Efficient Algorithms via Inexact Linear Solvers and Randomized Sampling

Edgar Solomonik

L P N A @CS@Illinois

Department of Computer Science
University of Illinois at Urbana-Champaign

SIAM ACDA Workshop
CAES-CNRS

Laboratory for Parallel Numerical Algorithms

Talk themes

- sequence of optimization problems
- inexact iterative solvers
- computationally-suitable random distributions

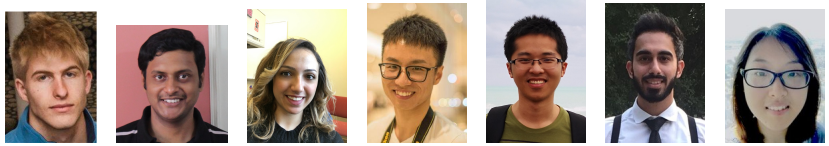
Talk parts

- solving KKT systems arising in interior point (w/ Samah Karim)
- randomized sketching for optimization of tensor decompositions (w/ Linjian Ma)
- randomized sampling for partitioning in parallel sorting (w/ Wentao Yang, Vipul Harsh)

See `http://lpna.cs.illinois.edu`

- Efficient sparse/dense tensor computations
- tensor network methods for simulation of quantum systems
- performance modeling and inexact autotuning
- parallel/HPC inexact graph computations

L·P·N·A @CS@Illinois



DOE CSGF

NSF

Karush-Kuhn-Tucker (KKT) conditions

Consider a general quadratic constrained program

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^T H x + x^T c \\ \text{s.t.} \quad & Ax = b, Cx \geq d \end{aligned}$$

- common in areas such as optimal control, arise when SQP/Newton is applied to general nonlinear programs
- we consider a standard primal-dual interior point optimization approach for this problem
  - augments KKT (optimality) conditions with auxiliary parameters (barrier parameters, based on slack variables and Lagrange multipliers, some of which go to zero later IPM iterations)
  - results in sequence of nonlinear KKT equations, each solved with Newton's method

Interior Point Method (IPM): KKT system

Interior point KKT equations can be written in matrix form as

$$\begin{bmatrix} -H & A^T & C^T \\ A & 0 & 0 \\ C & 0 & D^{(k)} \end{bmatrix} \begin{pmatrix} \Delta x^{(k)} \\ \Delta \lambda^{(k)} \\ \Delta \nu^{(k)} \end{pmatrix} = - \begin{pmatrix} r_g^{(k)} \\ r_e^{(k)} \\ r_a^{(k)} \end{pmatrix}$$

where $D^{(k)} = \left(V^{(k)}\right)^{-1} S^{(k)}$ is diagonal and changing with iteration k.

Traditional approach is to eliminate $\nu^{(k)}$ first, then solve iteratively

$$\begin{bmatrix} -\left(H + C^T \left(D^{(k)}\right)^{-1} C\right) & A^T \\ A & 0 \end{bmatrix} \begin{pmatrix} \Delta x^{(k)} \\ \Delta \lambda^{(k)} \end{pmatrix} = - \begin{pmatrix} r_u^{(k)} \\ r_e^{(k)} \end{pmatrix}$$

We instead use a single (for entire IPM execution) factorization of

$$F = \begin{bmatrix} -H & A^T \\ A & 0 \end{bmatrix}$$

Known Properties of IPM KKT Systems

- Iterative methods and preconditioners can be applied to both 2-by-2 and 3-by-3 systems
- Such saddle point systems are well-studied[1] and arise in numerical PDE solvers[2,3]
- Preconditioners have often been designed to exploit the block structure of the systems[4,5,6]
- The 3-by-3 system has better spectral properties, but the reduced system can nevertheless be preferable computationally[7,8]

---

[1] M. Benzi, G.H. Golub, J. Liesen. Numerical solution of saddle point problems. Acta Numerica, 2005.

[2] R. E. Ewing, R. D. Lazarov, P. Lu, P. S. Vassilevski, PCGM 1990.

[3] C. Greif, D. Schötzau, NLA 2007

[4] G.H. Golub and C. Greif, SISC 2003.

[5] C. Keller, N. I.M. Gould, and A. J. Wathen, SIMAX 2000.

[6] T. Rees, C. Greif, SISC 2007.

[7] B. Morini, V. Simoncini, M. Tani, NLA 2016.

[8] B. Morini, V. Simoncini, M. Tani, COA 2017.

