare performance metrics in SGM

method	$\mathcal{K}$	$\bar{t}_G$	$\bar{k}_G$	$\bar{k}_G^f$	$\bar{k}_G^g$	$\bar{k}_G^H$
Newton-TR	155.00	15.41	216.59	211.99	219.58	203.82
HSODM	170.00	4.13	80.22	159.76	180.04	80.22
HSODM-HVP	171.00	5.25	110.61	193.07	1080.57	0.00
ARC	167.00	5.32	185.03	185.03	888.35	0.00

- K – success #, t_G - geometric mean running time (SGM), k_G - geometric mean iteration # (SGM)
- Newton-TR and ARC are public solvers

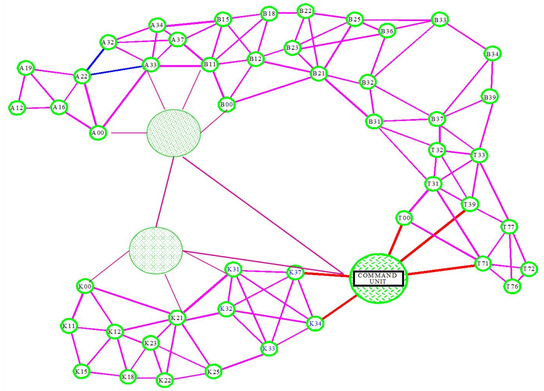
Application: HSODM for Sensor Network Localization

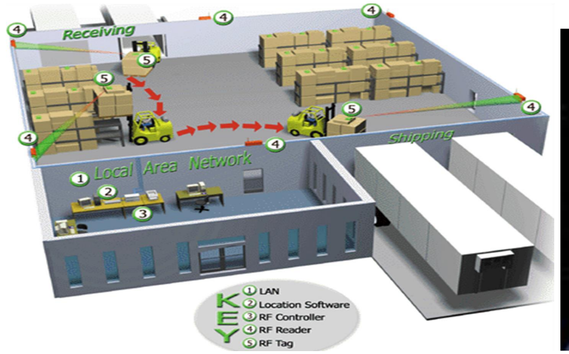
- Consider Sensor Network Location (SNL)

$$N_x = \{(i,j) : \|x_i - x_j\| = d_{ij} \leq r_d\}, N_a = \{(i,k) : \|x_i - a_k\| = d_{ik} \leq r_d\}$$

where r_d is a fixed parameter known as the radio range. The SNL problem considers the following QCQP feasibility problem,

$$\|x_i - x_j\|^2 = d_{ij}^2, \forall (i,j) \in N_x$$
$$\|x_i - a_k\|^2 = \bar{d}_{ik}^2, \forall (i,k) \in N_a$$

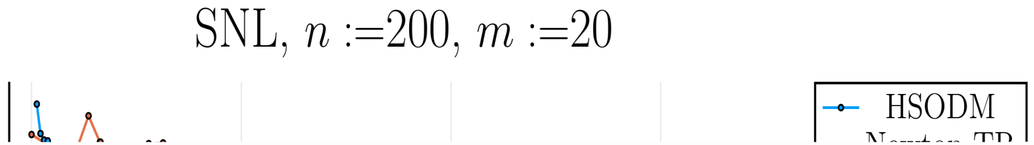




- We can solve SNL by the nonconvex nonlinear least square (NLS) problem

$$\min_X \sum_{(i<j,j)\in N_x} (\|x_i - x_j\|^2 - d_{ij}^2)^2 + \sum_{(k,j)\in N_a} (\|a_k - x_j\|^2 - \bar{d}_{kj}^2)^2.$$

Application: HSODM for Sensor Network Localization



- Compare HSODM (with HVP), and Newton-TR Method.
- HSODM is faster due to the eigenvalue procedure
- The solution quality is much better than the FOMs

Generalized Homogeneous Model (GHM) and HSODM

- **Can we equivalent HSODM to more second-order frameworks?**

- **Introduce *Generalized* Homogeneous Model (GHM)**

$$\begin{bmatrix} \boldsymbol{H_k} & \boldsymbol{g_k} \\ \boldsymbol{g_k^T} & \boldsymbol{\delta} \end{bmatrix} \Rightarrow \begin{bmatrix} \boldsymbol{H_k} & \boldsymbol{\phi_k} \\ \boldsymbol{\phi_k^T} & \boldsymbol{\delta_k} \end{bmatrix},$$

- **Adaptive δ_k and smart choice of ϕ_k (g_k suffices in most case)**

Method	Adaptive Controls		Complexity	References
	ϕ_k	δ_k		
Gradient Regularization		✓	$O(\epsilon^{-0.5})$	Mishchenko 2022, Doikov 2022
ARC	†	✓	$O(\epsilon^{-1.5})$, $O(\epsilon^{-0.5})$	Nesterov and Polyak 2006, Cartis et al. 2011
Trust-region Method	†	✓	$O(\epsilon^{-1.5})$	Curtis et al. 2017
Homotopy method (new)	✓	✓	$O(log(\epsilon^{-1}))$	Luenberger and Ye 2021 Lecture notes by Ye, 2015

Concordant Second-Order Lipschitz condition I

- **Consider** $\min_x f(x)$, **where** $f(x)$ **satisfies**

$$\left\|\nabla f(x+d) - \nabla f(x) - \nabla^2 f(x)d\right\| \leq \beta \cdot d^T \nabla^2 f(x) d$$

  whenever $\| d \| \leq O(1)$.

- **This condition is called *the concordant second-order Lipschitz condition (CSOLC),* first introduced in Luenberger & Ye (2015, 2022).**

- **CSOLC is motivated from the Scaled Lipschitz Condition, which was widely used in the IPMs and MCPs. see Zhu(1992), Kortane&Zhu(1993), Andersen&Ye(1999).**

Concordant Second-Order Lipschitz condition II

Properties of CSOLC:

- **Closed under positive scalar multiplications and summations;**
- **Closed under affine transformation: if $f(x)$ satisfies CSOLC, then $f(Ax$**

Examples of CSOLC:

- **Convex quadratic functions, exponential functions;**
- **γ -Regularized logistic regression: $f(x) = \frac{1}{m}\sum_{i=1}^{m} log\left(1 + e^{-b_i \cdot a_i^T x}\right) + \frac{\gamma}{2}|x|^2$**

The Homotopy Model

- **The homotopy model:**

$$x_{\mu_T} = \arg\min f(x) + \frac{\mu_T}{2}\|x\|^2$$

Where $\mu_T \to 0$. We say $\{X_{\mu_T}\}$ forms a central path.

- **At each iterate solve the homotopy model *inexactly (approximate "centering" condition, ACC)*:**

$$\|\nabla f(x_{T,k}) + \mu_T \cdot x_{T,k}\| \leq \frac{\mu_T}{1 + 3(\beta + 1)}.$$

- **Use GHM with proper $\boldsymbol{\delta_k}$ and $\boldsymbol{\phi_k}$!**

Homotopy HSODM I

- **For each homotopy model, we apply GHM to solve it:**

$$\min_{\|[v;t]\|\leq 1} \begin{bmatrix} v \\ t \end{bmatrix}^T \begin{bmatrix} H_{T,k} & g_{T,k} + \mu_T \cdot x_{T,k} \\ (g_{T,k} + \mu_T \cdot x_{T,k})^T & -\mu_T \end{bmatrix} \begin{bmatrix} v \\ t \end{bmatrix}$$

- **Lemma 2(a): (fixed distance from the "central" path)**

$$\|x_{T,k} - x_{\mu_T}\| \leq \frac{1}{1 + 3(\beta + 1)}$$

- **Lemma 2(b): (finite convergence for each epoch) For any μ_T, ACC can be satisfied within $K \leq 2$ steps, specifically**

$$K = \left\lceil \log_2 \left(\frac{\log(1 + 3(\beta + 1)) - \log(\beta + 1)}{\log 3 - \log 2} \right) \right\rceil$$

Homotopy HSODM II

A Non-Interior Homotopy HSODM:

- **Linearly decrease μ_T → simultaneously adaptive δ_k and ϕ_k**

$$\mu_{T+1} = \frac{1 + \|x_{T,k}\|}{1 + 3(1+\beta)(1 + \|x_{T,k}\|)} \cdot \mu_T \quad \Rightarrow \quad x_{T+1,0} := x_{T,k}$$

- **Use GHMs as each subproblem at μ_T with finite convergence**

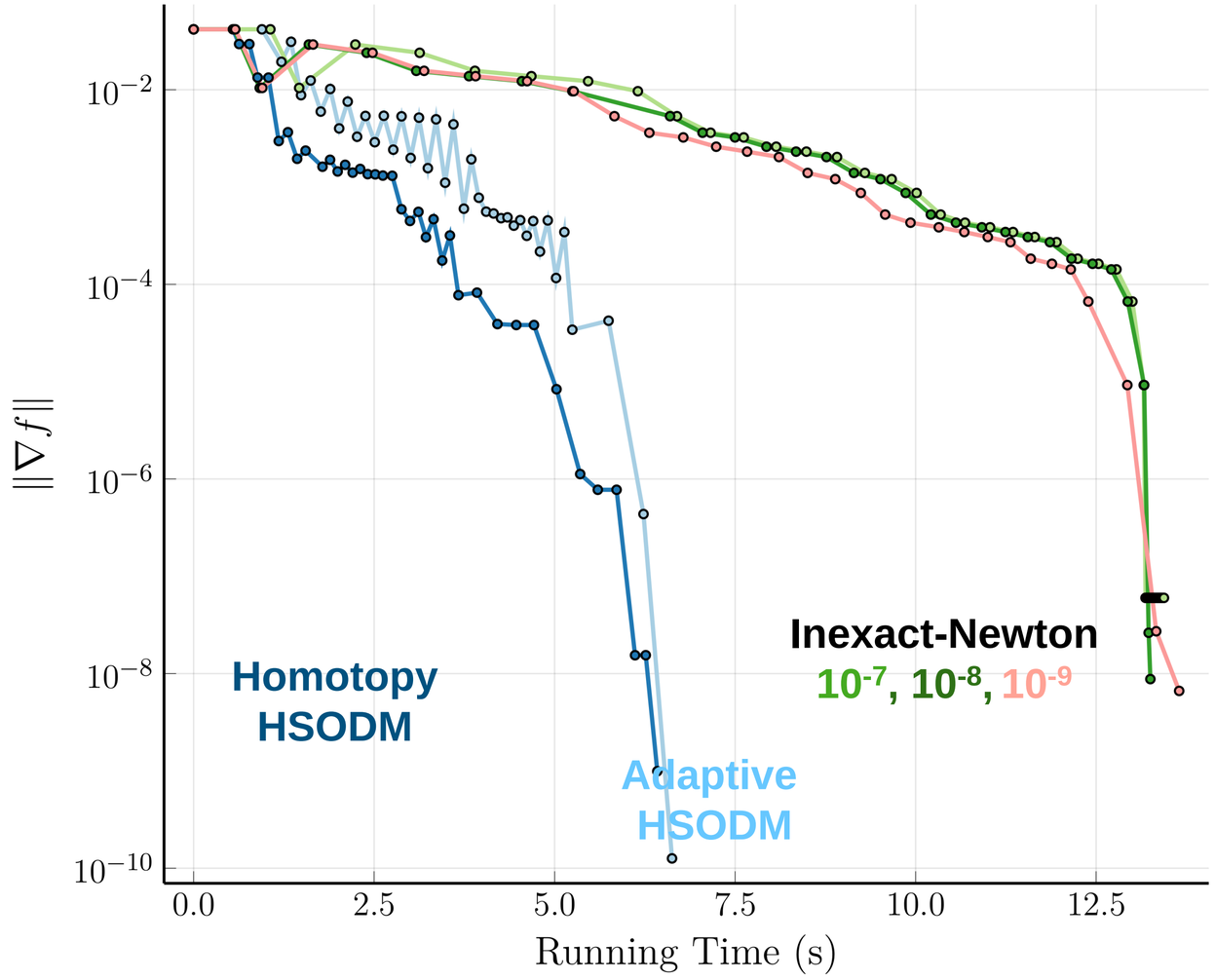
- **Theorem: (global rate of convergence) After at most**

