Optimization in Data Science and Machine Learning/Decision-Making

ANTAI, SJTU, MARCH 17, 2023

Yinyu Ye
Stanford University and CUHKSZ (Sabbatical Leave)

Today's Talk

1. Online Linear Programming Algorithms and Applications

2. Accelerated Second-Order Methods and Applications

3. Equitable Covering & Partition – Divide and Conquer

Topic 1. Online Linear Programming

- **1、在线学习理论与算法研究（Agrawal et al. 2010,14, Li&Y 2022)**
- 什么是在线学习问题?
  - 传统机器学习问题：有大量（训练）数据，找到最佳模型（例子：回归模型、树模型）

已有数据 $\longrightarrow$ 最佳模型

  - 在线学习：数据的生成和学习是同时发生的，由决策影响（例如多臂老虎机问题）

数据 $\xrightarrow{\text{算法}}$ 决策 $\xrightarrow{\text{潜在模型}}$ 反馈

需要一边学习，一边优化

From Zizhuo Wang

Linear Programming and LP Giants won Nobel Prize…

$$\max \quad \sum \pi_j x_j$$

$$\text{s.t.} \quad \sum_j \boldsymbol{a}_j x_j \leq \boldsymbol{b},$$

$$0 \leq x_j \leq 1 \quad \forall\, j = 1, \ldots, n$$

$$\min \quad \boldsymbol{b}^{\mathrm{T}} \boldsymbol{p} + \sum \max\{0, \pi_j - \boldsymbol{a}_j{}^T \boldsymbol{p}\}$$

$$\text{s.t.} \quad \boldsymbol{p} \geq \boldsymbol{0}$$

Online Auction Example

- There is a fixed selling period or number of buyers; and there is a fixed inventory of goods
- Customers come and require a bundle of goods and make a bid
- Decision: To sell or not to sell to each individual customer on the fly?
- Objective: Maximize the revenue.

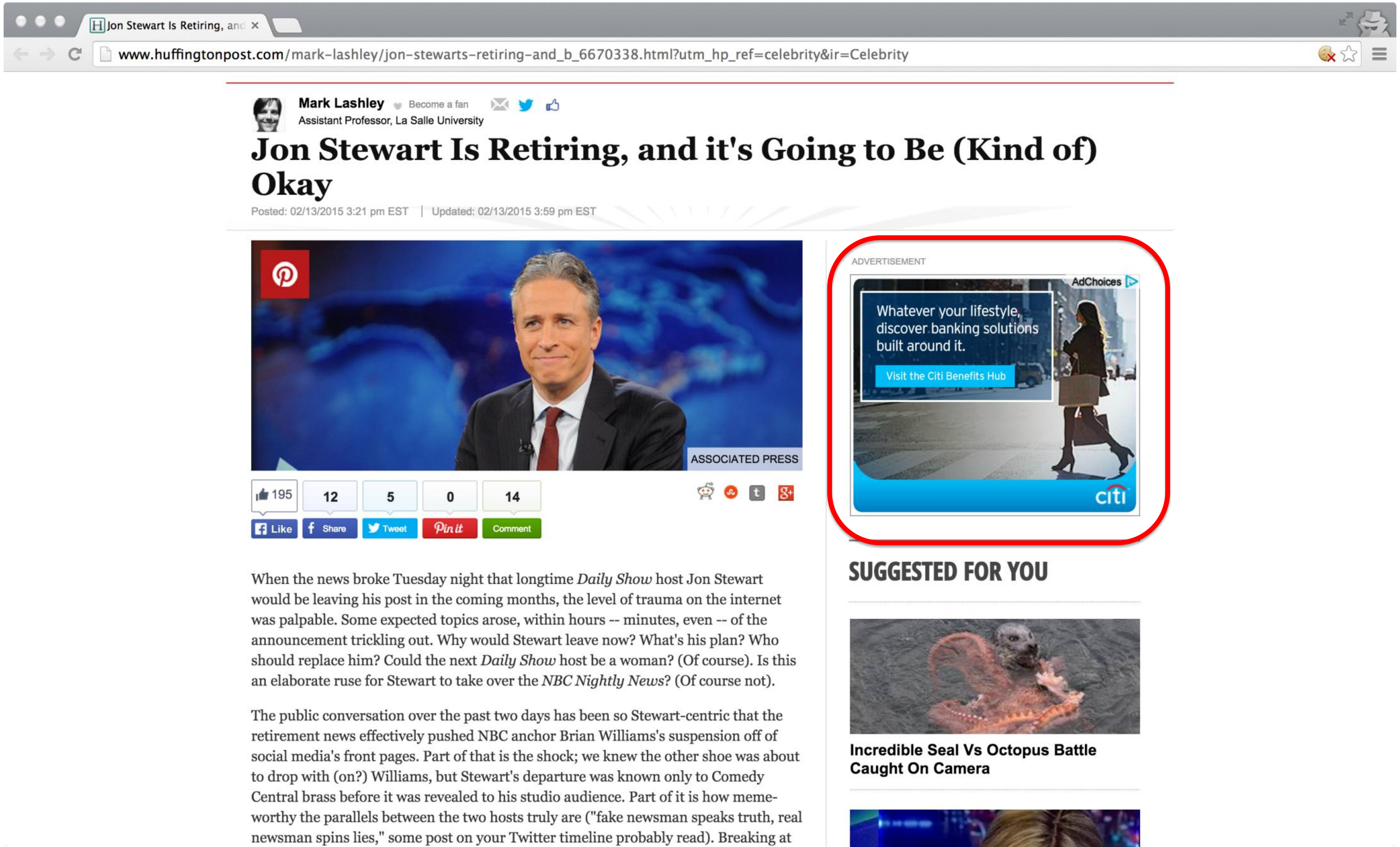
Bid #	$100	$30	….	…	…	Inventory
Decision	x1	x2				
Pants	1	0	….	…	…	100
Shoes	1	0				50
T-Shirts	0	1				500
Jackets	0	0				200
Hats	1	1	…	…	…	1000

Price Mechanism for Online Auction

- **Learn and compute itemized optimal prices**
- **Use the prices to price each bid**
- **Accept if it is a over bid, and reject otherwise**

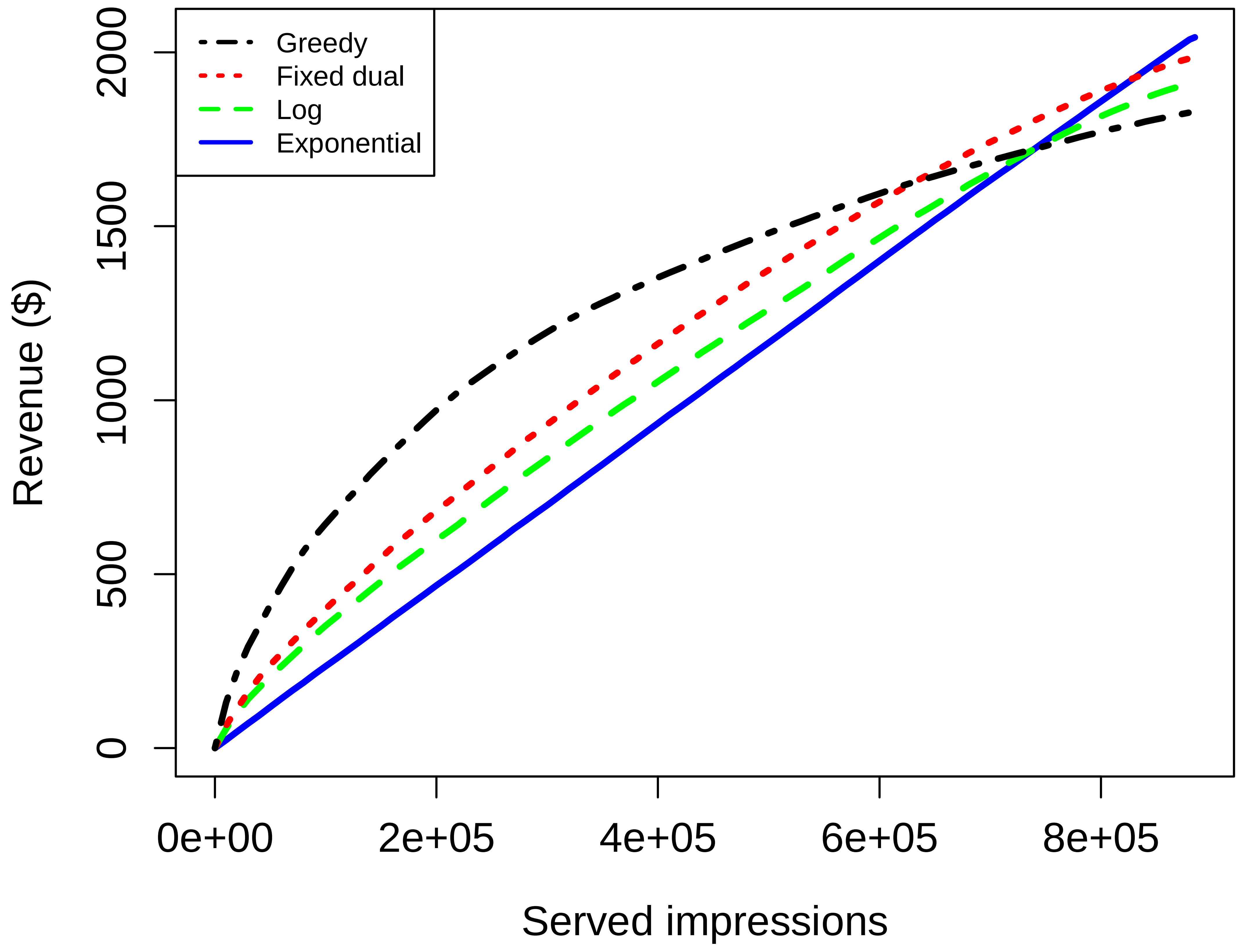
Bid #	$100	$30	….	…	…	Inventory	Price?
Decision	x1	x2					
Pants	1	0	….	…	…	100	45
Shoes	1	0				50	45
T-Shirts	0	1				500	10
Jackets	0	0				200	55
Hats	1	1	…	…	…	1000	15