Preconditioning New Reduced KKT System

At each IPM step, given a factorization of $F = \begin{bmatrix} -H & A^T \\ A & 0 \end{bmatrix}$, we iteratively solve a system with the matrix

$$\begin{aligned} K_F^{(k)} &= D^{(k)} - \begin{bmatrix} C & 0 \end{bmatrix} F^{-1} \begin{bmatrix} C^T \\ 0 \end{bmatrix} \\ &= D^{(k)} + CH^{-1}(H - \underbrace{A^T(AH^{-1}A^T)^{-1}A}_{H_A})H^{-1}C^T \end{aligned}$$

Since $H^{-1}A^T$ is in the null space of $H - H_A$, we have

$$\operatorname{rank}(H_A) \leq m_1, \quad \operatorname{rank}(H - H_A) \leq n - m_1$$

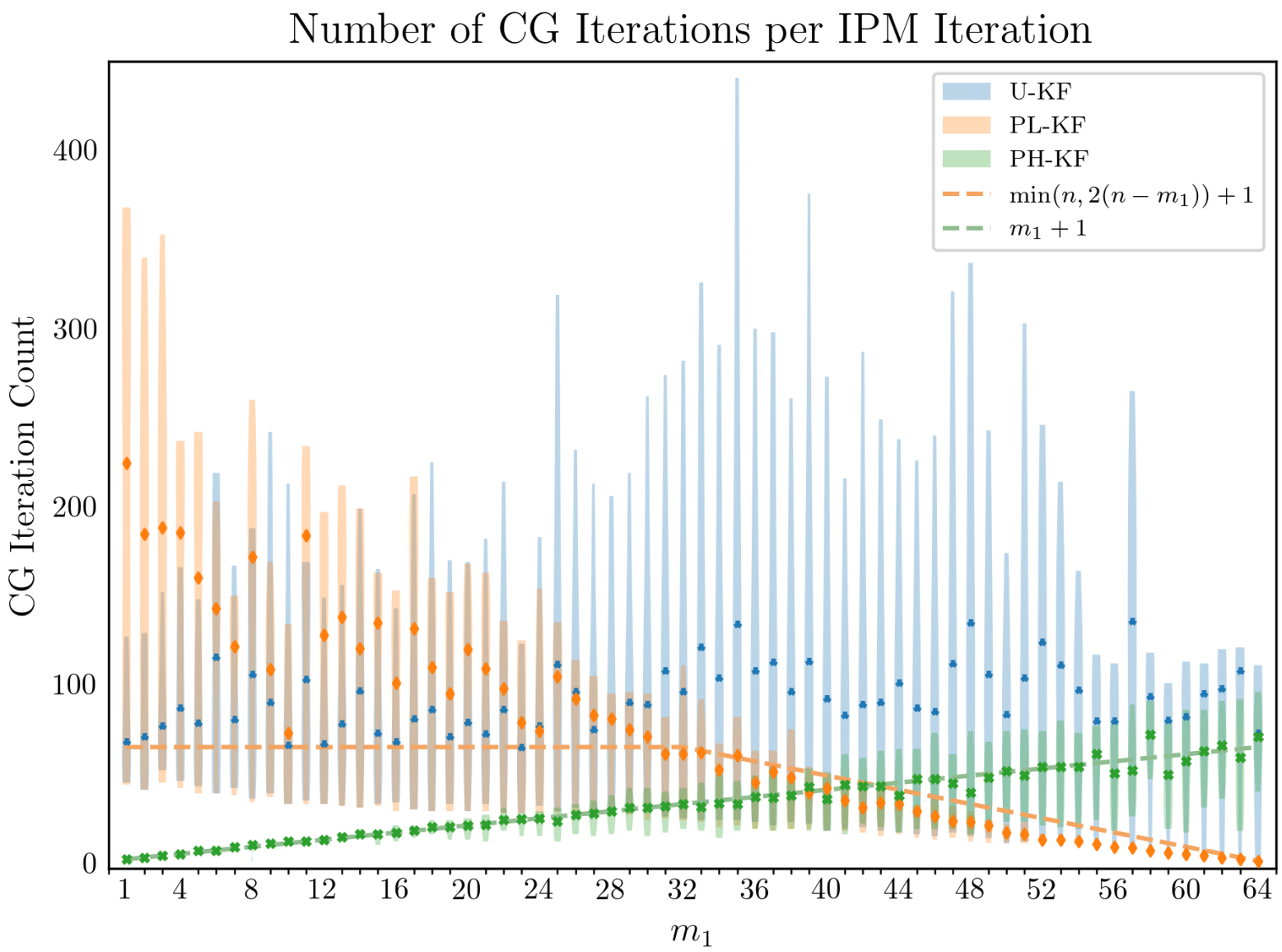
where n is # of variables and m_1 is # equality constraints.

We propose 2 preconditioners for different regimes of # d.o.f. $n - m_1$

$$M_L = D^{(k)} \quad M_H = D^{(k)} + CH^{-1}C^T$$

By the above rank analysis, the low-d.o.f. preconditioner M_L converges after m_1 iterations and M_H after $n - m_1$ (in exact precision)