$$\overline{T} = \left\lceil \log_\tau \left(\frac{(1 + 3(\beta+1))\epsilon}{2(\beta+1)(1 + \|\nabla f(0)\|^2)((3\beta+4)\|x^*\| + 2)} \right) \right\rceil$$

iterates, we could find an iterate that satisfies $|\nabla f\left(x_{\overline{T}+1,0}\right)| \leq \epsilon$

Application: A Comparison in L_2 - Logistic regression, $\gamma = 1e\text{-}5$

Logistic Regression `name` := `rcv1`, n :=47236, N :=20242

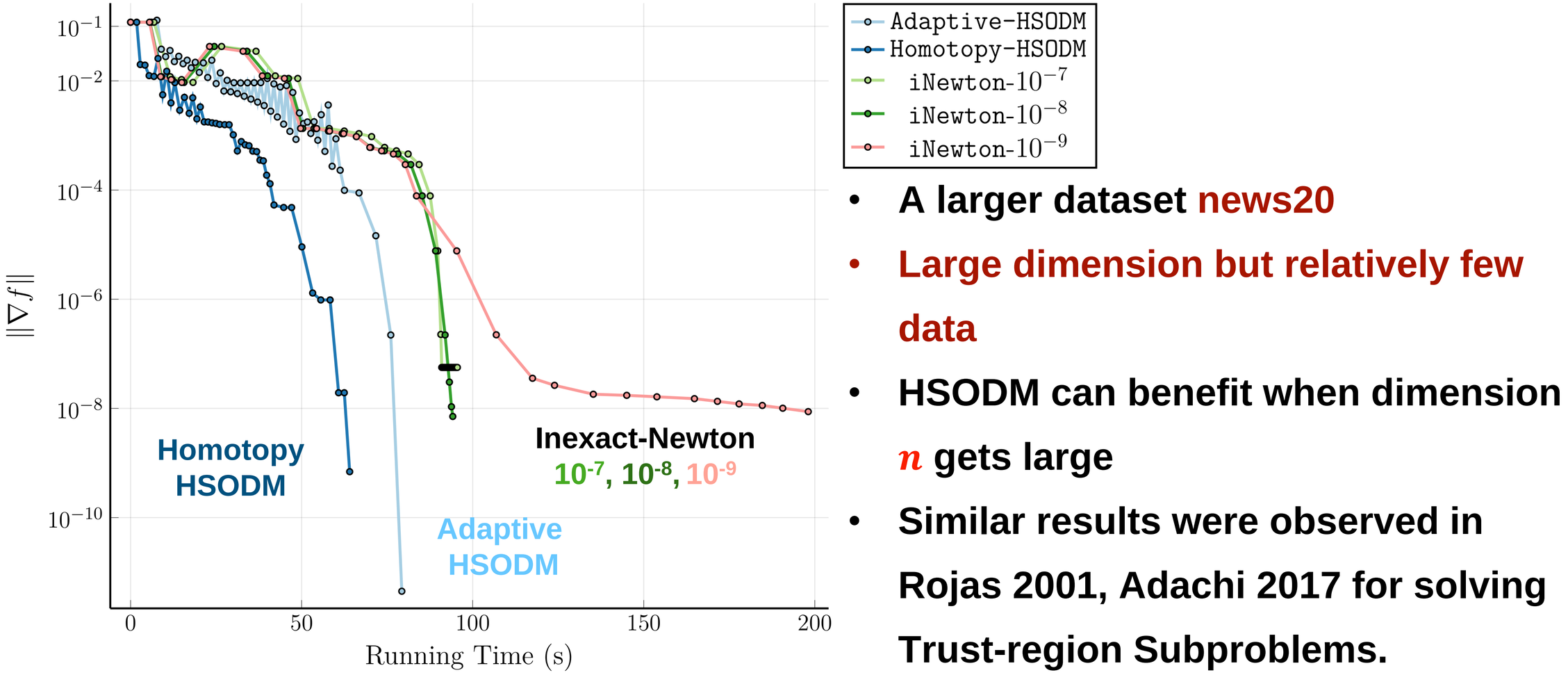


- L_2 -Logistic regression:

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} log\left(1 + e^{-b_i \cdot a_i^T x}\right) + \frac{\gamma}{2}|x|^2$$

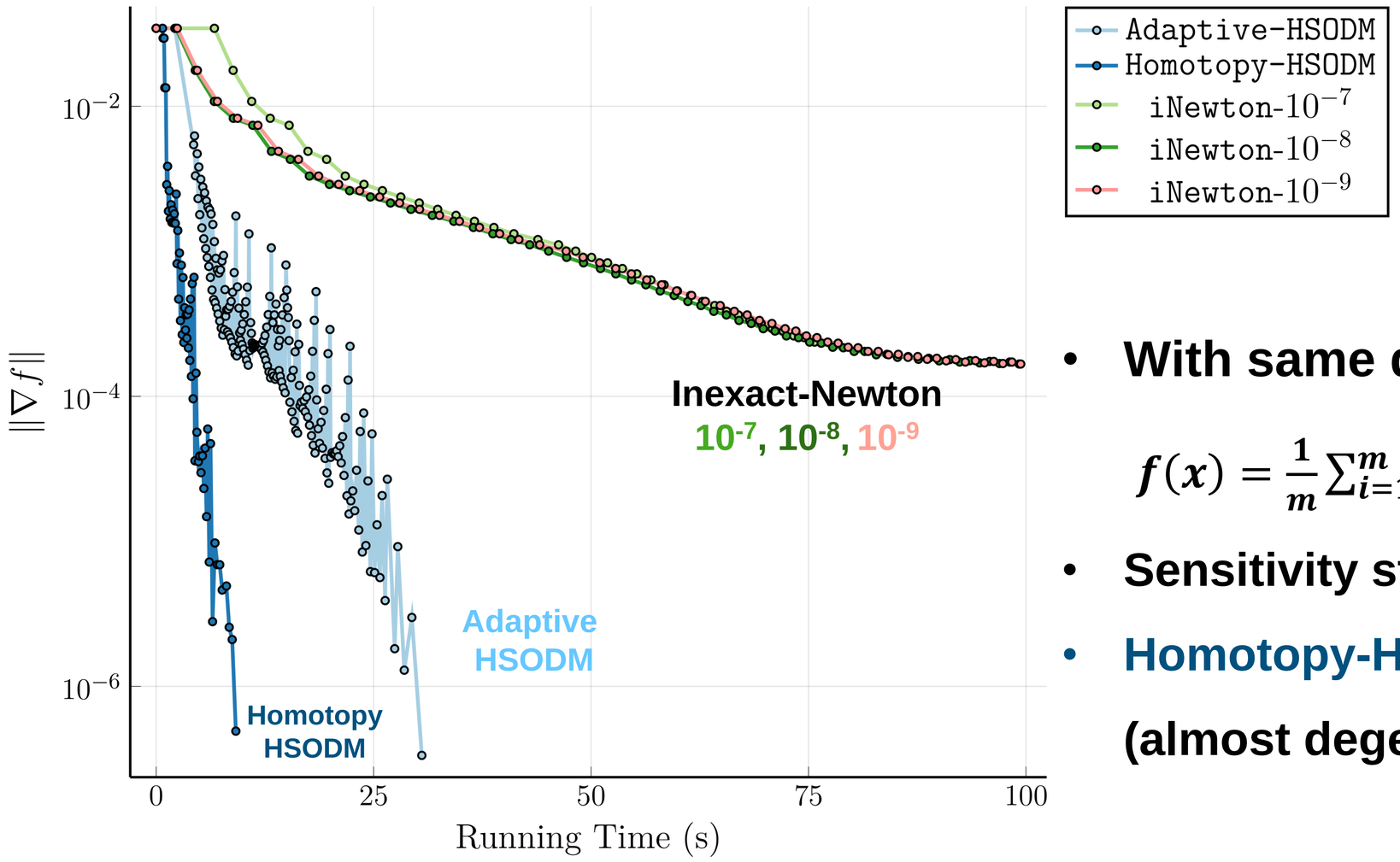
- Compare Homotopy-HSODM, Adaptive HSODM
- and inexact Newton with different accuracy (public open-source code)

A Comparison in L_2 - Logistic regression, $\gamma = 1e\text{-}5$



- **A larger dataset news20**
- **Large dimension but relatively few data**
- **HSODM can benefit when dimension n gets large**
- **Similar results were observed in Rojas 2001, Adachi 2017 for solving Trust-region Subproblems.**