Application I: Online Matching for Display Advertising



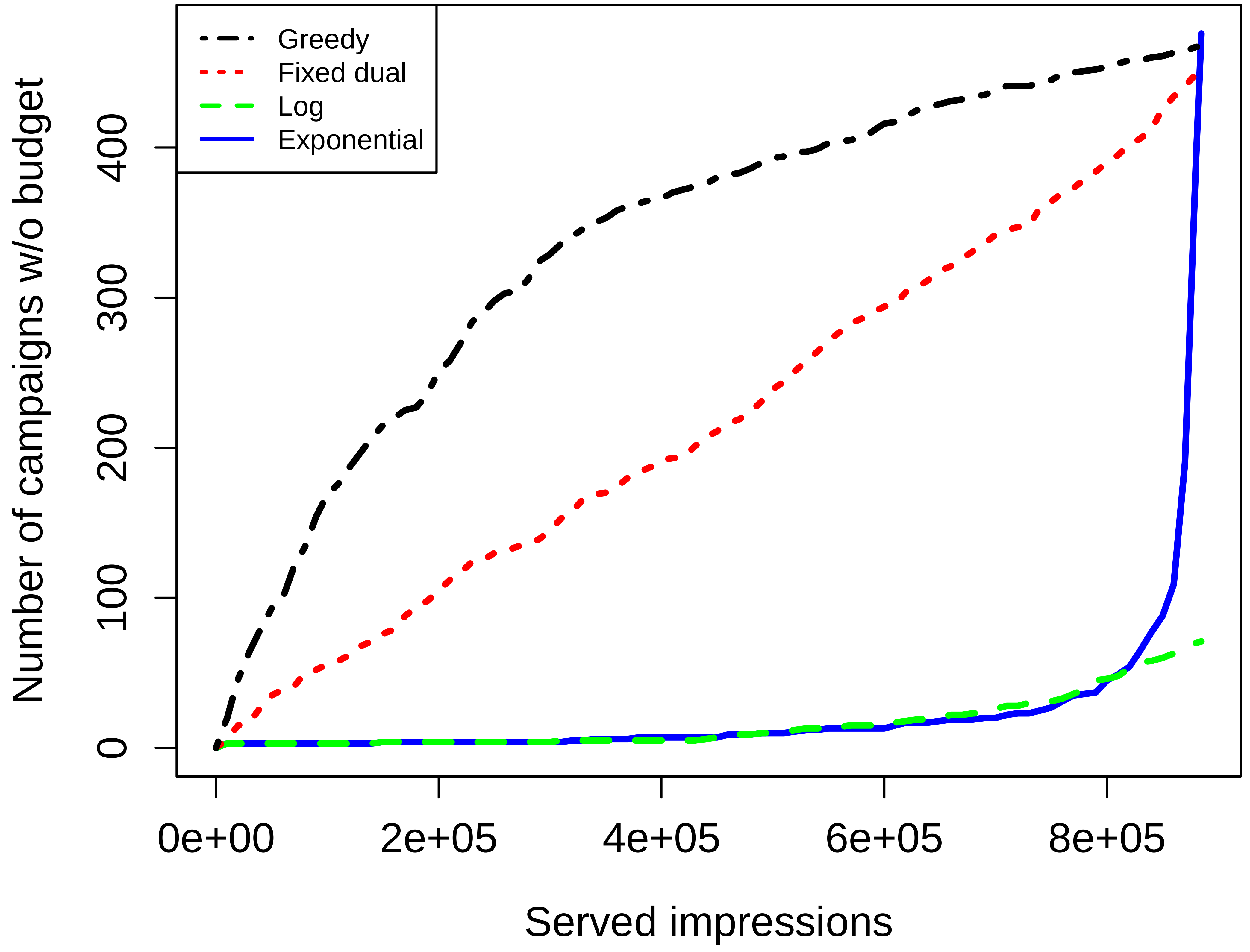
Revenues generated by different methods

- Total Revenue for impressions in T2 by Greedy and OLP with different allocation risk functions



of Out-of-Budget Advertisers

- Greedy exhausts budget of many advertisers early.
- Log penalty keeps advertisers in budget but it is very conservative.
- Exponential penalty Keeps advertisers in budget until almost the end of the timeframe.



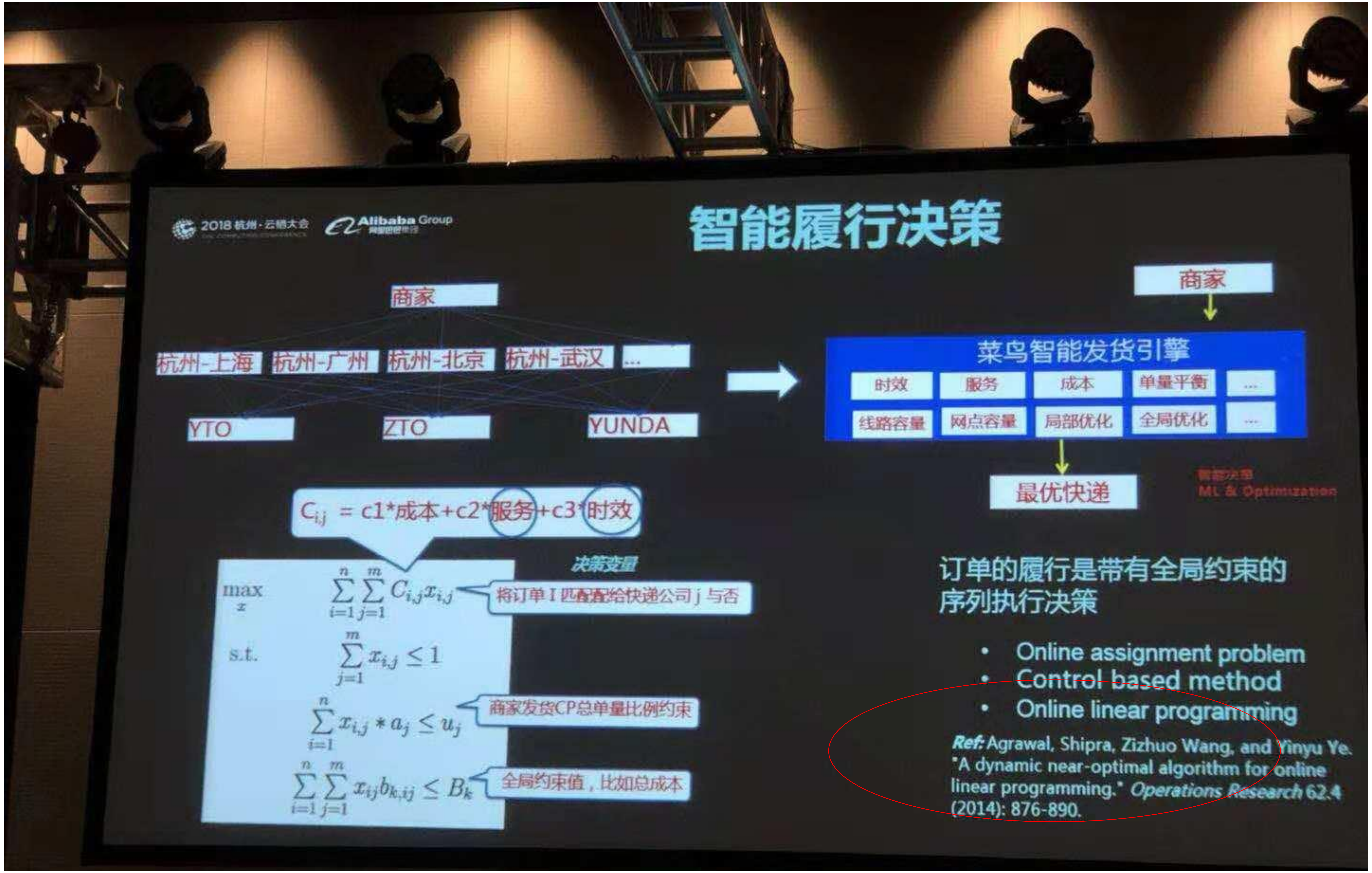
Detailed Performances

Allocation algorithm	Total Revenue	Improvement over greedy	Mid flight oob	Final oob
Greedy	$1829.94	-	366	467
Fixed dual	$1986.67	8.5%	192	452
Log	$1915.72	4.6%	5	71
Exponential	$2043.21	11.6%	7	476

oob: out of budget

https://arxiv.org/abs/1407.5710

阿里巴巴在2019年云栖大会上提到在智能履行决策上使用OLP的算法



阿里巴巴团队在2020年CIKM会议论文Online Electronic Coupon Allocation based on Real-Time User Intent Detection上提到他们设计的发红包的机制也使用了OLP的方法[2]

Spending Money Wisely: Online Electronic Coupon Allocation based on Real-Time User Intent Detection

Liangwei Li*
Liucheng Sun*
leon.llw@alibaba-inc.com
liucheng.slc@alibaba-inc.com
Alibaba Group
Hangzhou, Zhejiang

Chenwei Weng
wengchenwei.pt@alibaba-inc.com
Alibaba Group
Hangzhou, Zhejiang

Chengfu Huo
chengfu.huocf@alibaba-inc.com
Alibaba Group
Hangzhou, Zhejiang

Weijun Ren
afei@alibaba-inc.com
Alibaba Group
Hangzhou, Zhejiang

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{M} \sum_{j=1}^{N} v_{ij} x_{ij} \\
s.t. \quad & \sum_{i=1}^{M} \sum_{j=1}^{N} c_j x_{ij} \le B, \\
& \sum_{j}^{N} x_{ij} \le 1, \quad \forall i \\
& x_{ij} \ge 0, \quad \forall i, j
\end{aligned}
\tag{5}
$$

3.3 MCKP-Allocation

We adopt the primal-dual framework proposed by [2] to solve the problem defined in Equation 5. Let α and β_j be the associated dual variables respectively. After obtaining the dual variables, we can solve the problem in an online fashion. Precisely, according to the principle of the primal-dual framework, we have the following allocation rule:

$$
x_{ij} = \begin{cases} 1, & \text{where } j = \arg\max_i (v_{ij} - \alpha c_j) \\ 0, & \text{otherwise} \end{cases} \tag{9}
$$

App. II: The Online Algorithm can be Extended to Bandits with Knapsack (BwK) Applications

- For the previous problem, the decision maker first wait and observe the customer order/arm and then decide whether to accept/play it or not.
- An alternative setting is that the decision maker first decides which order/arm (s)he may accept/play, and then receive a random resource consumption vector $\mathbf{a}_j$ and yield a random reward π_j of the pulled arm.
- Known as the Bandits with Knapsacks, and it is a tradeoff exploration v.s. exploitation

$$\max \quad \sum \pi_j x_j \quad \text{s.t.} \quad \sum_j \boldsymbol{a}_j x_j \leq \boldsymbol{b}\,, \quad x_j \quad \geq 0 \qquad \forall\ j = 1, \dots, J$$

- The decision variable x_j represents the total-times of pulling the j-th arm.
- We have developed a two-phase algorithm
  - Phase I: Distinguish the optimal super-basic variables/arms from the optimal non-basic variables/arms with as fewer number of plays as possible
  - Phase II: Use the arms in the optimal face to exhaust the resource through an adaptive procedure and achieve fairness
- The algorithm achieves a problem dependent regret that bears a logarithmic dependence on the horizon T. Also, it identifies a number of LP-related parameters as the bottleneck or condition-numbers for the problem
  - Minimum non-zero reduced cost
  - Minimum singular-values of the optimal basis matrix.
- **First algorithm** to achieve the $\boldsymbol{O(\log T)}$ regret bound [Li, Sun & Y 2021 ICML] (**https://proceedings.mlr.press/v139/li21s.html**)

Topic 2. Accelerated Second-Order Methods and Applications

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

- 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:

$\| g_k \| \leq \epsilon$ (primary, first-order condition)

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

- First-order methods typically need $O(n^2\epsilon^{-2})$ operations
- Second-order methods typically need $O(n^3\epsilon^{-1.5})$ operations
- New? Yes, HSODM: a single-loop method with $O(n^2\epsilon^{-1.75})$ operations

(https://arxiv.org/abs/2211.08212)

App. III: HSODM for Policy Optimization in Reinforcement Learning

- Consider policy optimization of linearized objective in reinforcement learning

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

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

- M_k is usually a preconditioning matrix.

- 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:

$$\max \nabla L_{\theta_k}(\theta_k)^T (\theta - \theta_k)$$

Homogeneous NPG: Apply HSODM!