CG Convergence Results

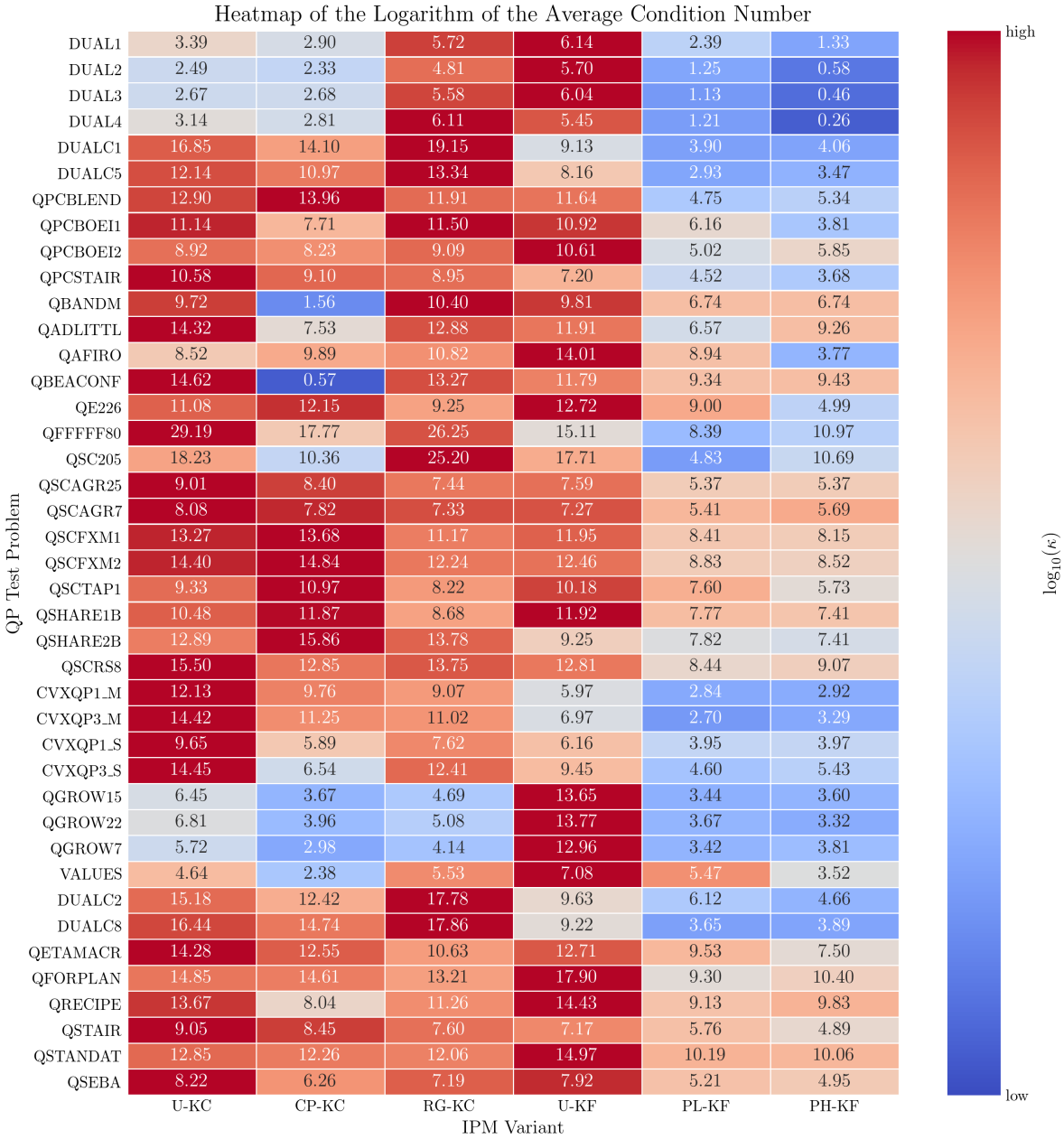


Comparison to Existing Approaches

- Factorize $F = \begin{bmatrix} -H & A^T \\ A & 0 \end{bmatrix}$
- for $k = 1$ until IPM converges
  - Construct preconditioner M to be M_L or M_H depending on # d.o.f. $n - m_1$
  - Factorize M
  - Iteratively solve $M^{-1}K_F x = M^{-1}b$, by applying $K_F = D^{(k)} - \begin{bmatrix} C & 0 \end{bmatrix} F^{-1} \begin{bmatrix} C^T \\ 0 \end{bmatrix}$ in implicit form using factorization of F