Resilience of Homotopy-HSODM for small γ, $\gamma = 1e\text{-}7$

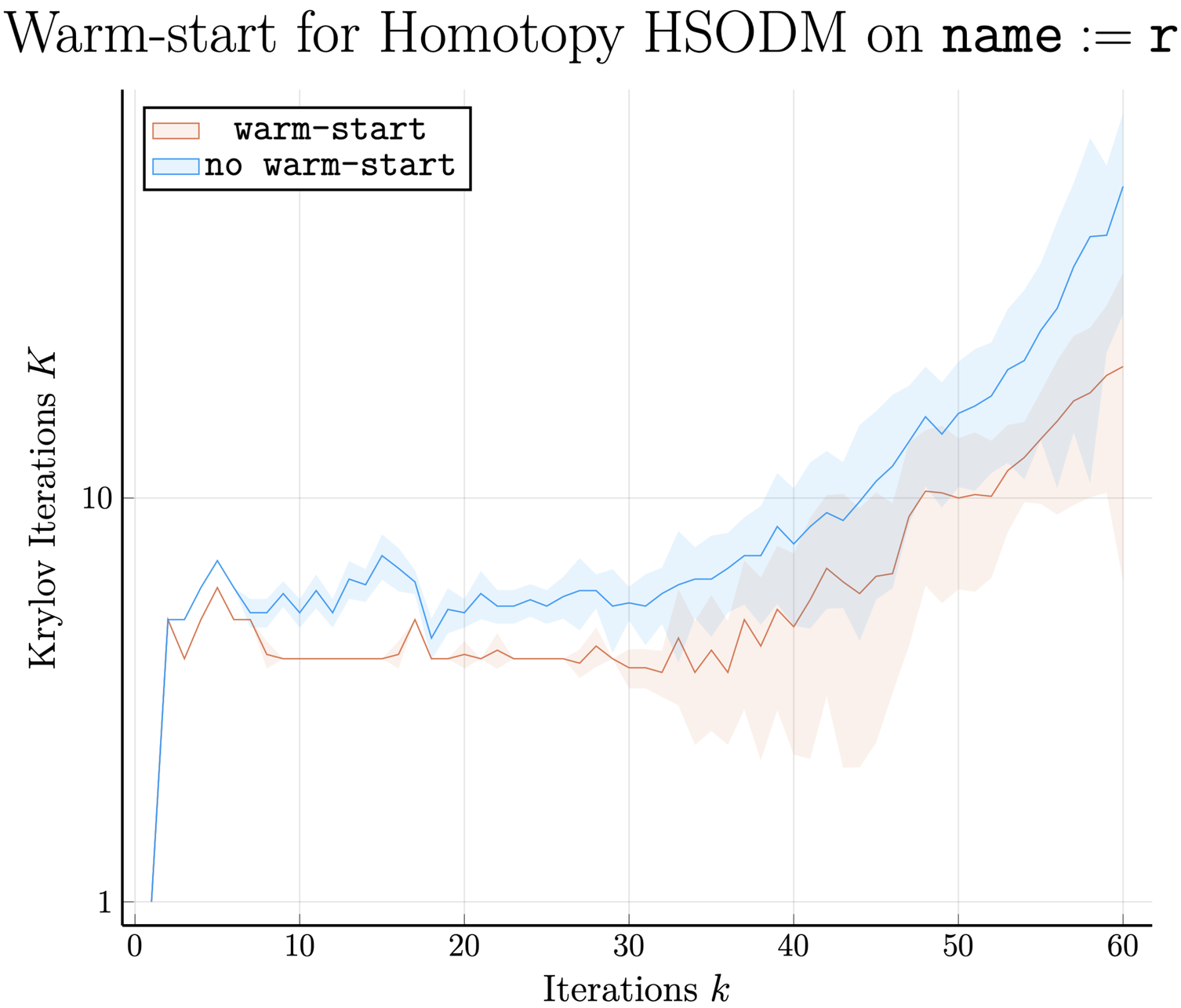


- **With same dataset rcv1**

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} log\left(1 + e^{-b_i \cdot a_i^T x}\right) + \frac{\gamma}{2}|x|^2$$

- **Sensitivity study from $\gamma = 1e\text{-}5 \rightarrow 1e\text{-}7$**
- **Homotopy-HSODM is resilient to small γ (almost degenerate case)**

Warm-starting HSODM



- **With same dataset rcv1**

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} log\left(1\right.$$

Today's Talk

I. Homogeneous Second-Order Descent Direction: An Alternative to the Newton Step

II. A Dimension-Reduced Trust-Region Method

III. Pre-Trained Statistical Cut Generation for Mixed-Integer Linear Programming Solvers

Dimension Reduced Second-Order Method (DRSOM) I

- Motivation from Multi-Directional FOM and Subspace Method, DRSOM in general uses **reduced** m-independent directions $d(\alpha) := D_k \alpha$, $D_k \in R^{nm}$, $\alpha \in R^m$
- Plug the expression into the full-dimension Trust-Region quadratic minimization model, we minimize a m-dimension trust-region subproblem to decide “m stepsizes”:

$$\min\ m_k^{\alpha}(\alpha) := (c_k)^T \alpha + \frac{1}{2}\alpha^T Q_k \alpha$$
$$||\alpha||_{G_k} \leq \Delta_k$$

$$G_k = D_k^T D_k, Q_k = D_k^T H_k D_k, c_k = (g_k)^T D_k$$

How to choose D_k? Provable complexity result?

DRSOM II

- In following, as an example, DRSOM adopts two FOM directions

$$d = -\alpha^1 \nabla f(x_k) + \alpha^2 d_k := d(\alpha)$$

where $g_k = \nabla f(x_k), H_k = \nabla^2 f\left(x^k\right), d_k = x_k - x_{k-1}$

- Then we minimize a 2-D trust-region problem to decide “two step-sizes”:

$$\min\ m_k^{\alpha}(\alpha) := f(x_k) + (c_k)^T \alpha + \frac{1}{2}\alpha^T Q_k \alpha$$

$$||\alpha||_{G_k} \leq \Delta_k$$

$$G_k = \begin{bmatrix} g_k^T g_k & -g_k^T d_k \\ -g_k^T d_k & d_k^T d_k \end{bmatrix}, Q_k = \begin{bmatrix} g_k^T H_k g_k & -g_k^T H_k d_k \\ -g_k^T H_k d_k & d_k^T H_k d_k \end{bmatrix}, c_k = \begin{bmatrix} -||g_k||^2 \\ g_k^T d_k \end{bmatrix}$$

DRSOM III

DRSOM can be seen as:

- "Adaptive" **Accelerated Gradient Method** (Polyak's momentum 60)
- A second-order method minimizing quadratic model in the reduced 2-D subspace

  $$m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d, d \in \text{span}\{-g_k, d_k\}$$

  compare to, e.g., Dogleg method, 2-D Newton **Trust-Region Method**

  $d \in \text{span}\{g_k, [H(x_k)]^{-1} g_k\}$ (e.g., Powell 70, Byrd 88)
- A conjugate direction method for convex optimization exploring the **Krylov Subspace**

  (e.g., Barzilai&Borwein 88, Yuan&Stoer 95, Yuan 2014, Liu et al. 2021)
- For convex quadratic programming with no radius limit, terminates in n steps

Computing the two-dimensional quadratic model is the Key

In the DRSOM with two directions:

$$Q_k = \begin{bmatrix} g_k^T H_k g_k & -g_k^T H_k d_k \\ -g_k^T H_k d_k & d_k^T H_k d_k \end{bmatrix}, c_k = \begin{bmatrix} -||g_k||^2 \\ g_k^T d_k \end{bmatrix}$$

How to cheaply obtain Q? Compute $H_k g_k, H_k d_k$ first.

- Finite difference:

  $$H_k \cdot v \approx \frac{1}{\epsilon}[g(x_k + \epsilon \cdot v) - g_k],$$

- Analytic approach to fit modern automatic differentiation,

  $$H_k g_k = \nabla(\frac{1}{2} g_k^T g_k), H_k d_k = \nabla(d_k^T g_k),$$

- Use Hessian if readily available !
- **Three(-or more)-Point Interpolation: it is almost as fast as Polyak and CG!**

DRSOM: key assumptions and theoretical results (Zhang at al. SHUFE, 2022)