Dimension Reduced Second-Order Method (DRSOM)

- Motivation from Multi-Directional FOM and Subspace Method, such as CG and ADAM, DRSOM applies the trust-region method in low dimensional subspace.
- This results in a low-dimensional quadratic sub-minimization problem:
- Typically, DRSOM adopts two directions $d = -\alpha^1 \nabla f(x_k) + \alpha^2 d_k$

  where $g_k = \nabla f(x_k), H_k = \nabla^2 f(x^k), d_k = x_k - x_{k-1}$
- Then we solve a 2-d quadratic minimization problem:

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

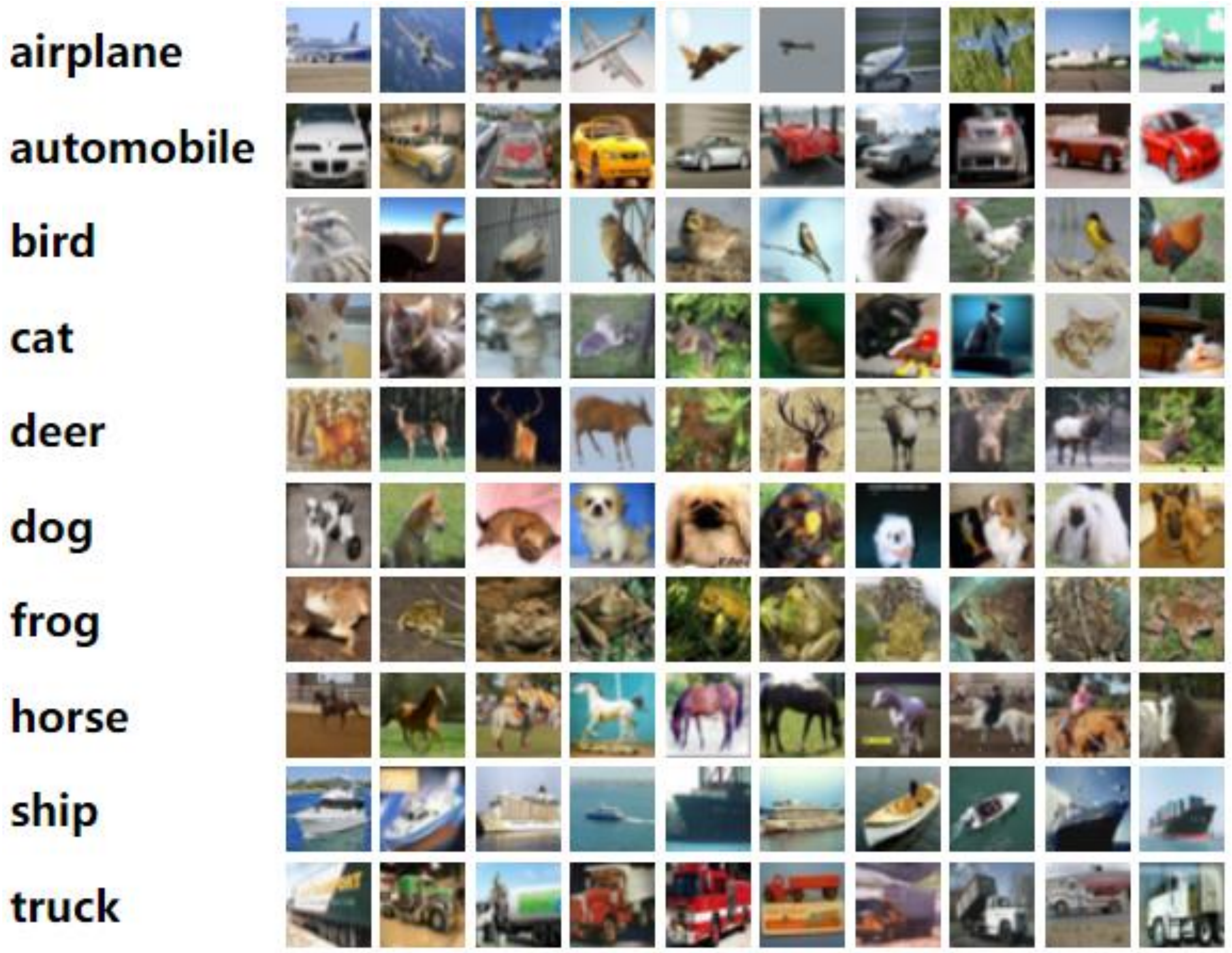
$$||\alpha||_{G_k} \leq \Delta_k$$

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

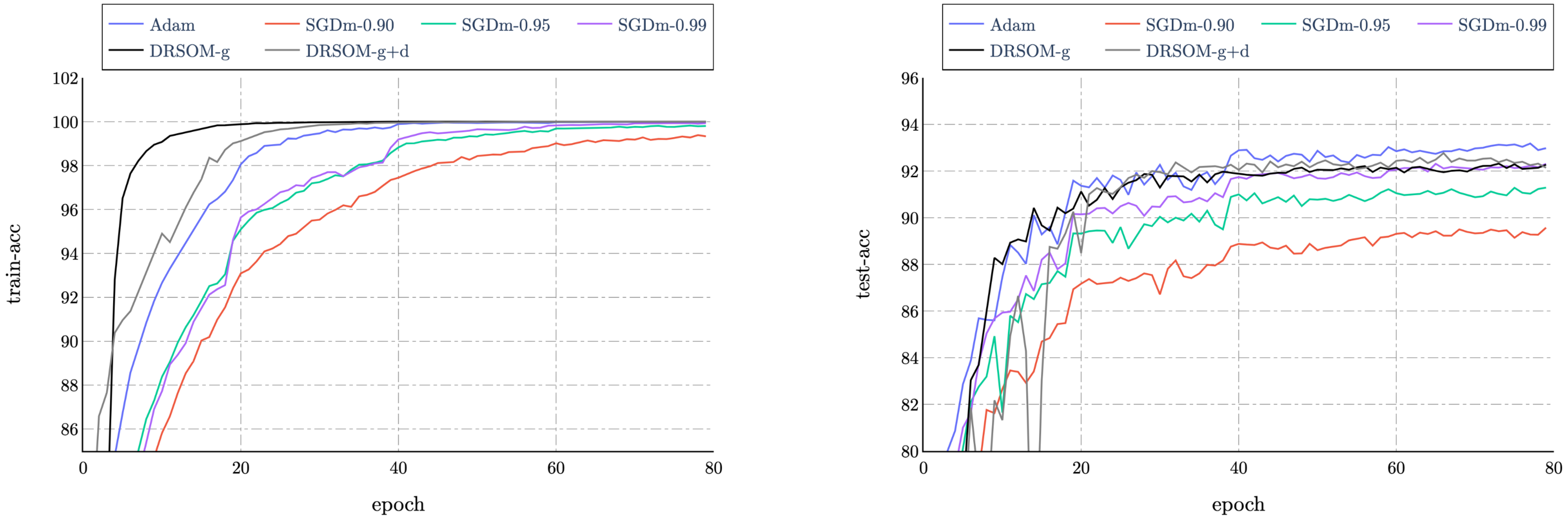
App. IV: Neural Networks and Deep Learning

To use DRSOM in machine learning problems

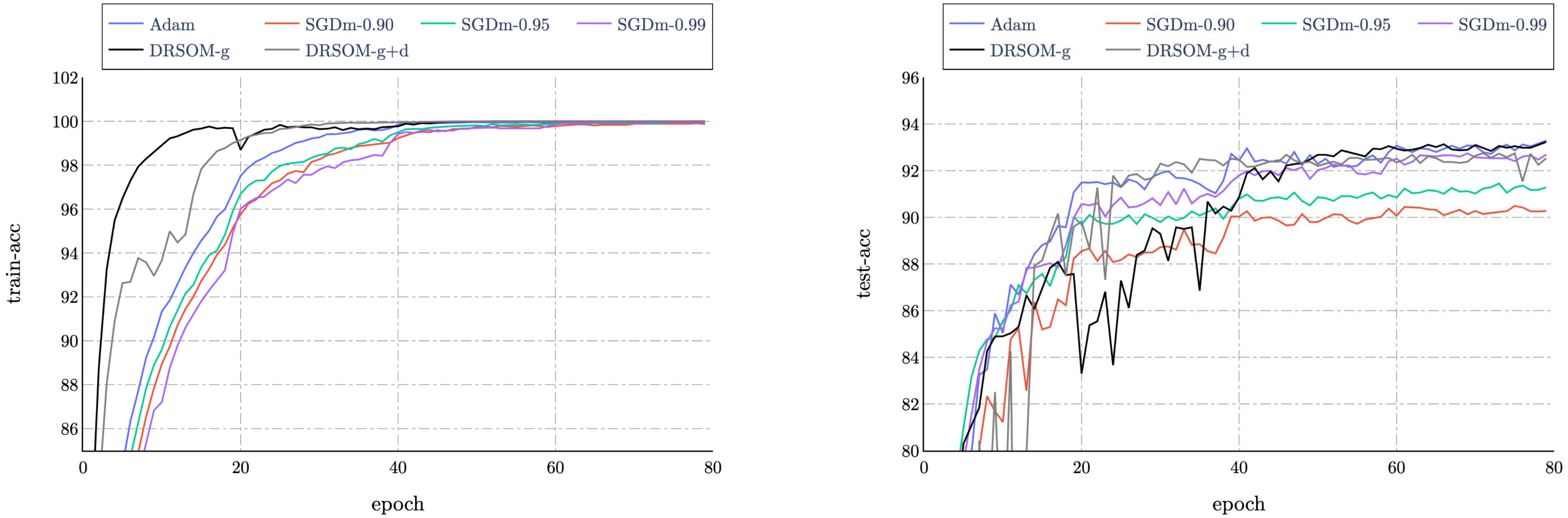
- We apply the mini-batch strategy to a vanilla DRSOM
- Use Automatic Differentiation to compute gradients
- Train ResNet18/Resnet34 Model with CIFAR 10
- Set Adam with initial learning rate 1e-3



Preliminary Results: 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 **(https://arxiv.org/abs/2208.00208)**