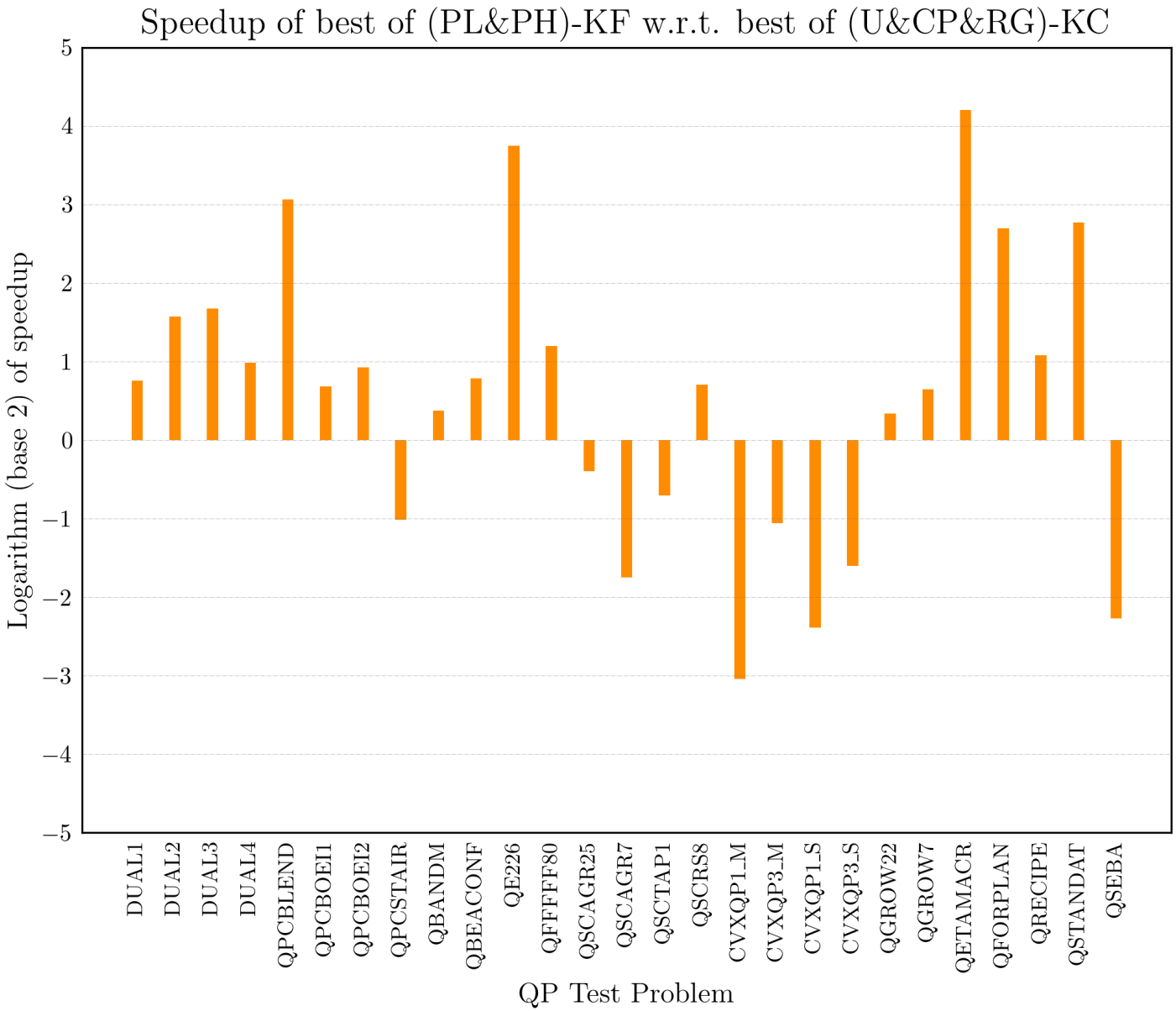
- for $k = 1$ until IPM converges
  - Form augmented system $K_D = \begin{bmatrix} -\left(H + C^T \left(D^{(k)}\right)^{-1} C\right) & A^T \\ A & 0 \end{bmatrix}$
  - Choose M among preconditioners, e.g., constraint preconditioner $\begin{bmatrix} \tilde{D}^{(k)} & A^T \\ A & 0 \end{bmatrix}$ or block-diagonal $\begin{bmatrix} \tilde{D}^{(k)} - A^T W^{(k)} A & \\ & \gamma I \end{bmatrix}$
  - Factorize M
  - Iteratively solve $M^{-1}K_D x = M^{-1}b$