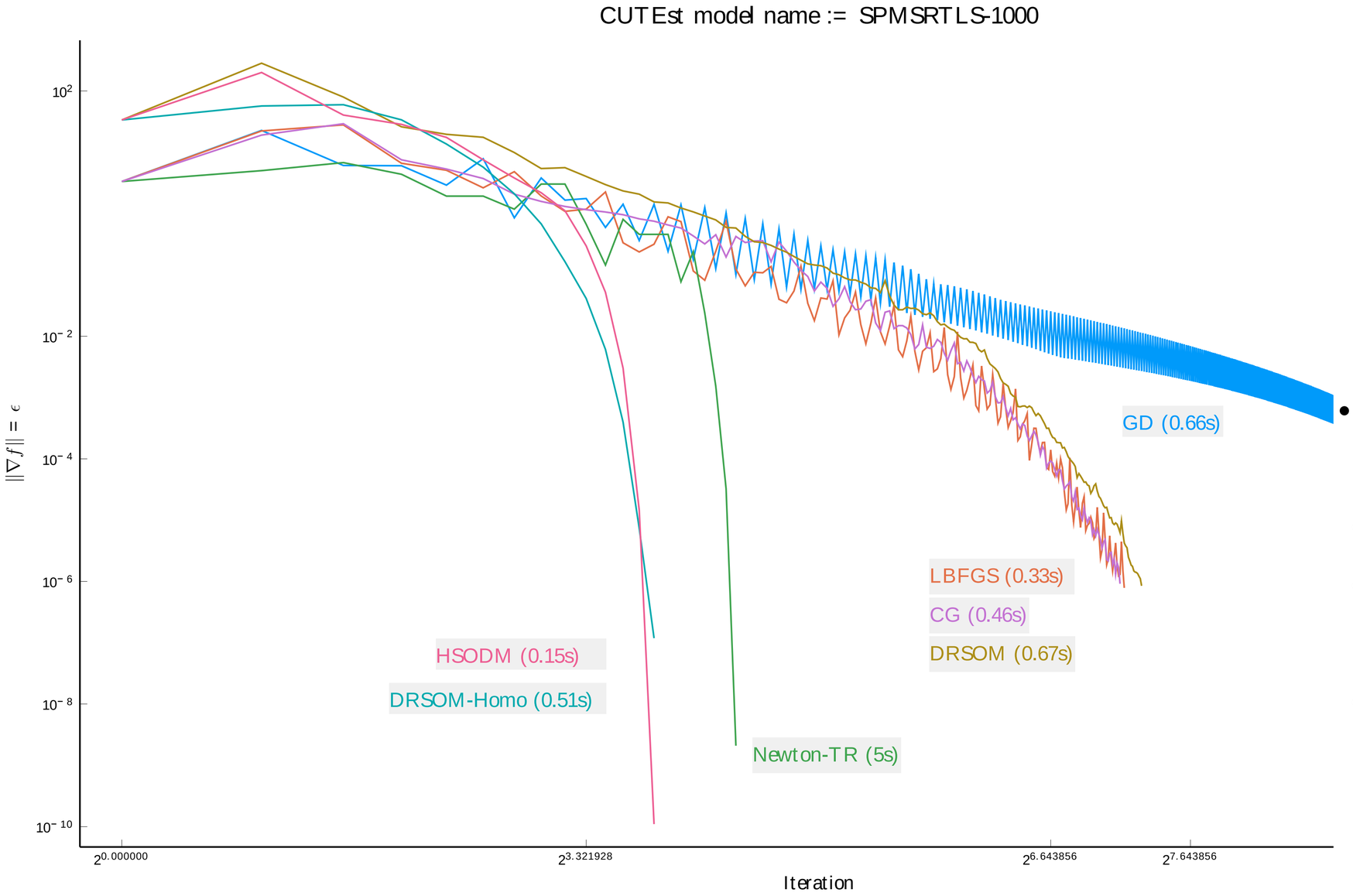
Assumption. (a) f has Lipschitz continuous Hessian. (b) **If the Lagrangian multiplier $\boldsymbol{\lambda_k} < \sqrt{\epsilon}$, assume $\| (\boldsymbol{H_k} - \widetilde{\boldsymbol{H}}_{\boldsymbol{k}})\boldsymbol{d_{k+1}} \| \leq \boldsymbol{C} \| \boldsymbol{d_{k+1}} \|^2$ (Cartis et al.),** where $\widetilde{H}_k$ is the projected Hessian in the subspace (commonly adopted for approximate Hessian)

Theorem 1. If we apply DRSOM to QP, **then** the algorithms terminates in at most n steps to find a first-order stationary point

Theorem 2. (Global convergence rate) For *f* with second-order Lipschitz condition, let $\Delta_k = 2\epsilon^{1/2}/M$, then DRSOM terminates in $O(\epsilon^{-3/2})$ iterations. Furthermore, the iterate x_k satisfies the first-order condition, and the Hessian is positive semi-definite in the subspace spanned by the gradient and momentum.

Theorem 3. (Local convergence rate) If the iterate x_k converges to a strict local optimum x^* such that $H(x^*) \succ 0$, and if **<u>Assumption (c)</u>** is satisfied as soon as $\lambda_k \leq C_\lambda \| d_{k+1} \|$, then DRSOM has a local superlinear (quadratic) speed of convergence, namely: $\| x_{k+1} - x^* \| = O(\| x_k - x^* \|^2)$

Preliminary Results: HSODM and DRSOM + HSODM

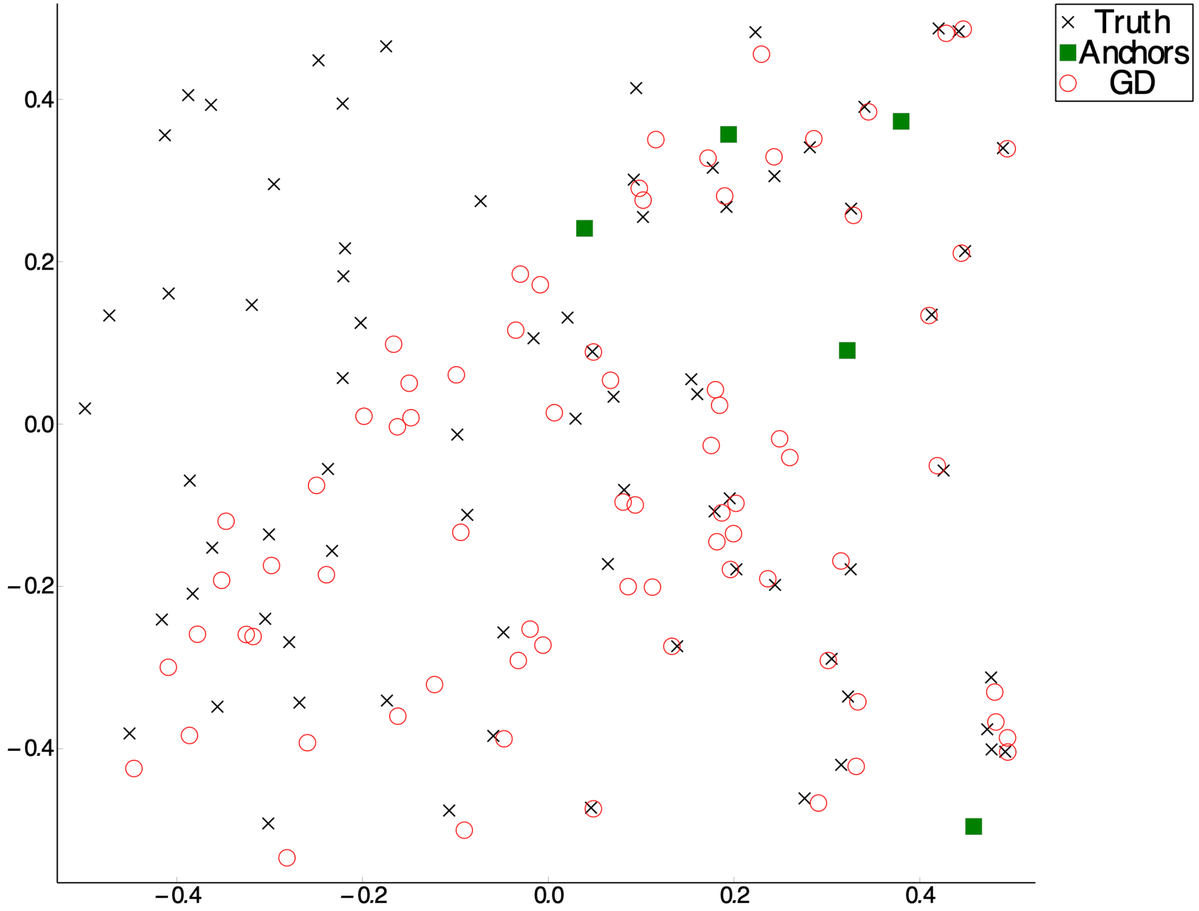


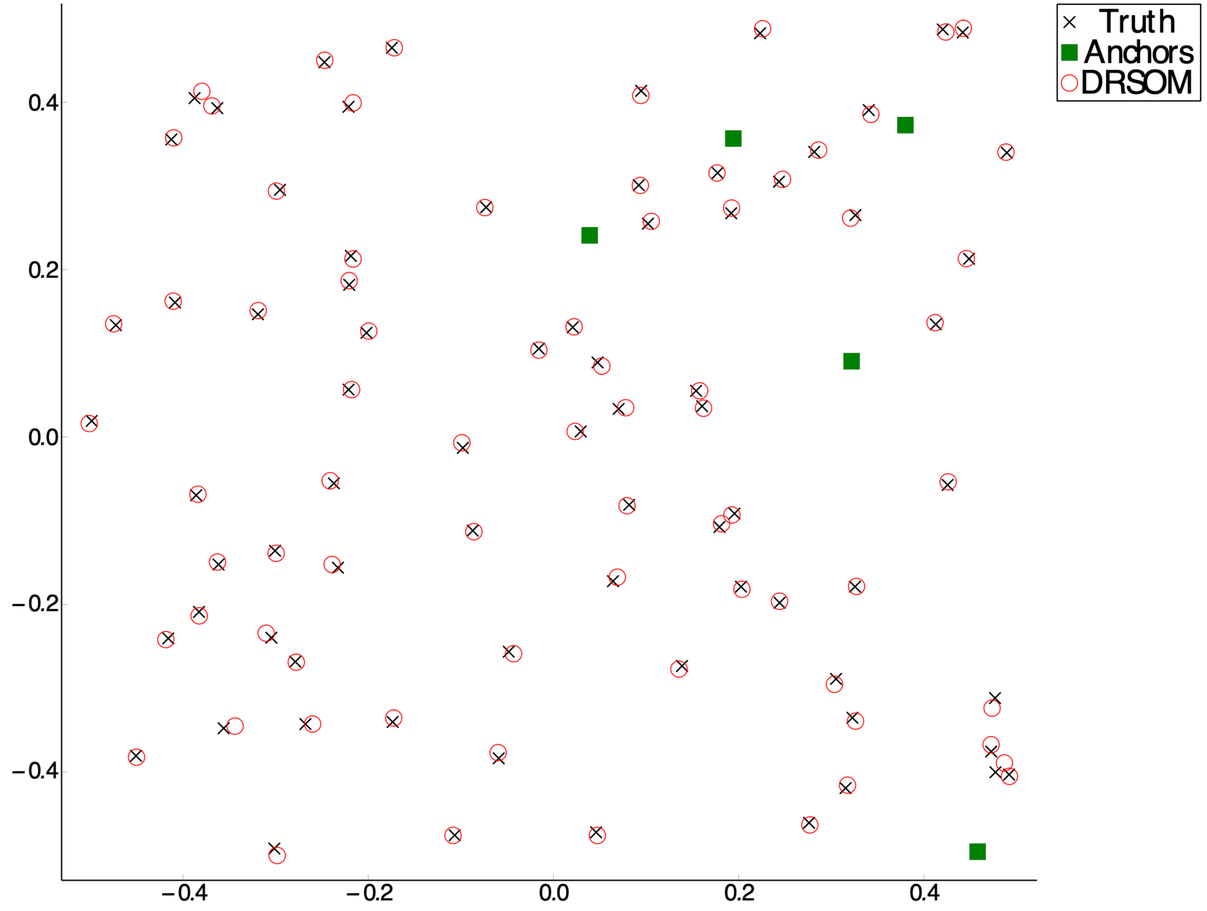
CUTEst example

- *GD* and *LBFGS* both use a Line-search (Hager-Zhang)
- *DRSOM* uses 2-D subspace
- HSODM and DRSOM + HSODM are much better!
- *DRSOM* can also benefit from the homogenized system