Pros

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

Cons

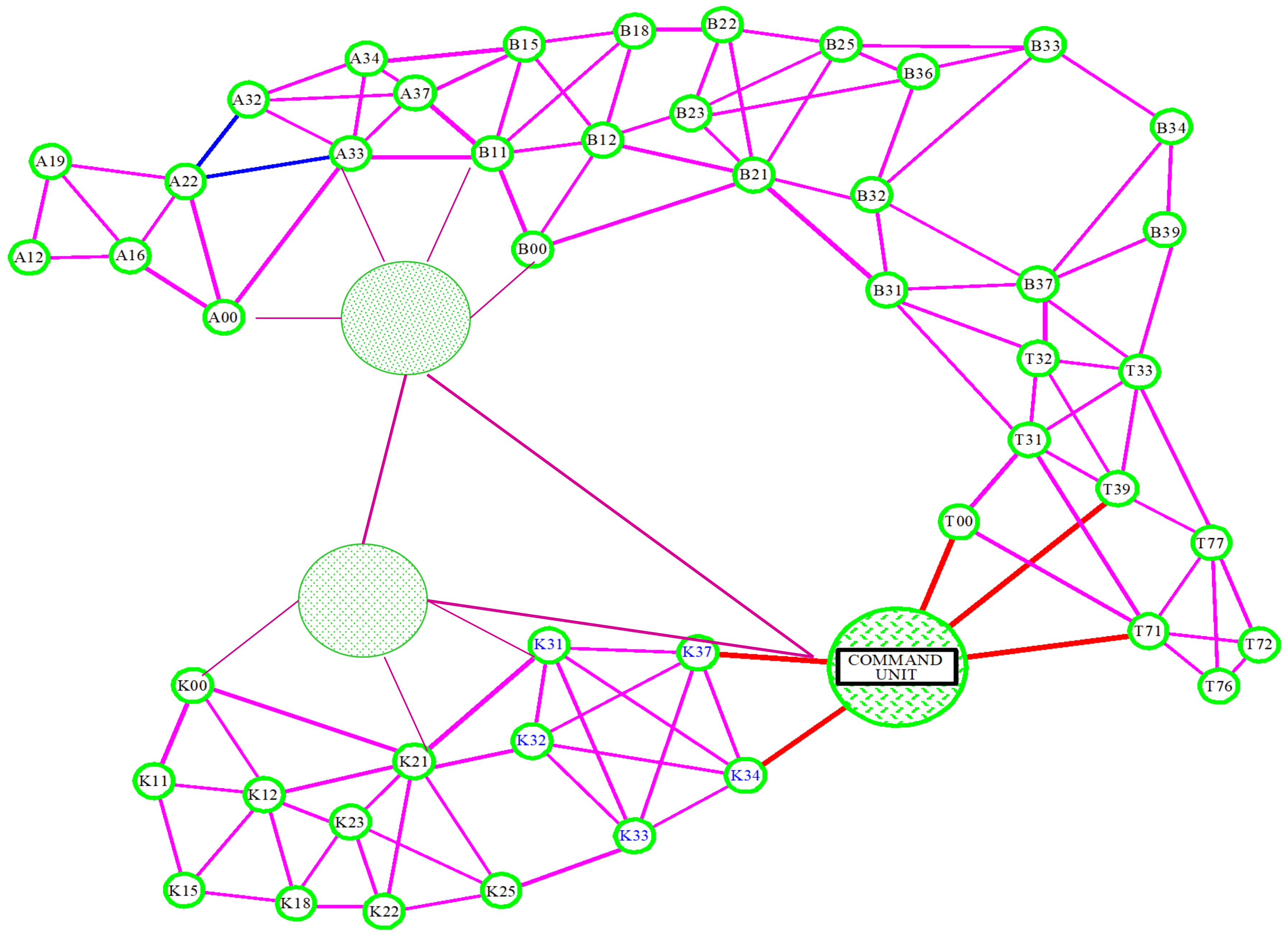
- DRSOM may over-fit the models
- Running time can benefit from Interpolation
- Single direction DRSOM is also good

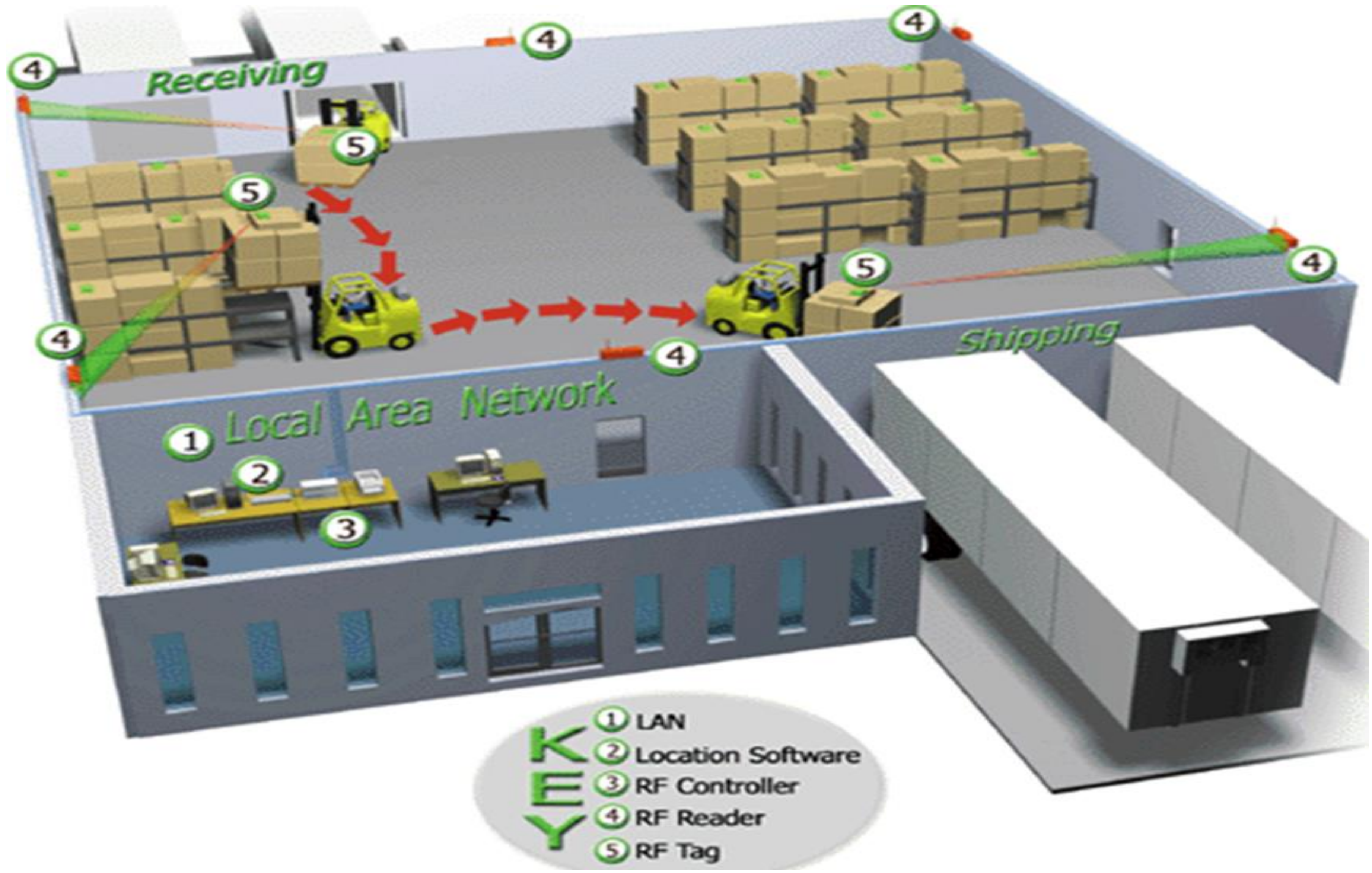
Good potential to be a standard optimizer for deep learning!

App. V: Sensor Network Location (SNL)

- **Localization**
  - –Given partial pair-wise measured distance values
  - –Given some anchors’ positions
  - –Find locations of all other sensors that fit the measured distance values

This is also called graph realization on a fixed dimension Euclidean space





Mathematical Formulation of Sensor Network Location (SNL)

- Consider Sensor Network Location (SNL)

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

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

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

- Alternatively, one can solve SNL by the nonconvex nonlinear least square (NLS) problem

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

Semidefinite Programming Relaxation

<u>Step 1</u>: Linearization

$$\|x_i - x_j\|^2 = \underbrace{\cancel{x_i^T x_i}}_{Y_{ii}} - 2\underbrace{\cancel{x_i^T x_j}}_{Y_{ij}} + \underbrace{\cancel{x_j^T x_j}}_{Y_{jj}}$$

$$\|a_k - x_j\|^2 = a_k^T a_k - 2a_k^T x_j + \underbrace{\cancel{x_j^T x_j}}_{Y_{jj}}$$

Tighten: $Y = X^T X$, $X=[x_1, \ldots, x_n]$

<u>Step 2:</u> Relax

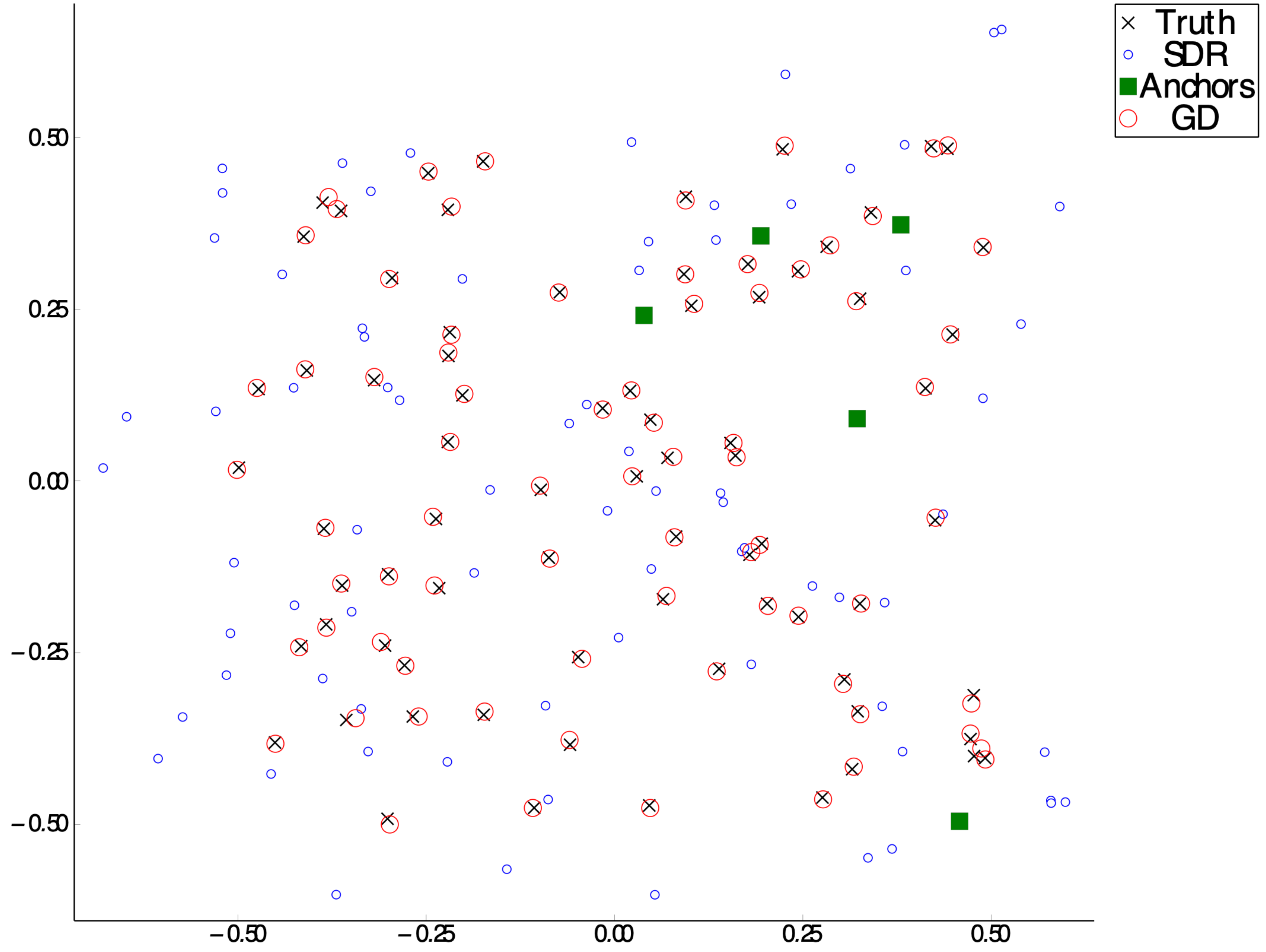
$$Y \geq X^T X \Leftrightarrow Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \geq PSD$$