Condition Number Improvement



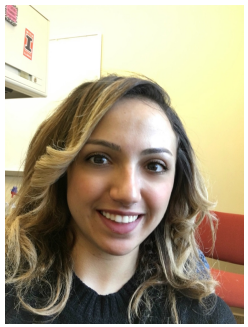
QP Test Problem	U-KC	CP-KC	RG-KC	U-KF	PL-KF	PH-KF
DUAL1	3.39	2.90	5.72	6.14	2.39	1.33
DUAL2	2.49	2.33	4.81	5.70	1.25	0.58
DUAL3	2.67	2.68	5.58	6.04	1.13	0.46
DUAL4	3.14	2.81	6.11	5.45	1.21	0.26
DUALC1	16.85	14.10	19.15	9.13	3.90	4.06
DUALC5	12.14	10.97	13.34	8.16	2.93	3.47
QPCBLEND	12.90	13.96	11.91	11.64	4.75	5.34
QPCBOEI1	11.14	7.71	11.50	10.92	6.16	3.81
QPCBOEI2	8.92	8.23	9.09	10.61	5.02	5.85
QPCSTAIR	10.58	9.10	8.95	7.20	4.52	3.68
QBANDM	9.72	1.56	10.40	9.81	6.74	6.74
QADLITTL	14.32	7.53	12.88	11.91	6.57	9.26
QAFIRO	8.52	9.89	10.82	14.01	8.94	3.77
QBEACONF	14.62	0.57	13.27	11.79	9.34	9.43
QE226	11.08	12.15	9.25	12.72	9.00	4.99
QFFFFF80	29.19	17.77	26.25	15.11	8.39	10.97
QSC205	18.23	10.36	25.20	17.71	4.83	10.69
QSCAGR25	9.01	8.40	7.44	7.59	5.37	5.37
QSCAGR7	8.08	7.82	7.33	7.27	5.41	5.69
QSCFXM1	13.27	13.68	11.17	11.95	8.41	8.15
QSCFXM2	14.40	14.84	12.24	12.46	8.83	8.52
QSCTAP1	9.33	10.97	8.22	10.18	7.60	5.73
QSHARE1B	10.48	11.87	8.68	11.92	7.77	7.41
QSHARE2B	12.89	15.86	13.78	9.25	7.82	7.41
QSCRS8	15.50	12.85	13.75	12.81	8.44	9.07
CVXQP1_M	12.13	9.76	9.07	5.97	2.84	2.92
CVXQP3_M	14.42	11.25	11.02	6.97	2.70	3.29
CVXQP1_S	9.65	5.89	7.62	6.16	3.95	3.97
CVXQP3_S	14.45	6.54	12.41	9.45	4.60	5.43
QGROW15	6.45	3.67	4.69	13.65	3.44	3.60
QGROW22	6.81	3.96	5.08	13.77	3.67	3.32
QGROW7	5.72	2.98	4.14	12.96	3.42	3.81
VALUES	4.64	2.38	5.53	7.08	5.47	3.52
DUALC2	15.18	12.42	17.78	9.63	6.12	4.66
DUALC8	16.44	14.74	17.86	9.22	3.65	3.89
QETAMACR	14.28	12.55	10.63	12.71	9.53	7.50
QFORPLAN	14.85	14.61	13.21	17.90	9.30	10.40
QRECIPE	13.67	8.04	11.26	14.43	9.13	9.83
QSTAIR	9.05	8.45	7.60	7.17	5.76	4.89
QSTANDAT	12.85	12.26	12.06	14.97	10.19	10.06
QSEBA	8.22	6.26	7.19	7.92	5.21	4.95

Arithmetic Cost Model Comparison



Extensions

- What if H is semidefinite? What if F is singular?
  - our approach assumed we can factorize F
  - our high-d.o.f. preconditioner $M_H = D^{(k)} + CH^{-1}C^T$ assumed H is nonsingular
  - regularization can ensure H and F are nonsingular
- When H is semidefinite but F is nonsingular
  - can factorize F with pivoting, use pseudoinverse of H in preconditioner
- When H and F are singular (A is assumed to be full rank) or when parts of H are changing (non-quadratic)
  - can use our approach with a smaller fixed (factorized) subsystem
- For further details, see "Efficient Preconditioners for Interior Point Methods via a New Schur Complementation Strategy", Samah Karim, E.S., (SIMAX/arXiv:2104.12916)



Randomized Sketching

- Linear sketching provides an alternative to iterative solvers or can be used to precondition them[1]
- A random sketching matrix $S \in \mathbb{R}^{k \times n}$ is called (δ, ϵ)-accurate if it satisfies the Johson-Lindenstrauss (JL) Lemma[2]

$$\forall x \in \mathbb{S}^{n-1}, \quad \Pr[|\|Sx\|_2 - 1| > \epsilon] < \delta$$

- The JL Lemma also implies approximate preservation of distances between sketches of arbitrary set of points $x_1, \ldots, x_d$
- To sketch a linear LSQ problem, a $(\delta, \max(\epsilon/d^2, \epsilon^2/d))$-accurate sketching matrix S (with $k = O(\min(d^2/\epsilon, d/\epsilon^2)\log(1/\delta))$) gives

$$\forall A \in \mathbb{R}^{n \times d}, b \in \mathbb{R}^n, \text{ if } Ax \cong b \text{ and } SA\hat{x} \cong Sb,$$
$$\Pr[\|A\hat{x} - b\|_2 > (1+\epsilon)\|Ax - b\|_2] < \delta$$

[1] H. Avron, P. Maymounkov, and S. Toledo, SISC 2010

[2] For a review, see "Sketching as a Tool for Numerical Linear Algebra", D. Woodruff