Sensor Network Location (SNL)

- Graphical results without SDP relaxation
- DRSOM can still converge to optimal solutions





Sensor Network Location, Large-scale instances

- Test large SNL instances (terminate at 3,000s and $|\boldsymbol{g_k}| \leq \boldsymbol{1e^{-5}}$)

- Compare GD, CG, and DRSOM. (GD and CG use Hager-Zhang Linesearch)

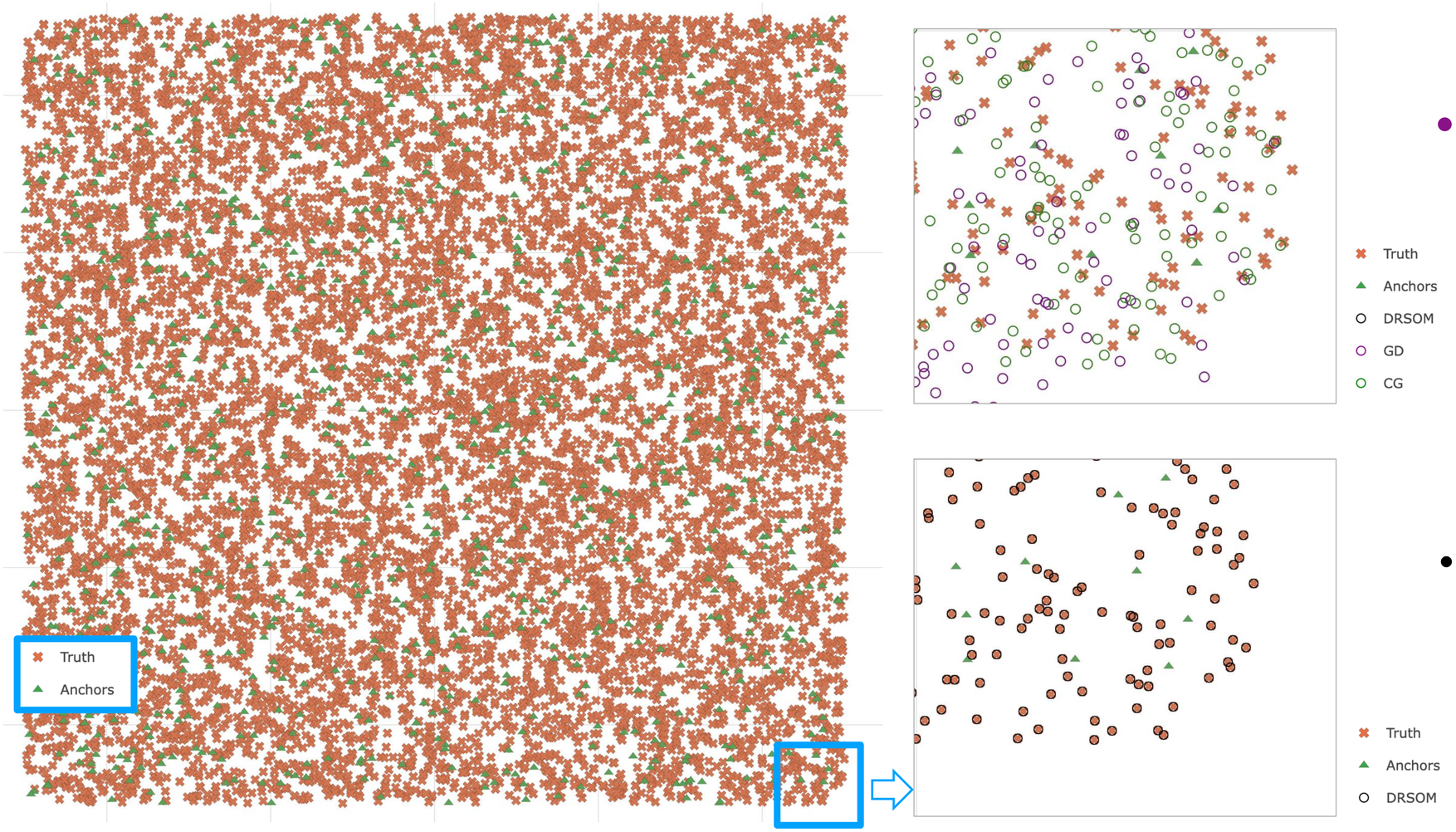
n	m	$\lvert E\rvert$	t		
			CG	DRSOM	GD
500	50	2.2e+04	1.7e+01	1.1e+01	2.3e+01
1000	80	4.6e+04	7.3e+01	3.9e+01	1.8e+02
2000	120	9.4e+04	2.5e+02	1.4e+02	1.1e+03
3000	150	1.4e+05	6.5e+02	1.4e+02	-
4000	400	1.8e+05	1.3e+03	5.0e+02	-
6000	600	2.7e+05	2.0e+03	1.1e+03	-
10000	1000	4.5e+05	-	2.2e+03	-

Table 2: Running time of CG, DRSOM, and GD on SNL instances of different problem size, $|E|$ denotes the number of QCQP constraints. "-" means the algorithm exceeds 3,000s.

- DRSOM has the best running time (benefits of 2nd order info and interpolation!)

Sensor Network Location, Large-scale instances

- Graphical results with 10,000 nodes and 1000 anchors (no noise) **within 3,000 seconds**



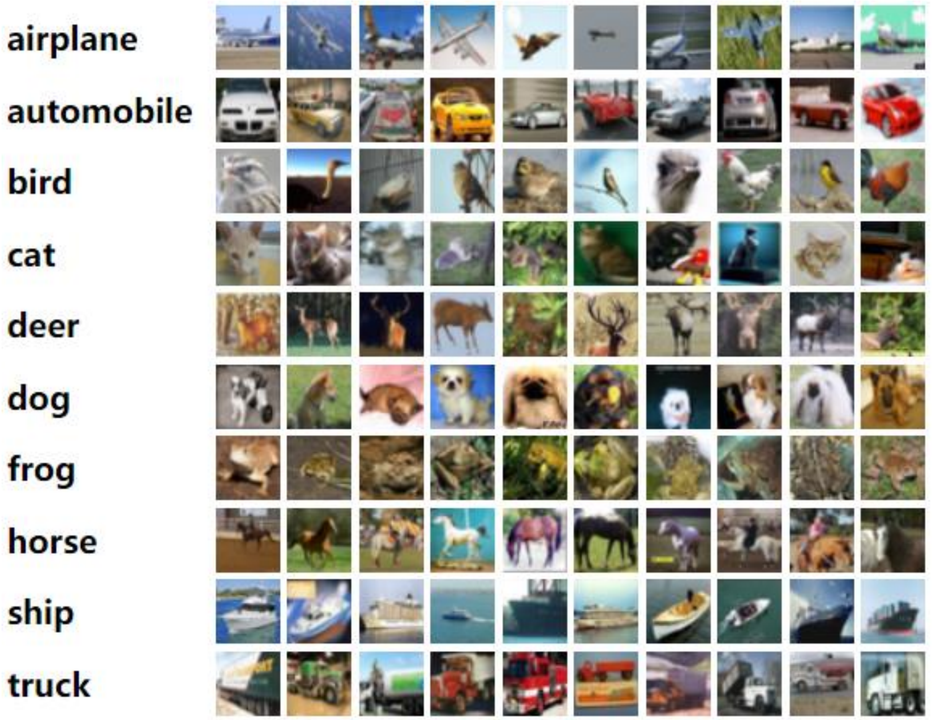
- **GD with Line-search and Hager-Zhang CG both timeout**
- **DRSOM can converge to $|g_k| \leq 1e^{-5}$ in 2,200s**

Sensor Network Online Tracking, 2D and 3D

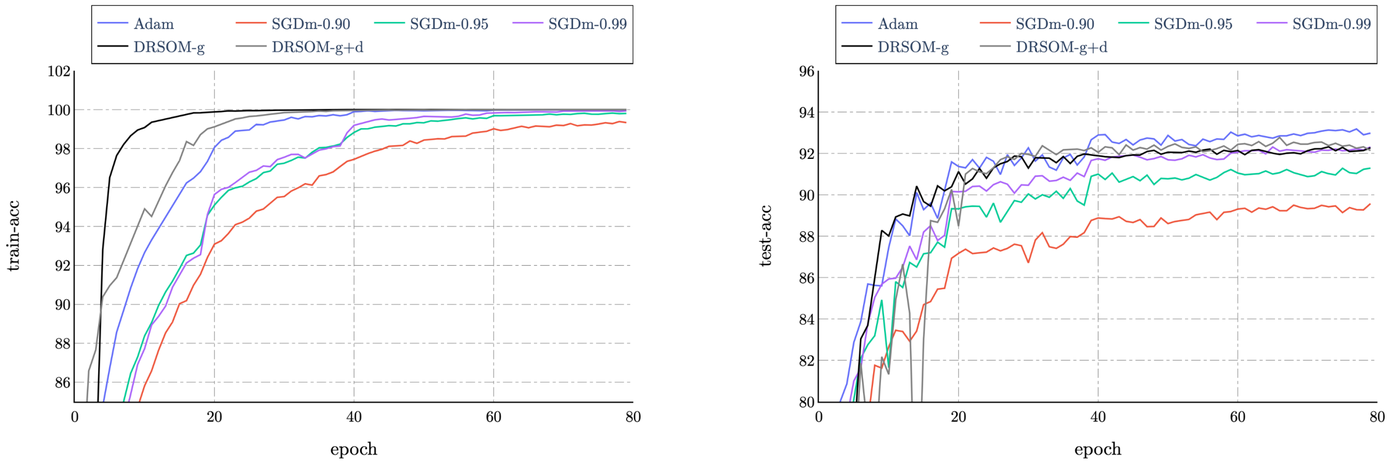
Application: Neural Networks and Deep Learning