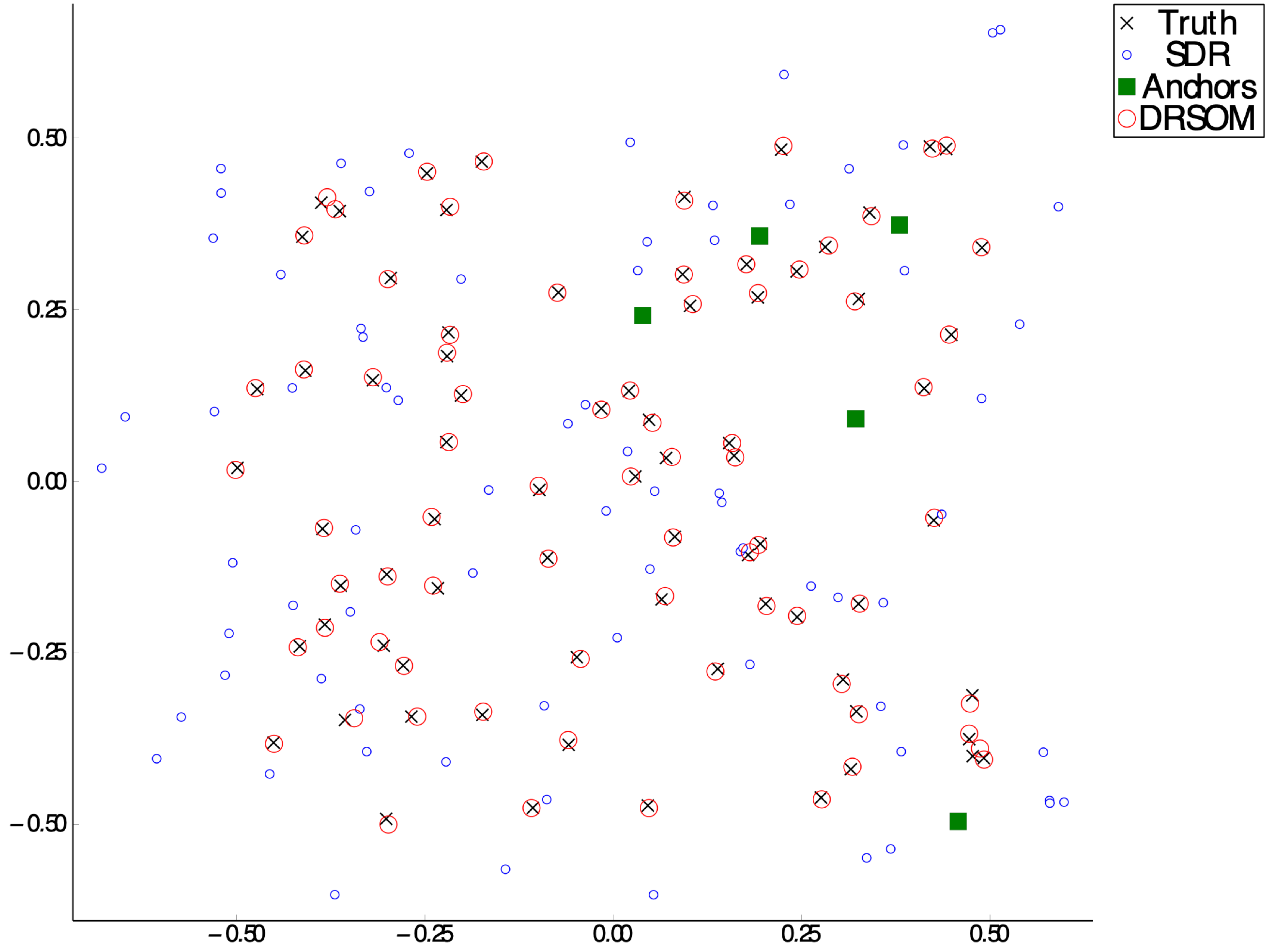
This is a conic linear program which is a convex optimization problem, but $O(n^{3.5}\log(\epsilon^{-1}))$

Biswas and Y 2004, So and Y 2005

Sensor Network Location (SNL) I

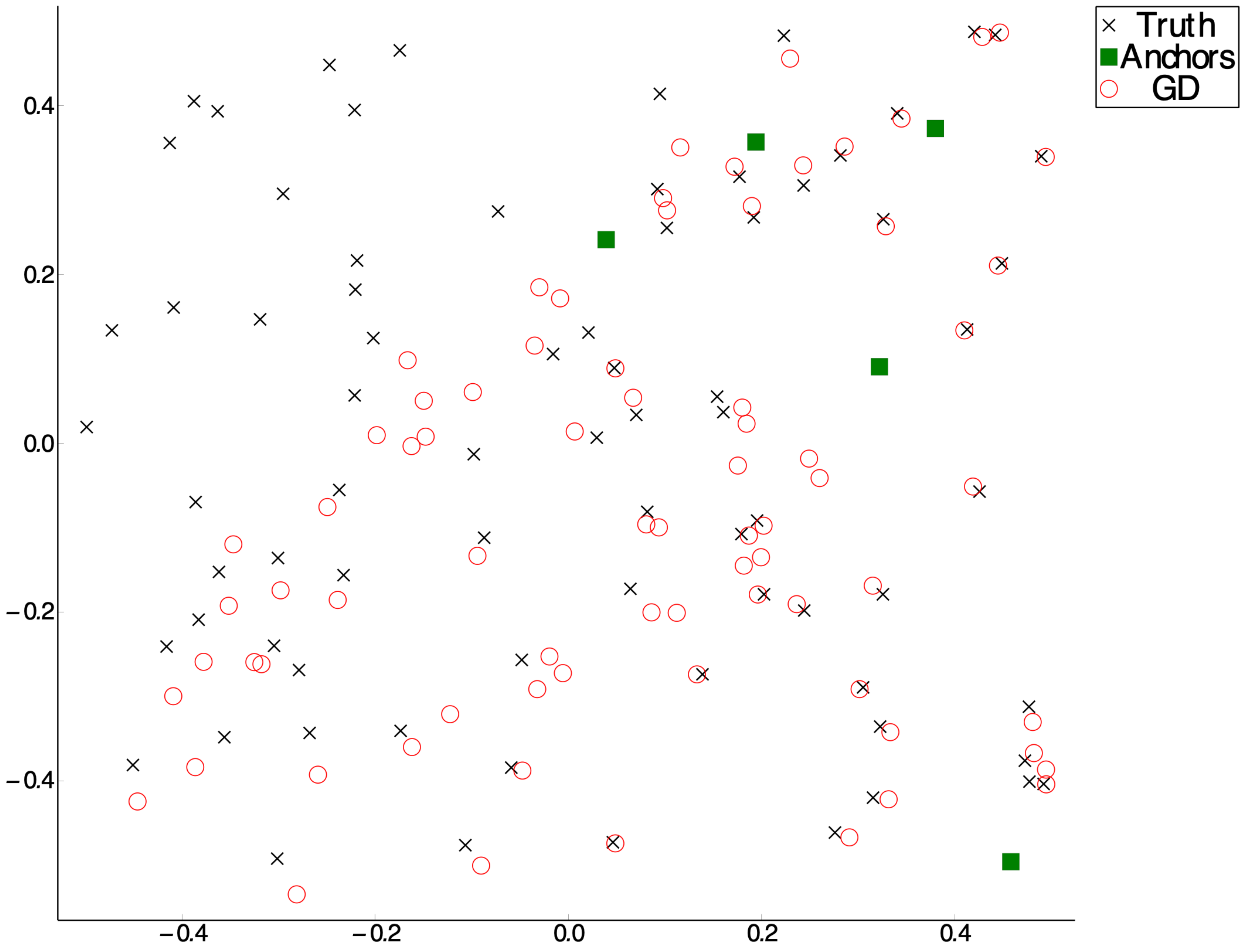
- Graphical results using SDP relaxation (Biswas&Y 2004, SO&Y 2007) to initialize the NLS
- n = 80, m = 5 (anchors), radio range = 0.5, degree = 25, noise factor = 0.05
- Both Gradient Descent and DRSOM can find good solutions !