Random Distributions for Efficient Sketching

- If elements s_{ij} are independently drawn from a sub-Gaussian distribution, S satisfies JL Lemma with $k = O(\log(\delta)/\epsilon^2)$
- Selecting each column s_i from $\{1, -1\} \times \{e_1, \ldots e_k\}$ (CountSketch) yields JL Lemma with same k
  - CountSketch preserves sparsity, $\#\text{nnz}(SA) \leq \#\text{nnz}(A)$
- Selecting S as $S_1 S_2$, $I \otimes S_1$, $S_1 \otimes S_2$, or with other tensor substructure also yields provably accurate sketches[1]
  - if input $x = u \otimes v$, cost of computing Sx reduced from $O(\dim(x))$ to $O(\dim(u) + \dim(v))$
- If columns of S are drawn independently from the same random distribution D, for the JL lemma to hold, we need[2]

$$\text{for } s \sim D, \mathbb{E}[\|s\|_2] = 1, \quad \Pr[\|s\|_2^2 > t] \qquad < 2e^{-t/C} \text{ and}$$

$$\text{for } s_1, s_2 \sim D, \mathbb{E}[\langle s_1, s_2 \rangle] = 0, \Pr[|\langle s_1, s_2 \rangle| > t] < 2e^{-t/C}$$

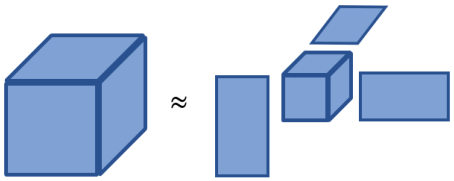
[1] R. Pagh, TOCT 2013; T. D. Ahle, M. Kapralov, J. B. Knudsen, R. Pagh, A. Velingker, D. P. Woodruff, and A. Zandieh, SODA 2020

[2] preliminary work with Changsheng Chen and Linjian Ma

Background on Tensor Decompositions

Tucker decomposition

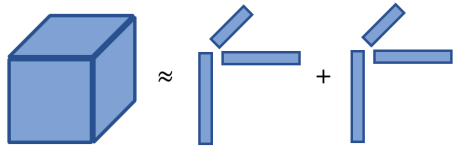
$$\mathcal{T} \approx \mathcal{X} \times_1 A \times_2 B \times_3 C$$



- $\mathcal{T} \in \mathbb{R}^{n \times n \times n}$, $\mathcal{X} \in \mathbb{R}^{R \times R \times R}$
- $A, B, C \in \mathbb{R}^{n \times R}$ with orthonormal columns, $R < n$

CP decomposition

$$\mathcal{T} \approx \sum_{r=1}^{R} a_r \circ b_r \circ c_r$$



- $\mathcal{T} \in \mathbb{R}^{n \times n \times n}$, $A = [a_1, \ldots, a_R] \in \mathbb{R}^{n \times R}$
- $R < n^2$

Higher order orthogonal iteration (HOOI)

$$\min_{A, \mathcal{X}} \frac{1}{2} \left\| (C \otimes B) X_{(1)}^T A^T - T_{(1)}^T \right\|_F^2$$

CP-Alternating least squares (CP-ALS)

$$\min_{A} \frac{1}{2} \left\| (C \odot B) A^T - T_{(1)}^T \right\|_F^2$$

Prior work on sketched tensor decompositions

- Sketching for CP-ALS: C. Battaglino, G. Ballard, and T. Kolda, SIMAX 2018
- Sketching for Tucker-ALS (not HOOI): Malik and Becker, NeurIPS 2018

Sketching HOOI

Higher order orthogonal iteration (HOOI)

$$\min_{A,\boldsymbol{\mathcal{X}}} \frac{1}{2} \left\| (C \otimes B) X_{(1)}^T A^T - T_{(1)}^T \right\|_F^2$$

- Kronecker product $C \otimes B \in \mathbb{R}^{n^2 \times R^2}$
- Costs $\Theta(n^3 R)$ or $\Theta(\text{nnz}(\boldsymbol{\mathcal{T}}) R^2)$
- Fast convergence

CP-Alternating least squares (CP-ALS)

$$\min_{A} \frac{1}{2} \left\| (C \odot B) A^T - T_{(1)}^T \right\|_F^2$$

- Khatri-Rao product $C \odot B \in \mathbb{R}^{n^2 \times R}$
- Costs $\Theta(n^3 R)$ or $\Theta(\text{nnz}(\boldsymbol{\mathcal{T}}) R)$
- Slow convergence

New result for sketched low rank approximation ($R \ll n$):

- Sketched HOOI for Tucker decomposition (Linjian Ma and E.S., NeurIPS 2021 / arXiv:2104.01101)
- Overall cost with t HOOI sweeps reduced to $O\left(\text{nnz}(\boldsymbol{\mathcal{T}}) + t\left(nR^3 + R^6\right)\right)$
- Can also accelerate CPD via performing CP-ALS on the Tucker core tensor

Cost comparison for order 3 tensor

ALS + TensorSketch (Malik and Becker, NeurIPS 2018)