To use DRSOM in machine learning problems

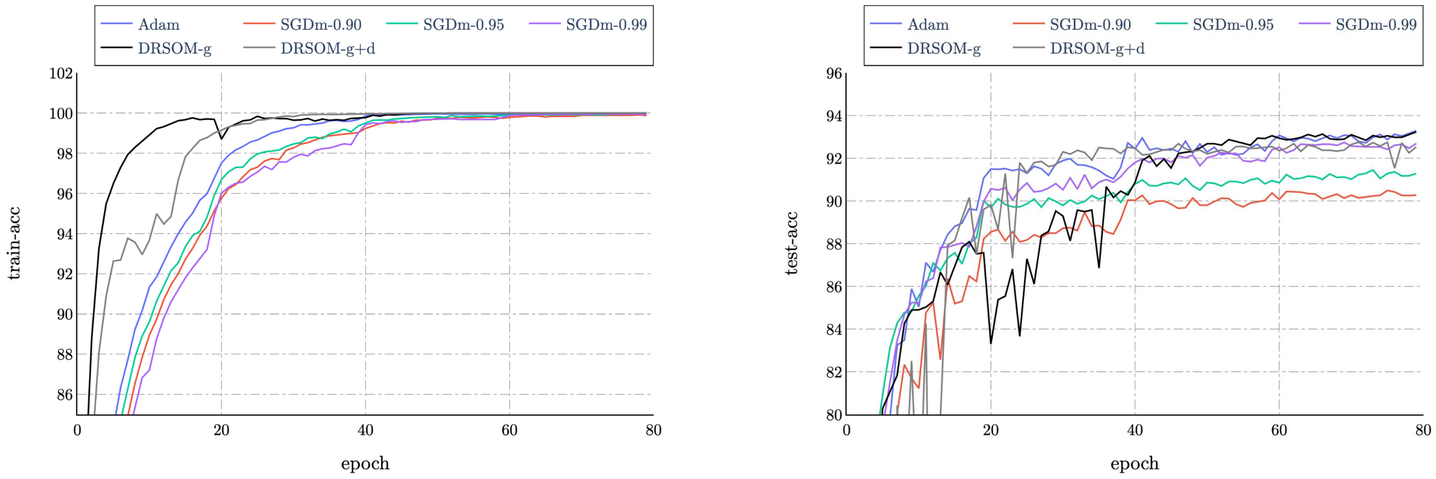
- We apply the mini-batch strategy to a vanilla DRSOM
- Use Automatic Differentiation to compute gradients
- Train ResNet18/Resnet34 Model with CIFAR 10
- Set Adam with initial learning rate 1e-3



Neural Networks and Deep Learning



Training and test results for ResNet18 with DRSOM and Adam



Training and test results for ResNet34 with DRSOM and Adam

Pros

- DRSOM has rapid convergence (30 epochs)
- DRSOM needs little tuning

Cons

- DRSOM may over-fit the models
- Running time can benefit from Interpolation
- Single direction DRSOM is also good

Good potential to be a standard optimizer for deep learning!

Today’s Talk

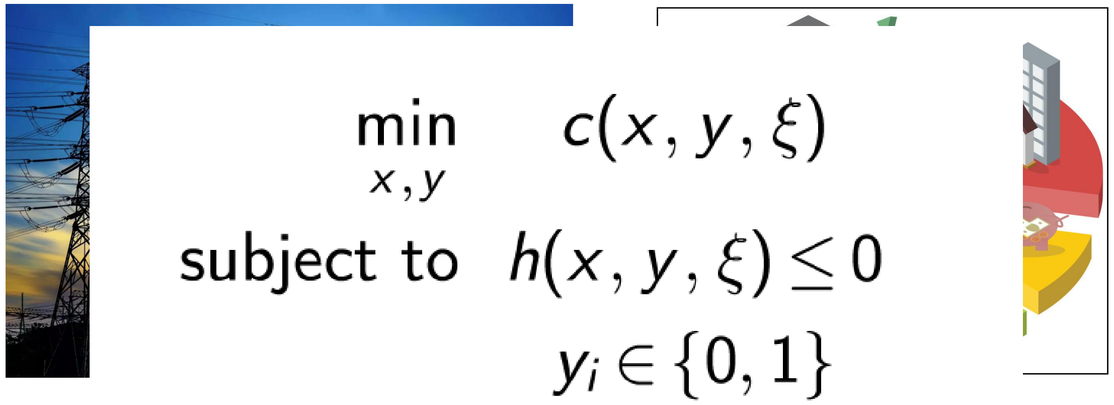
I. Homogeneous Second-Order Descent Direction: An Alternative to the Newton Step

II. A Dimension-Reduced Trust-Region Method

III. **Pre-Trained Statistical Cut Generation for Mixed-Integer Linear Programming Solvers**

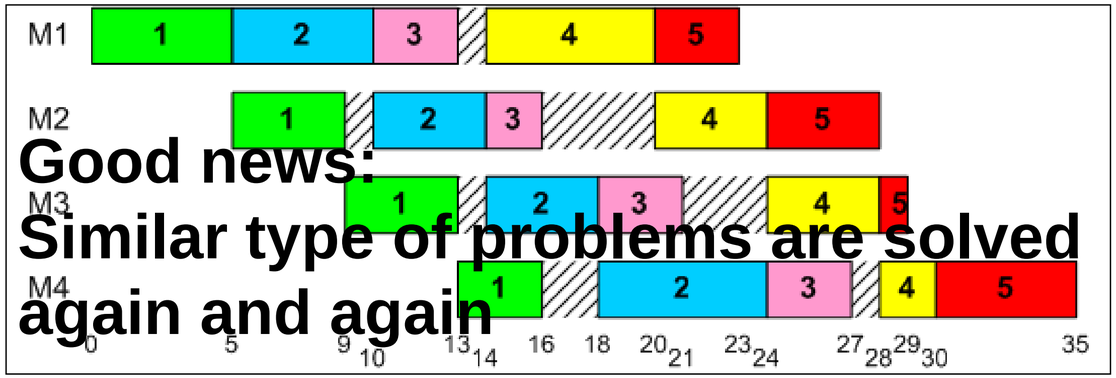
Mixed Integer Optimization is Real

- MIO is a mathematical programming model with discrete integer (binary) decisions
- Mixed integer optimization (MIO) is around us
  Many practical MIO applications are modeled by Mixed Integer Linear Programming (MILP)
  Energy, Finance, Scheduling…
  The simplest MIO model



How easy/hard is MIO?

Bad news: MIOs are notoriously HARD
State-of-the-art solvers may not solve hard MIOs



Good news:
Similar type of problems are solved again and again

Scheduling

Application:
Unit Commitment Problem

- Electricity is generated from units (various generators)
- Transmitted *safely* and *stably* through power grids
- Consumed at minimum (reasonable) price

Optimization has its role to play

minimize Cost of electricity

subject to Safety and Stability

Adaptivity to various units

Unit commitment problem dispatches the units
safely and stably at minimum cost

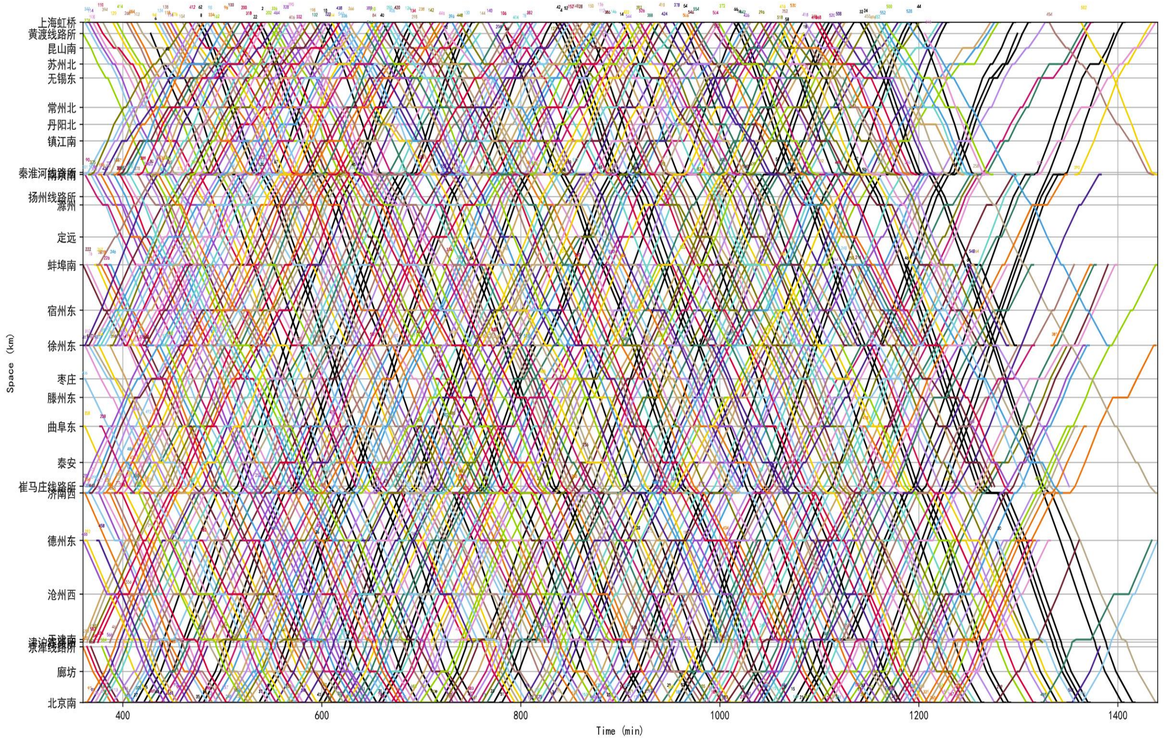
Application: Beijing Public Transport Intelligent Urban Bus Operations Management with Mixed Fleet Types and Charging Schedule