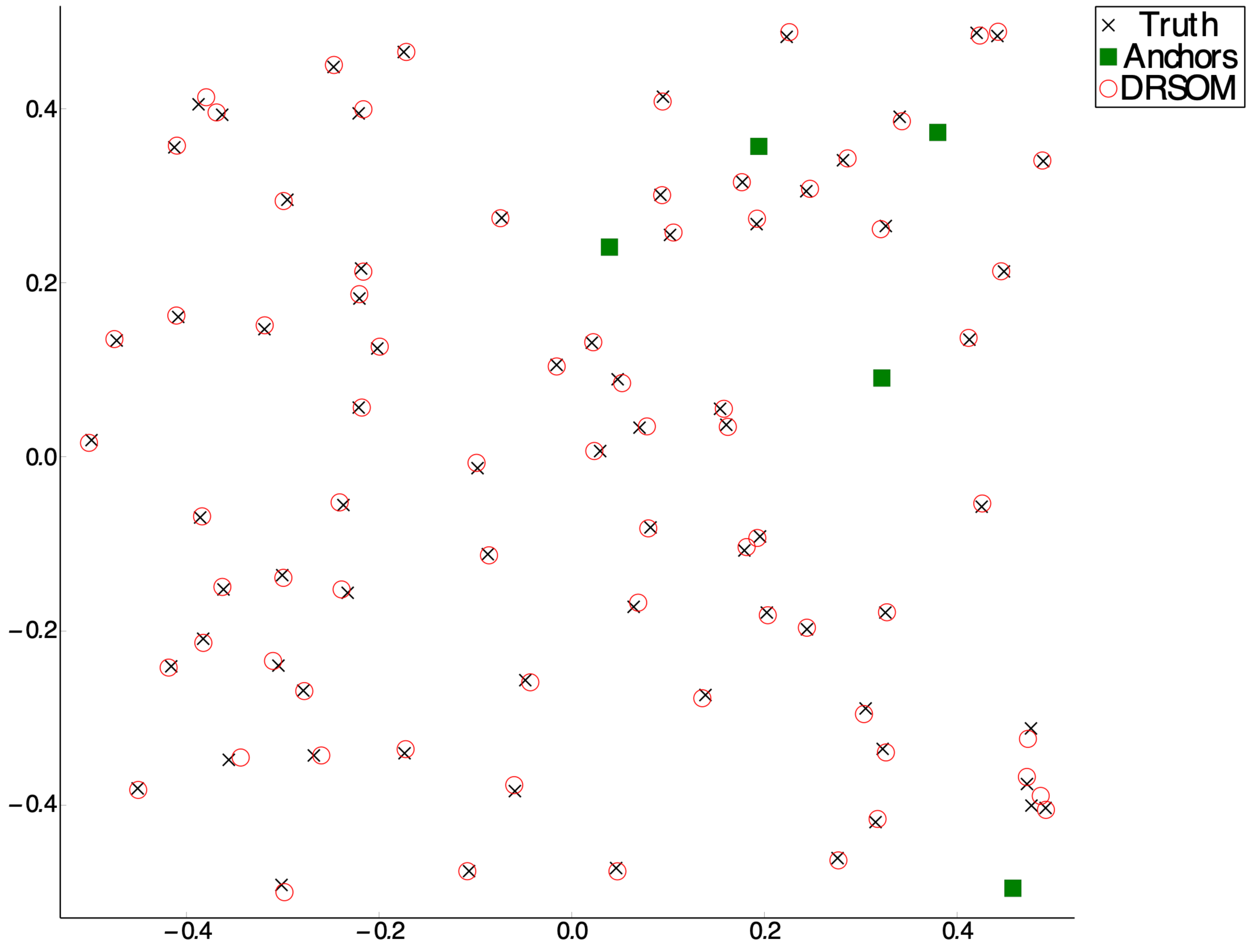




Sensor Network Location (SNL) II

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





Sensor Network Location, Large-Scale Instances I

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

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

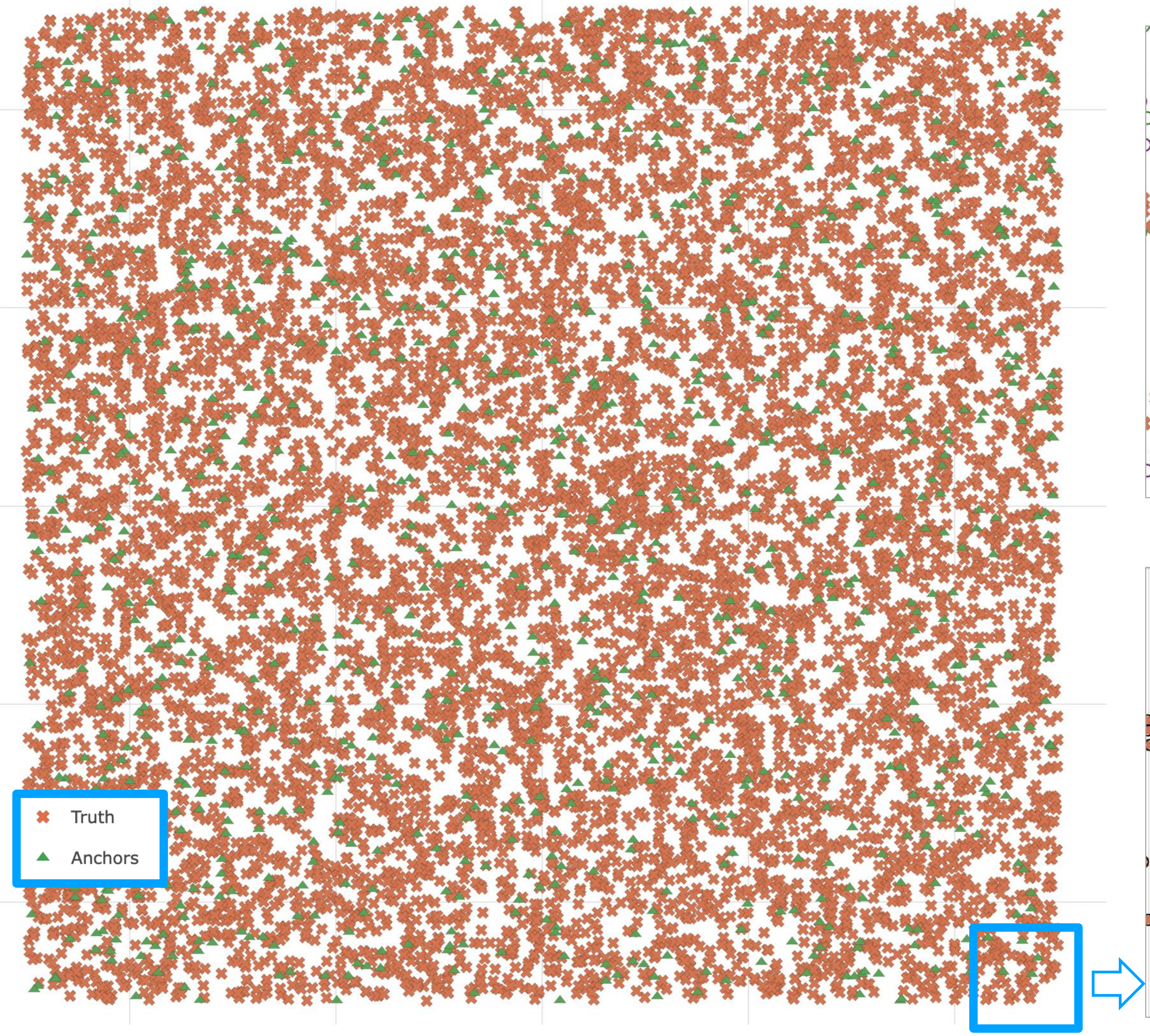
n	m	$\lvert E\rvert$	t		
			CG	DRSOM	GD
500	50	2.2e+04	1.7e+01	1.1e+01	2.3e+01
1000	80	4.6e+04	7.3e+01	3.9e+01	1.8e+02
2000	120	9.4e+04	2.5e+02	1.4e+02	1.1e+03
3000	150	1.4e+05	6.5e+02	1.4e+02	-
4000	400	1.8e+05	1.3e+03	5.0e+02	-
6000	600	2.7e+05	2.0e+03	1.1e+03	-
10000	1000	4.5e+05	-	2.2e+03	-

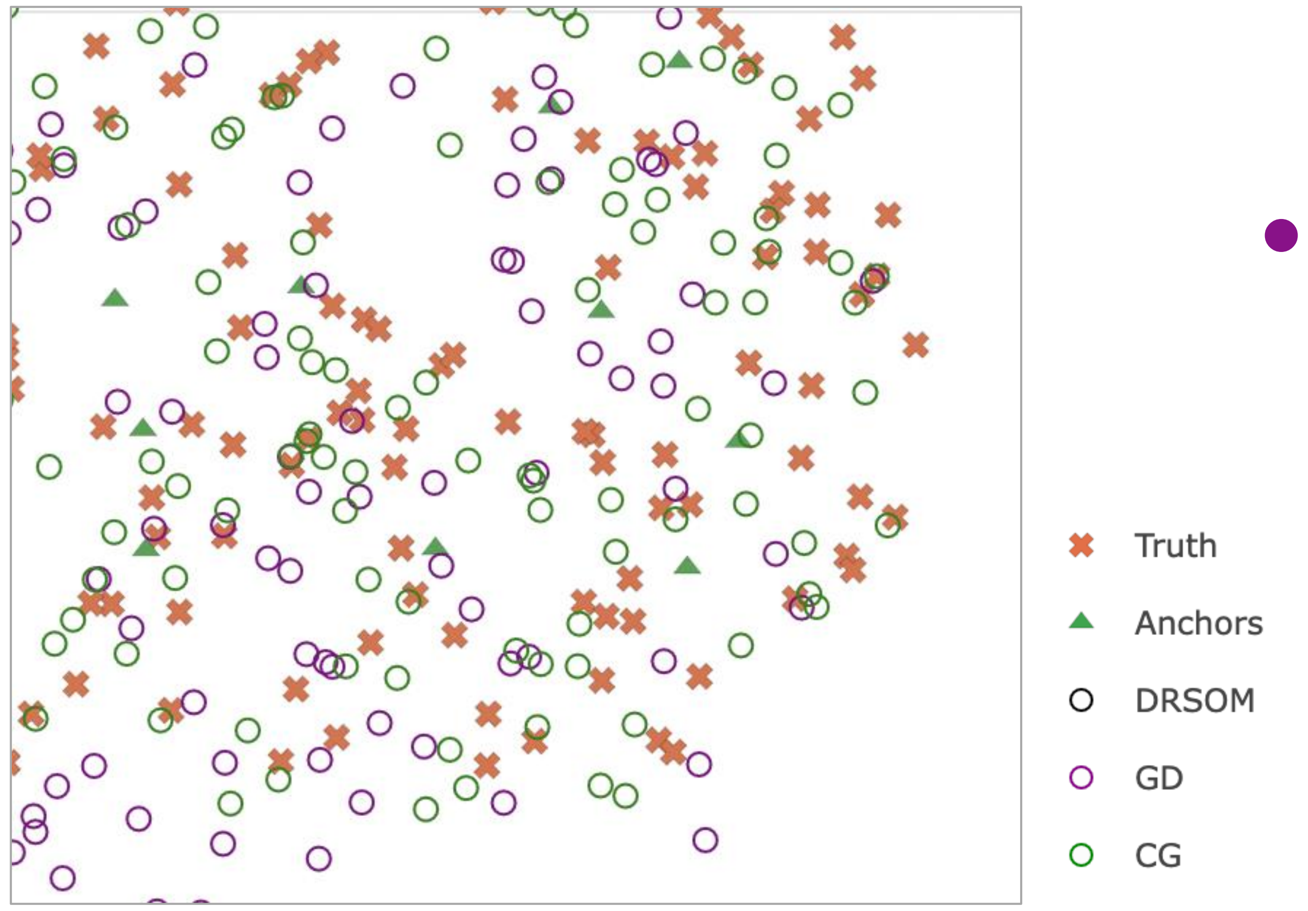
Table 2: Running time of CG, DRSOM, and GD on SNL instances of different problem size, $|E|$ denotes the number of QCQP constraints. "-" means the algorithm exceeds 3,000s.

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

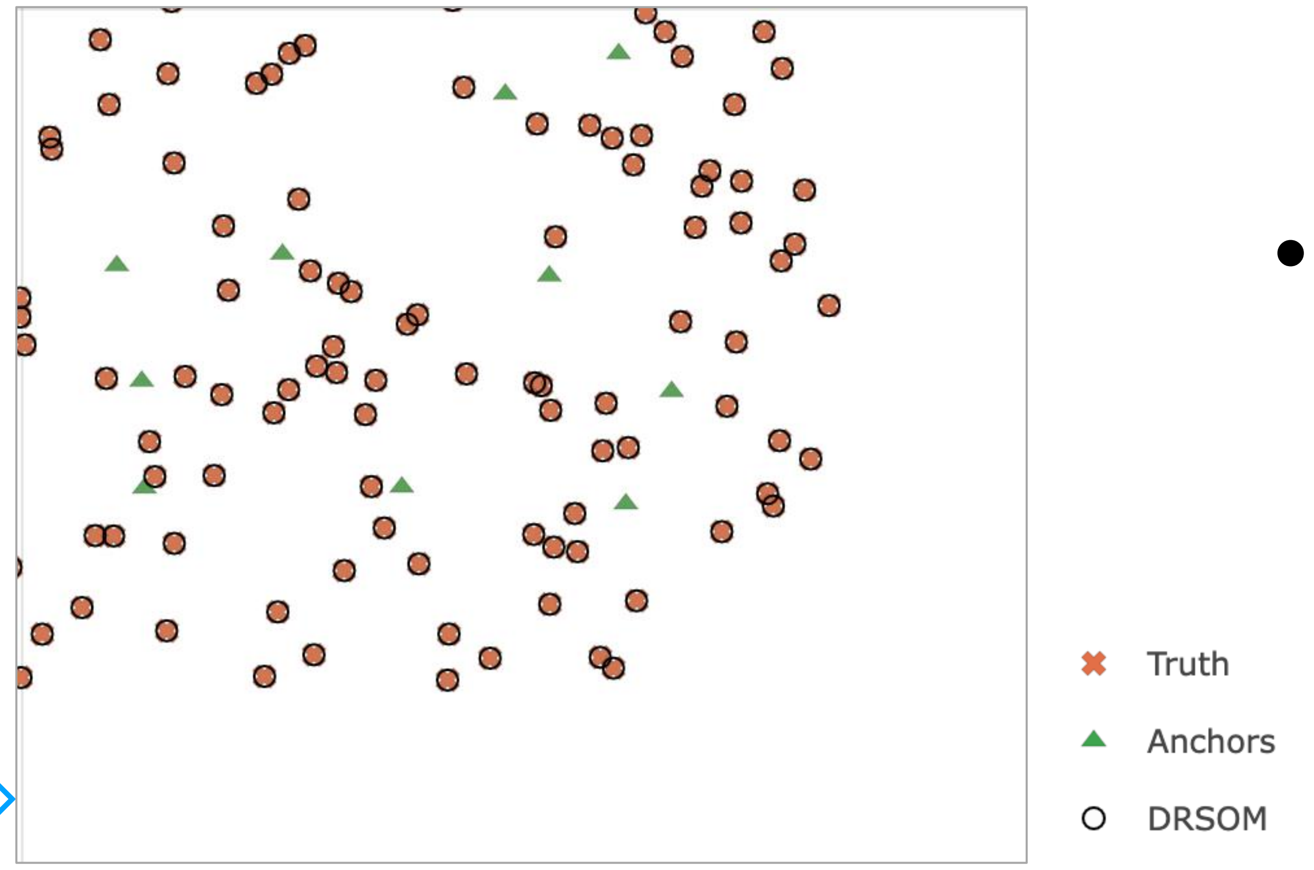
Sensor Network Location, Large-Scale Instances II