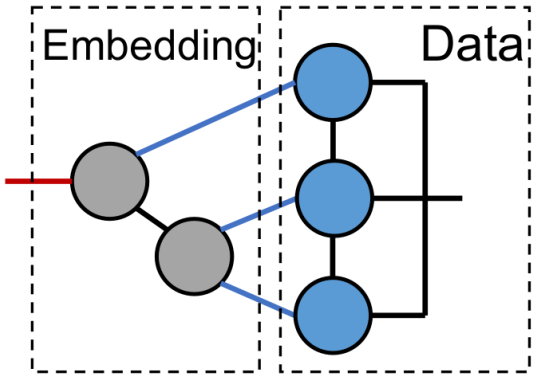
- Solving for each factor matrix or the core tensor at a time
- $\min_A \frac{1}{2} \left\| (C \otimes B) X_{(1)}^T A^T - T_{(1)}^T \right\|_F^2$ or
  $\min_{\boldsymbol{x}} \frac{1}{2} \| (C \otimes B \otimes A) \mathrm{vec}(X) - \mathrm{vec}(T) \|_F^2$

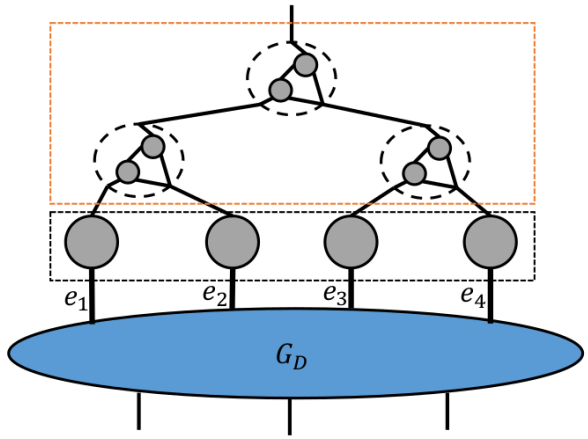
Algorithm for Tucker	LS subproblem cost	Sketch size (k)
HOOI	$\Omega(\mathrm{nnz}(\boldsymbol{\mathcal{T}})R)$	/
ALS + TensorSketch	$\tilde{O}(knR + kR^3)$	$O((R^2/\delta) \cdot (R^2 + 1/\epsilon))$
HOOI + TensorSketch	$O(knR + kR^4)$	$O((R^2/\delta) \cdot (R^2 + 1/\epsilon^2))$
HOOI + leverage scores	$O(knR + kR^4)$	$O(R^2/(\epsilon^2\delta))$

Sketched HOOI performs well in experiments

- Across a few test matrices, sketched HOOI converges to at least 98% of the accuracy of plain HOOI with $k = 16R^2$ (same number of iterations)
- ALS+TensorSketch attains noticeably lower accuracy than HOOI

Optimal Sketching for Arbitrary Tensor Networks





Given input data with tensor network structure, seek cost-optimal accurate embeddings (Linjian Ma and E.S., arXiv:2205.13163)

- Assume Gaussian sketching and classical $O(n^3)$ matmul cost
- Any 'linearizable' tensor network embedding is accurate (follows from $S = S_1 \cdots S_m$, $S_i = I \otimes \cdots \otimes \hat{S}_i \otimes \cdots \otimes I$ satisfying JL Lemma)
- Ahle et al (SODA 2020) consider binary tree embeddings
- We derive a non-tree embedding that reduces cost by up to $O(\sqrt{k})$ for the same accuracy for Kronecker product inputs
- We also derive a general embedding and prove optimality (with some restrictions) for general inputs

Partitioning for Parallel Sorting

Problem (Parallel Sorting):
Given p processors with n/p keys per process, sort keys so that processor i owns the ith subsequence of $\Theta(n/p)$ keys in the global order.

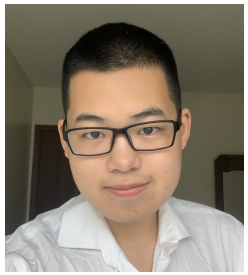
We focus on sampling keys to obtain a balanced p-way partition

- Communication-optimal algorithms based on mergesort[1] communicate all keys multiple times and have not shown to be effective in practice
- State-of-the-art approaches communicate most keys once or twice, by first computing an approximate partition of the global keys
- Obtaining a balanced partition essentially amounts to finding a sample key for each interval of n/p keys in the global order

[1] Goodrich, ACM STOC 1996 (derived from Cole's parallel mergesort, SIAM JC 1988)