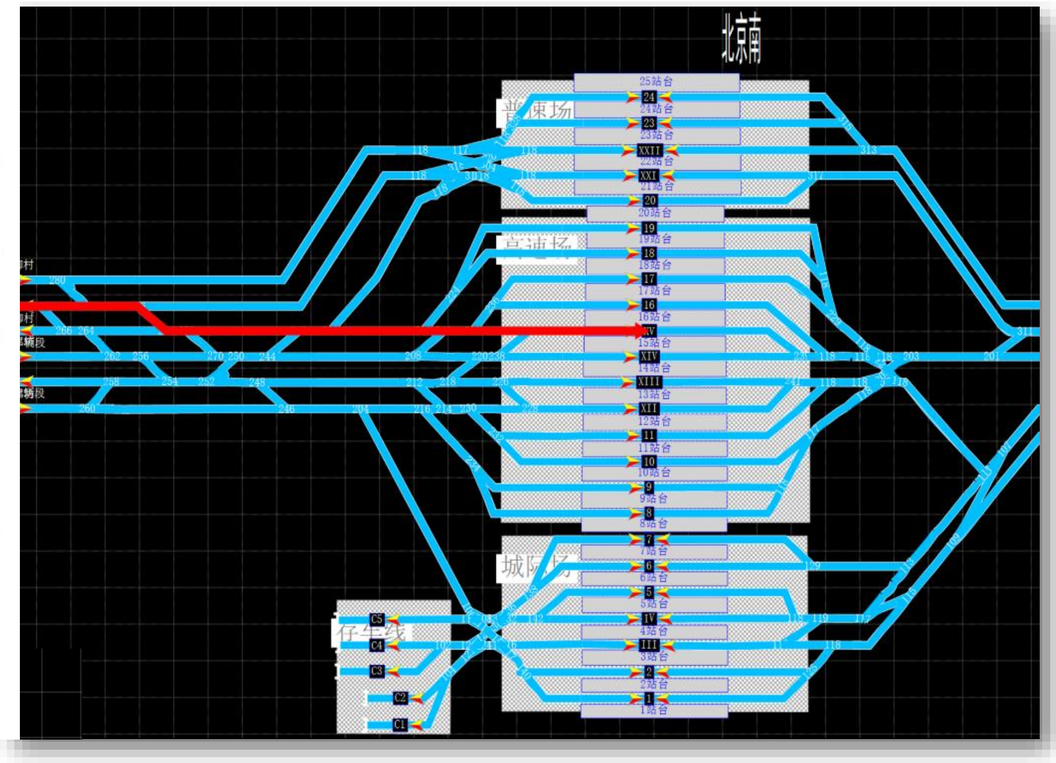
北京公交
Beijing Public Transport

Kickoff 2022.8

Application: Beijing-Shanghai High-speed Railway Scheduling Optimization

COPT, Cardinal Operations 2022





Pooling the Risk via Variance Reduction

- **Given an MILP, the interior point solution of the LP relaxation tells us**

$$\begin{pmatrix} \hat{y}_1(\xi) \\ \hat{y}_2(\xi) \\ \vdots \\ \hat{y}_n(\xi) \end{pmatrix} = \begin{pmatrix} 0.99 \\ 0.12 \\ \vdots \\ 0.38 \end{pmatrix}$$

- **Each $\hat{y}$ is the likelihood a variable takes 1 or 0 in the optimal solution**

- **Each variable introduces some risk/variance of such rounding**

  so that dealing them separately results in extremely risk outcomes

Q: What should we do seeing a set of risky guesses? **A: Put them in a pool!**

Risk Pooling through Variance Reduction

- **Pooling the binary variables by adding "confidence" cardinality cuts**

$$\sum_{i \in \mathcal{U} = \{j : \hat{y}_j(\xi) \geq 0.9\}} y_i^*(\xi) \geq \alpha \cdot |\mathcal{U}| \qquad \sum_{i \in \{j : \hat{y}_j(\xi) \leq 0.1\}} y_i^*(\xi) \leq \beta \cdot |\mathcal{L}|$$

- **Intuitively we know that the above two inequalities are expectedly to hold for $\alpha \to 0.9$ and $\beta \to 0.1$**
- **These two inequalities are exactly *cutting planes* for MILP**
- **The last issue is how to choose α, β to increase the confidence level:**

  Interpret y_j^* as Bernoulli random variables with expectation $\hat{y}_j$,

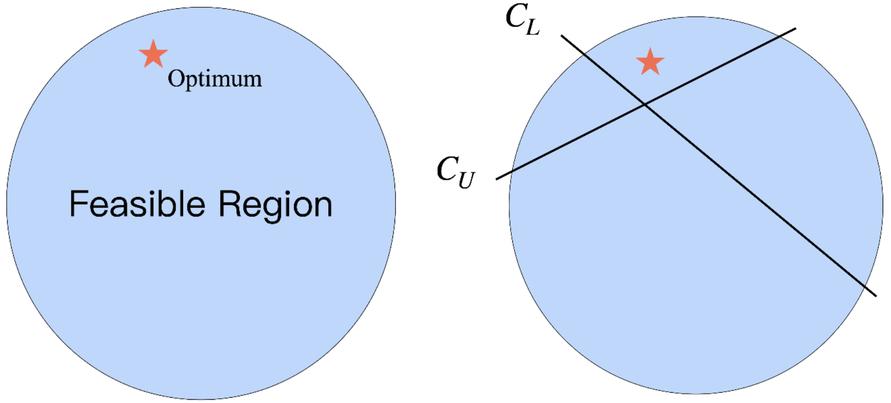
  then justify by concentration inequalities

Statistical Confidence Cut Generation (Gao at al. SHUFE, 2023)

Theorem 2. *Given independent random variables* $\{y_1^*(\xi),\ldots,y_n^*(\xi)\}$ *such that* $\mathbb{E}[y_i^*(\xi)|\xi]=\hat{y}_i(\xi)$, *letting* $\mathcal{U}:=\{i\colon \hat{y}_i(\xi)\geq\tau\}$ *and* $\mathcal{L}:=\{i\colon \hat{y}_i(\xi)<1-\tau\}$ *for* $0.5\leq\tau\leq 1$. *Then w.p.* $1-\delta$, *each of the inequalities below holds.*

$$C_{\mathcal{U}}\colon \sum_{i\in\mathcal{U}} y_i^*(\xi)\geq\sum_{i\in\mathcal{U}}\hat{y}_i(\xi)-\sqrt{\frac{|\mathcal{U}|}{2}\log(1/\delta)}$$

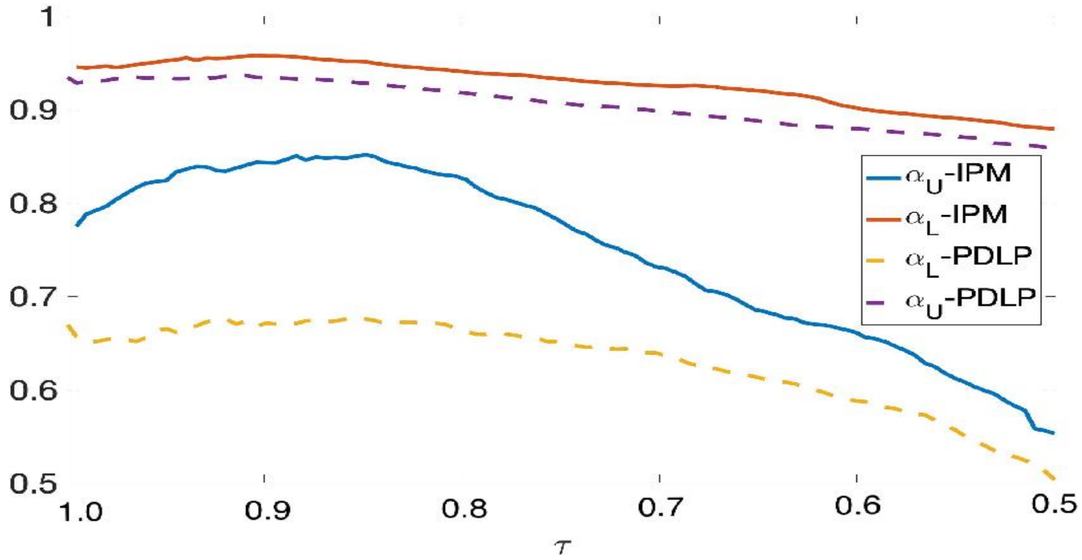
$$C_{\mathcal{L}}\colon \sum_{i\in\mathcal{L}} y_i^*(\xi)\leq\sum_{i\in\mathcal{L}}\hat{y}_i(\xi)+\sqrt{\frac{|\mathcal{L}|}{2}\log(1/\delta)}$$



- **Overall, the two cuts (and their complement) split the whole feasible region into four regions**
- **Solving the most likelihood region of two cuts often gives a satisfying solution with confidence**
- **Branching over all four regions independently will not miss the optimal solution**

Numerical Experiments: Online Cut-Generation

- Tested on IEEEE unit commitment problems using COPT
- Using pre-solved instances to compare speed



Accuracy of prediction by the IPM and PDLP

- Accuracy of interior point prediction can reach 80%
- No loss of optimality
- Remarkable acceleration using proper choosing cut generation parameters

Instance	Original	Stat. cut	Instance	Original	Stat. cut
1	882	187	6	1151	405
2	242	170	7	3600	1510
3	268	168	8	1820	1029
4	495	167	9	3600	758
5	241	182	10	3600	579

Improvement of COPT on IEEE instances

Offline-Training: Using past instances to improve prediction quality

Data-Driven Approaches to Mixed Integer Optimization

Learn from the past and predict the future such as the unit commitment problem in Electrical Power Generation