- 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**

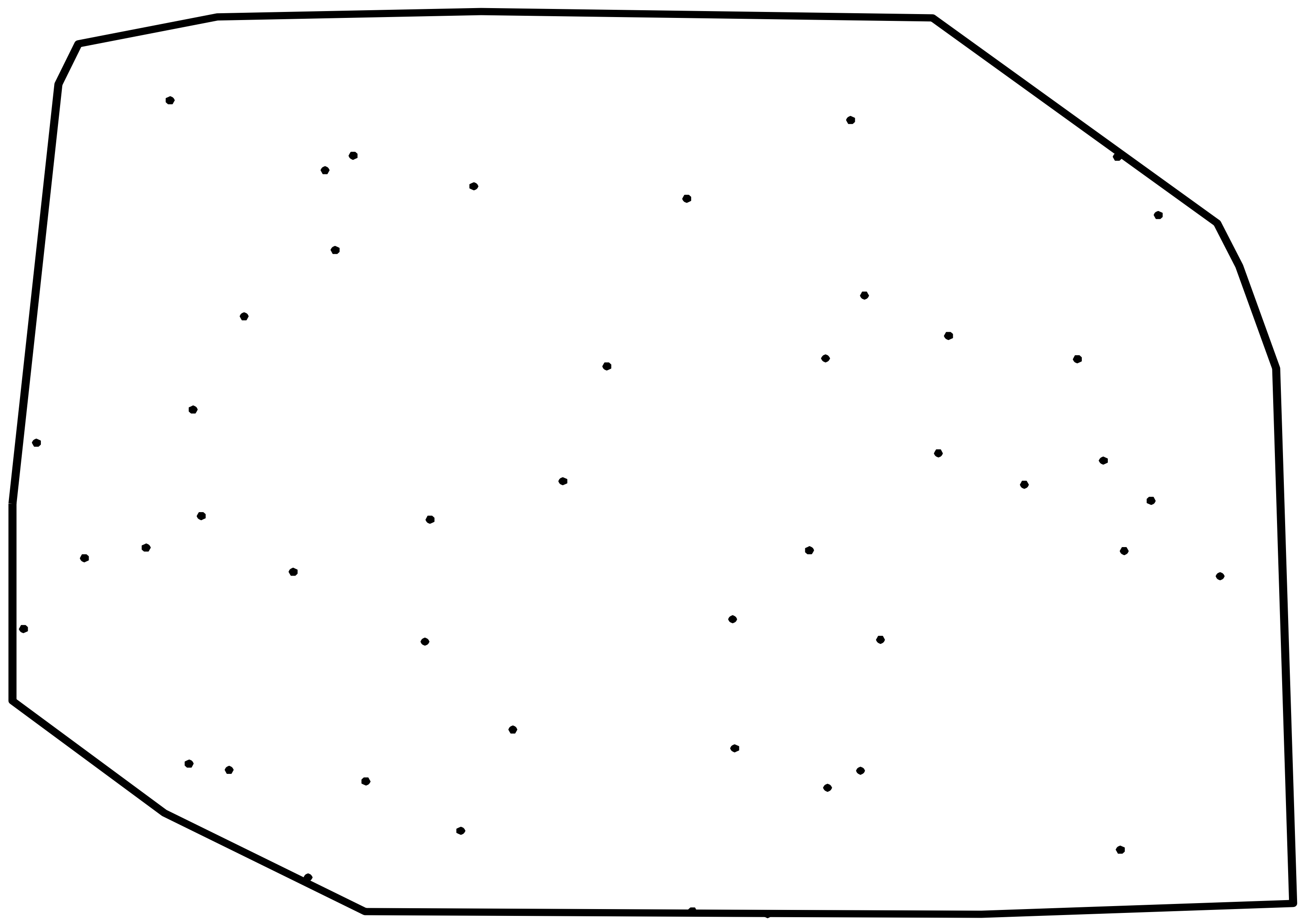


- **DRSOM can converge to $|g_k| \leq 1e^{-5}$ in 2,200s**

Sensor Network Online Tracking, 2D and 3D

Topic 3: Equitable Covering & Partition – Divide and Conquer (Carlsson et al. 2009)

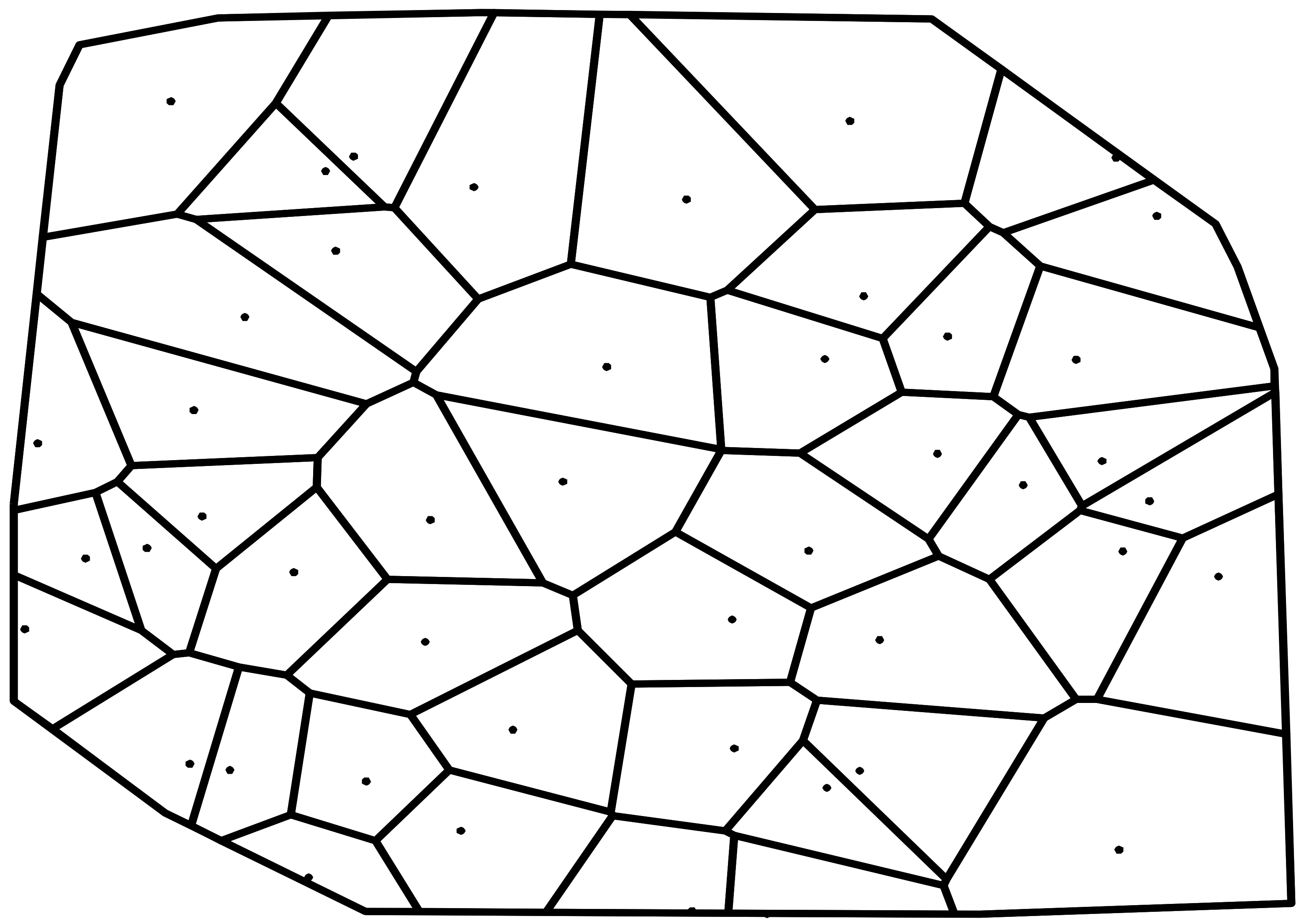
Problem Statement: Divide-Conquer



n points are scattered inside a convex polygon P (in 2D) with m vertices. Does there exist a partition of P into n sub-regions satisfying the following:

- Each sub-region is a convex polygon
- Each sub-region contains one point
- All sub-regions have equal area