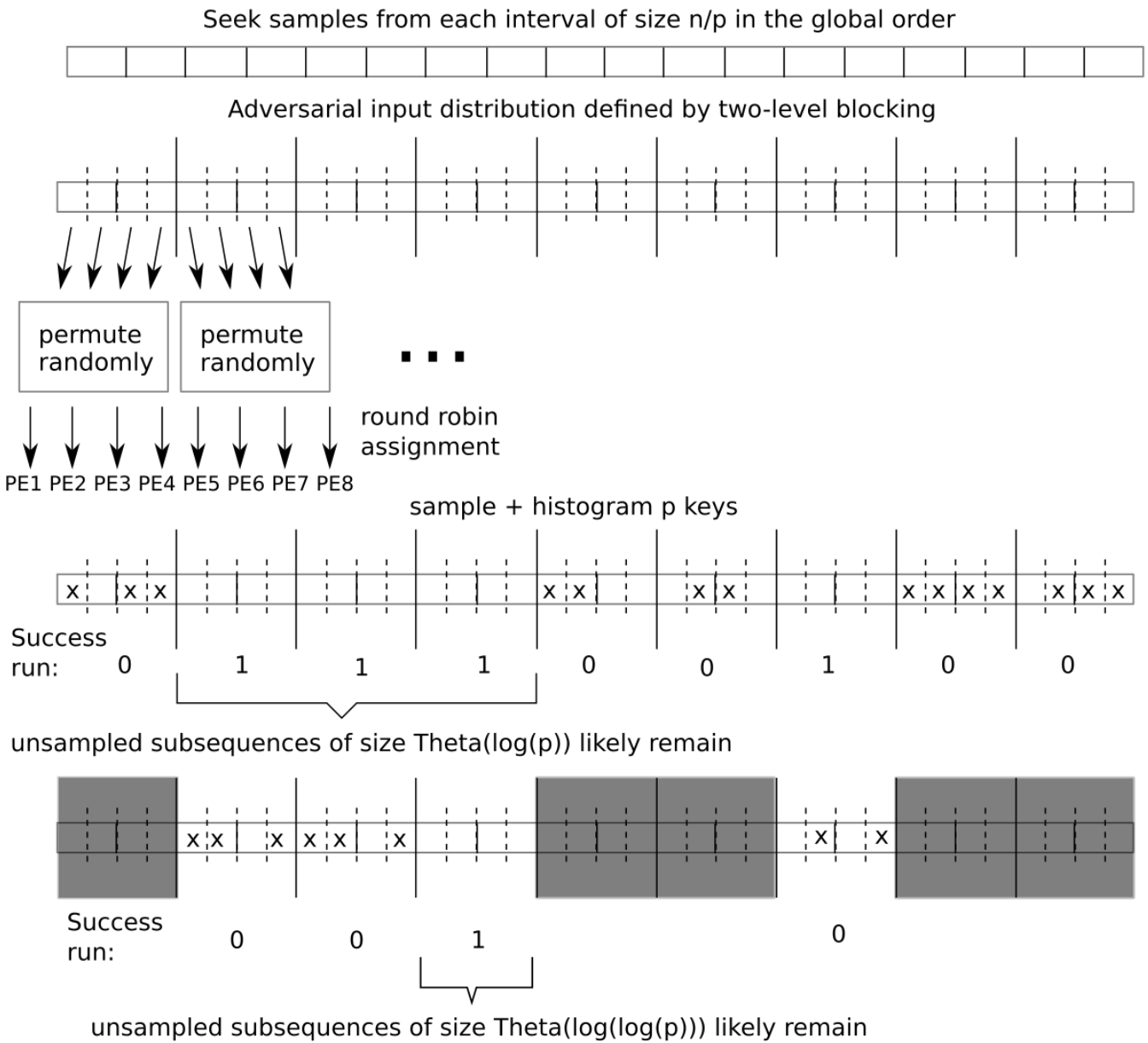
Sample and Histogram Sort

- Sample Sort[1] finds balanced splitting with a sample of size $O(p \log p)$
- Histogram Sort[2] calculates histograms – global ranks of a set of an iteratively-refined set of keys, and is practical[3]
- Histogram Sort with Sampling[4] selects keys to histogram by selective sampling, needs $O(\log \log p)$ rounds
- We show[5] $\Theta(\log^* p)$ rounds with $O(p)$ total samples suffice; lower bound uses Yao's principle and distribution theory of runs (A.M. Mood, 1940)



[1] H. Shi and J. Schaeffer, 1992 and others
[2] L. Kale and S. Krishnan, 1993
[3] E.S., L. Kale, IPDPS 2010
[4] Vipul Harsh, E.S., L. Kale, SPAA 2019
[5] W. Yang, V. Harsh, E.S., arXiv:2204.04599

Partition Sampling Lower Bound Proof Overview



Laboratory for Parallel Numerical Algorithms

Talk themes

- sequence of optimization problems
- inexact iterative solvers
- computationally-suitable random distributions

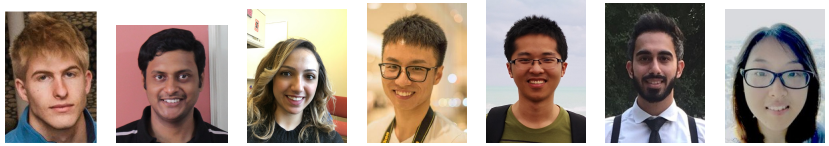
Talk parts

- solving KKT systems arising in interior point (w/ Samah Karim)
- randomized sketching for optimization of tensor decompositions (w/ Linjian Ma)
- randomized sampling for partitioning in parallel sorting (w/ Wentao Yang, Vipul Harsh)

See `http://lpna.cs.illinois.edu`

- Efficient sparse/dense tensor computations
- tensor network methods for simulation of quantum systems
- performance modeling and inexact autotuning
- parallel/HPC inexact graph computations

L·P·N·A @CS@Illinois



DOE CSGF

NSF