- **Many real-life MIO applications are solved on a regular basis**

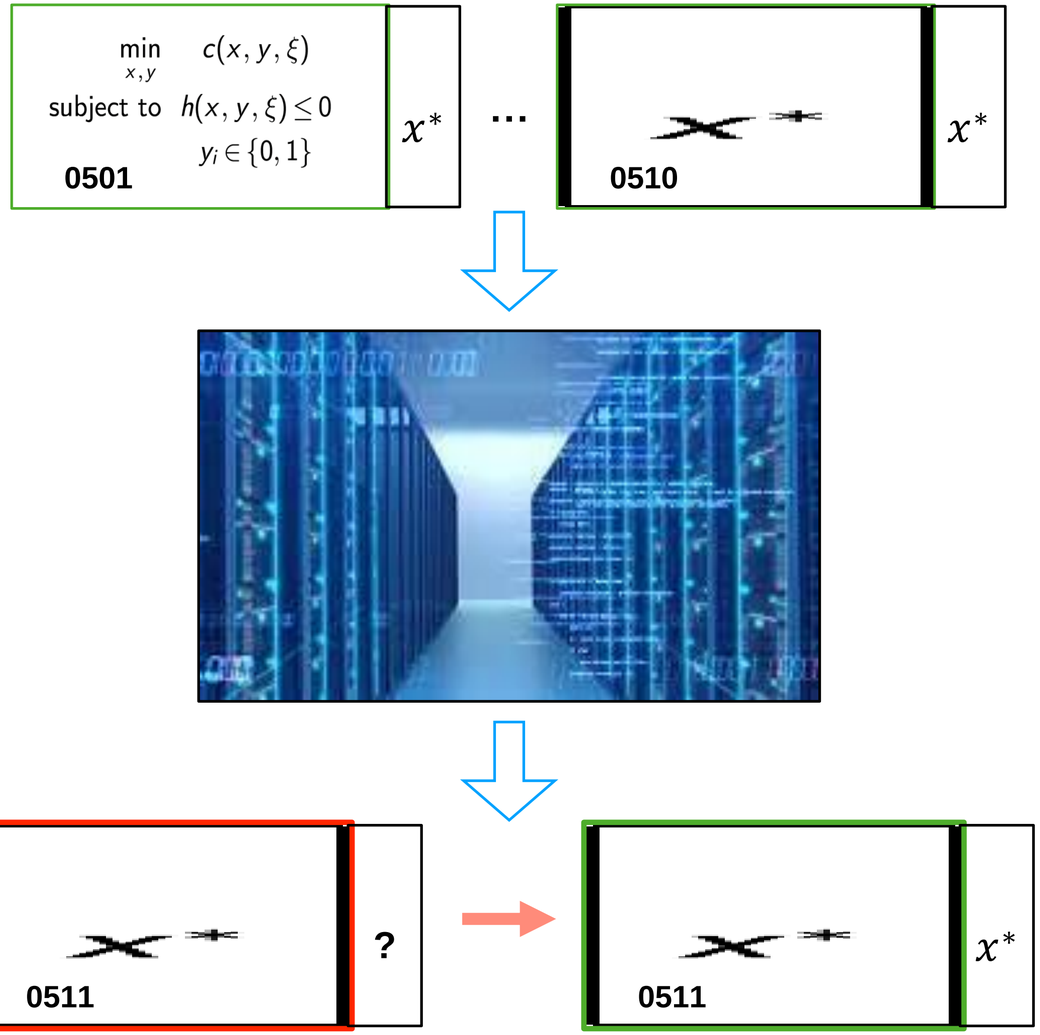
  Unit commitment; portfolio; scheduling…

- **Large amount of data and solutions collected from the past**

  Future instances are similar to the past

- **A natural idea: use machine learning to learn from history**

  Known as pre-trained data-driven approaches

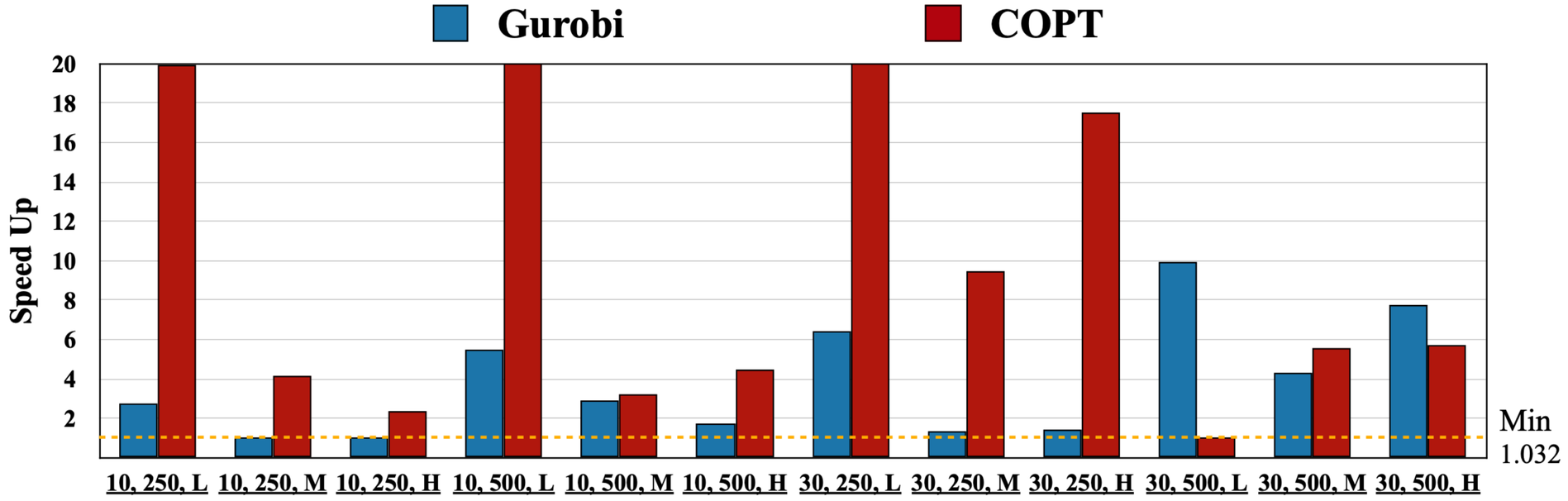


Numerical Test Results I

- **The method is tested on multi-knapsack, set-covering and unit-commitment problems**

  Train from 500 instances and test on 20 instances

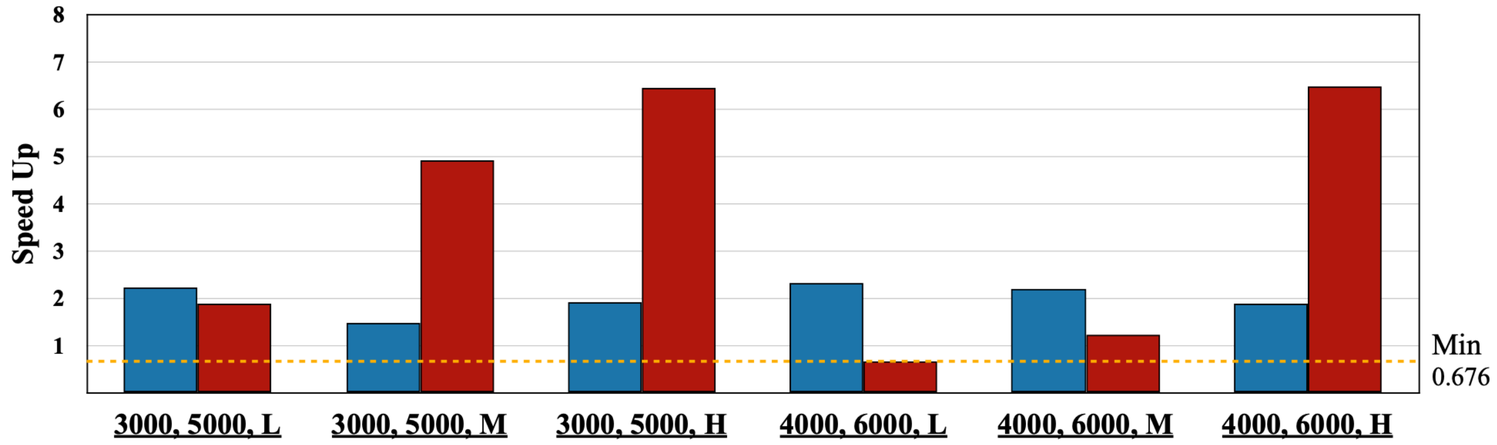
- **Measure the speedup of finding a good solution on in the region formed by two cuts**



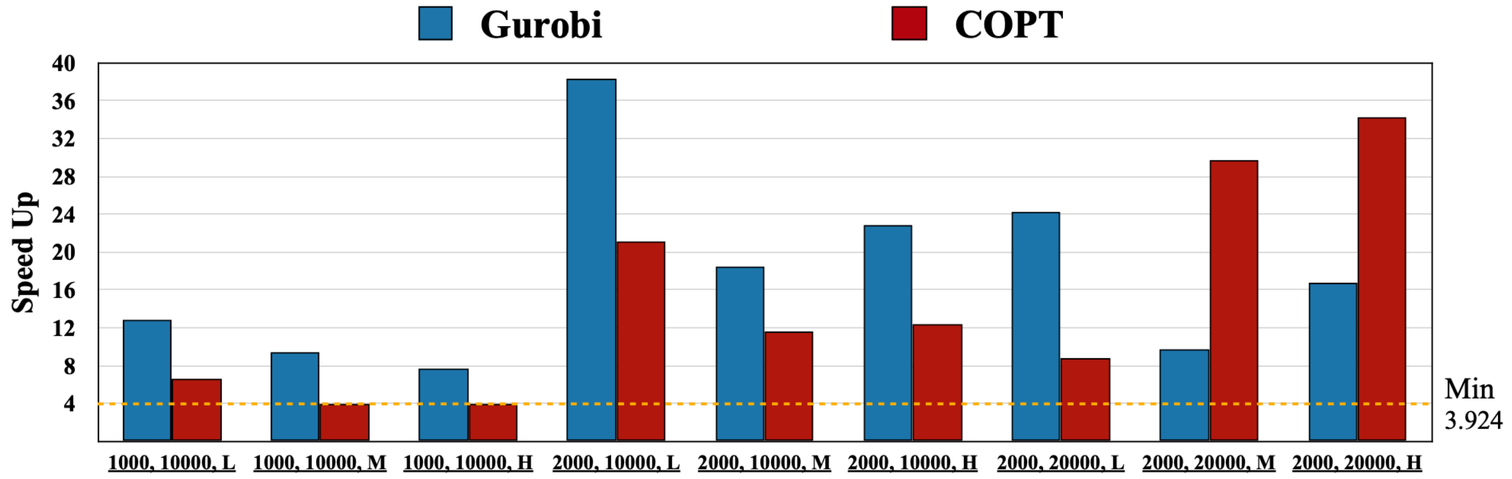
Average speedup on knapsack instances

Numerical Test Results II

- Acceleration by two lines of code
- Remarkable speedup on primal solution finding for both the state of art MIP solvers Gurobi and COPT
- No loss of optimality



Unit Commitment



Set-Covering

Overall Takeaways

Second-Order Derivative information matters and better to integrate FOM and SOM for nonlinear optimization!

Homogeneous second-order direction as an extreme eigenvalue computation is a “cheaper” alternative to the Trust-Region or Newton step computation

Generalized Homogeneous direction is flexible using different $\boldsymbol{\delta\_k}$ and $\boldsymbol{\phi\_k}$ and substitutes for other SOM step

It is possible to do dimension reduction of the trust-region method

It is possible to train Mixed-Integer Linear Programming Solvers and add Statistical Confidence Cuts to significantly accelerate the solution process.

- THANK YOU