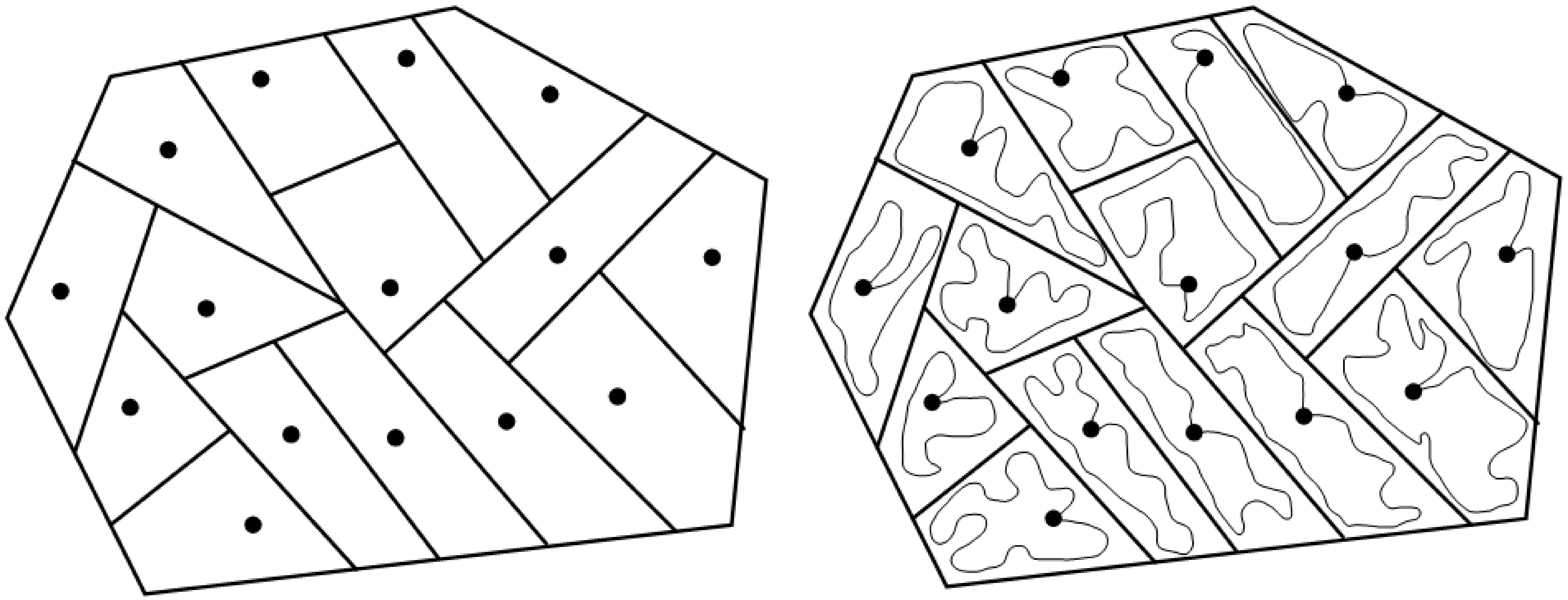
Related ML Problem: Voronoi Diagram



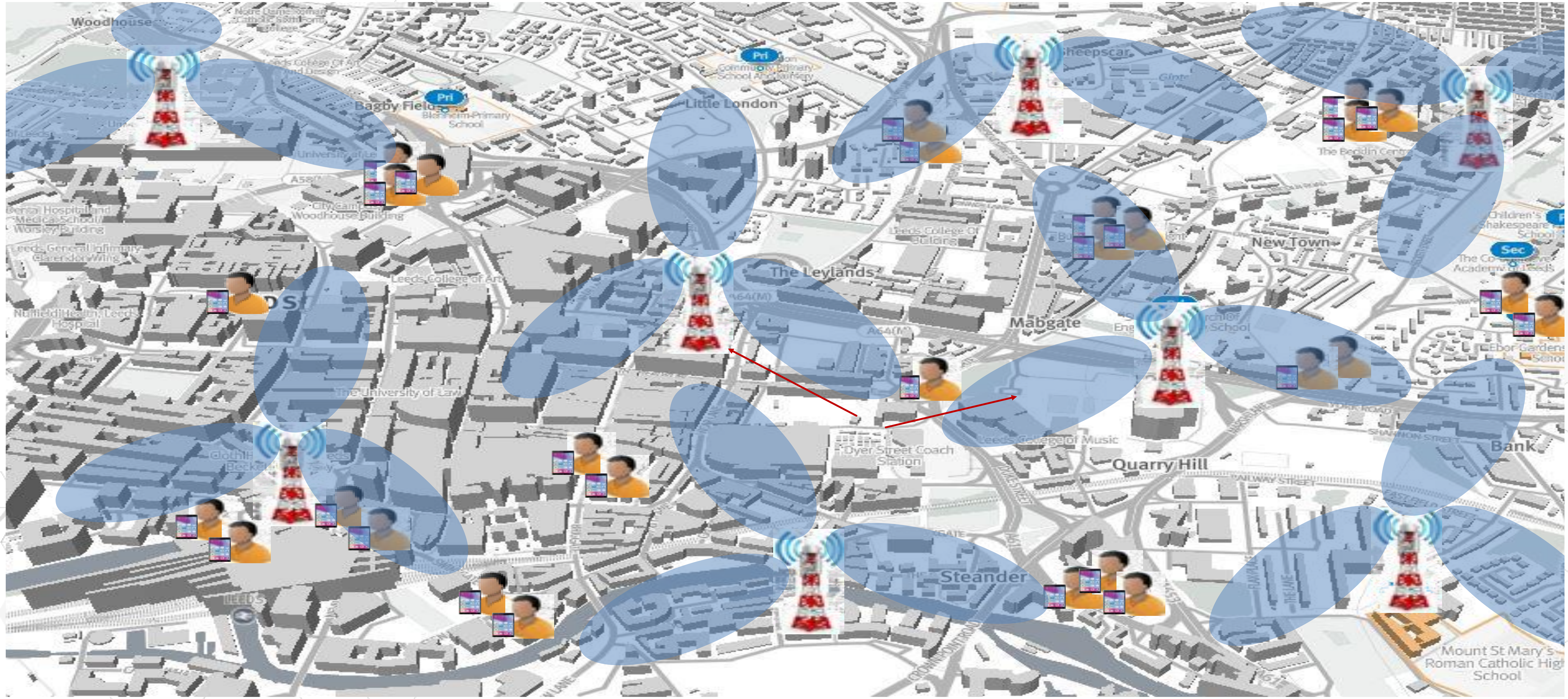
In the *Voronoi Diagram*, we satisfy the first two properties (each sub-region is convex and contains one point), but the sub-regions have different areas.

Our Result

Not only such an equitable partition always exists, but also we can find it exactly in running time O(Nn log N), where $N = m + n$.



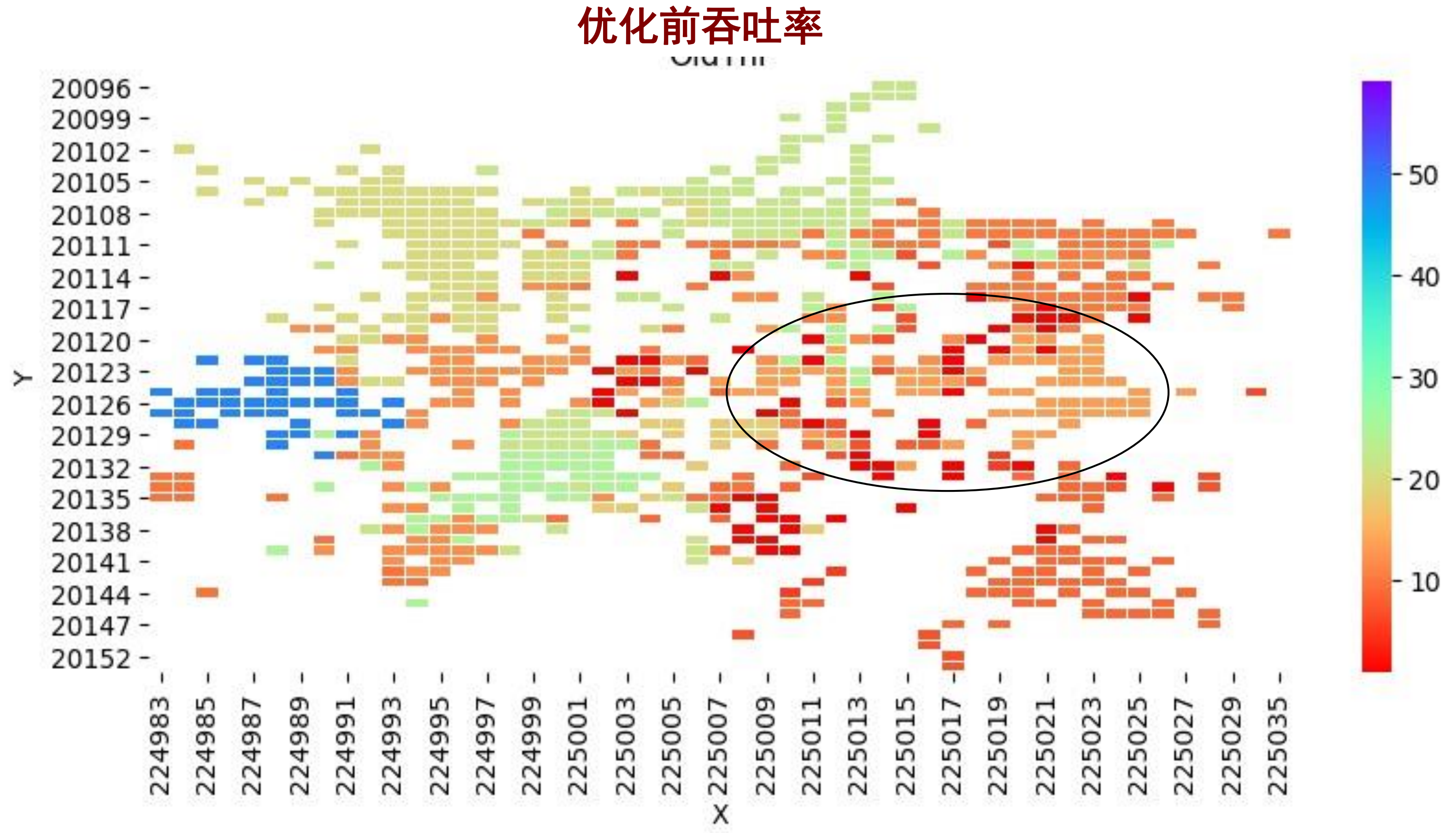
App. VI: Wireless Tower - Resource Allocation

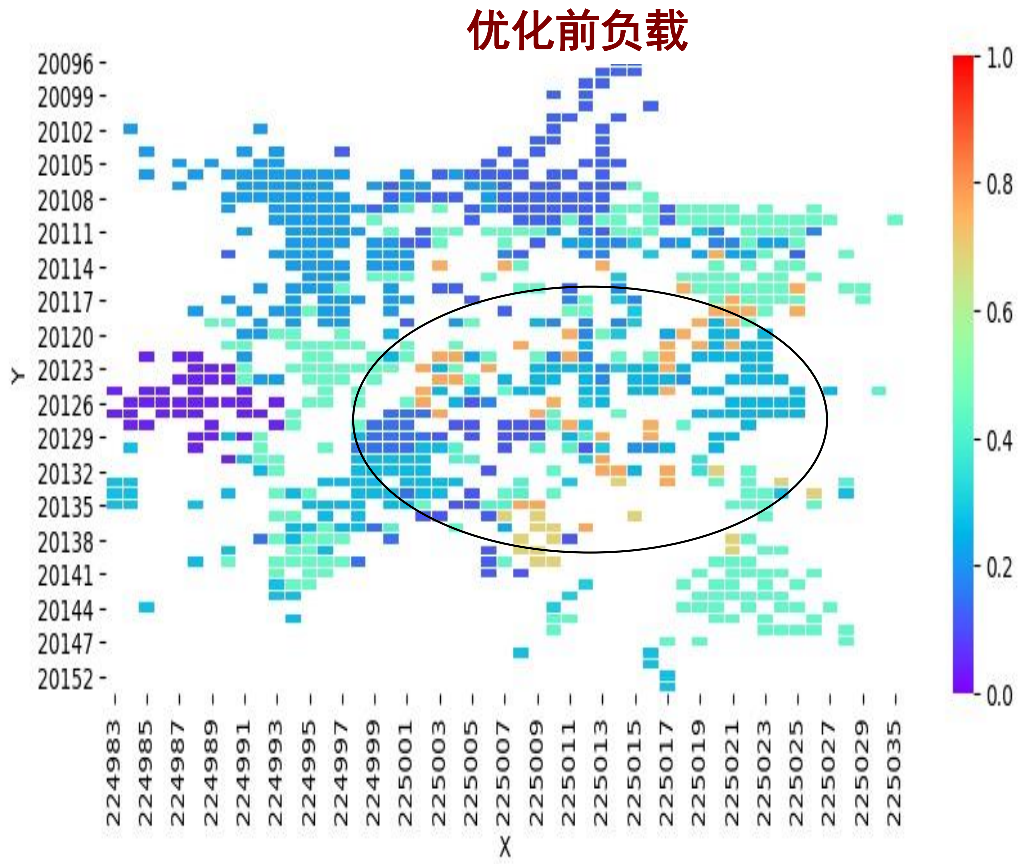


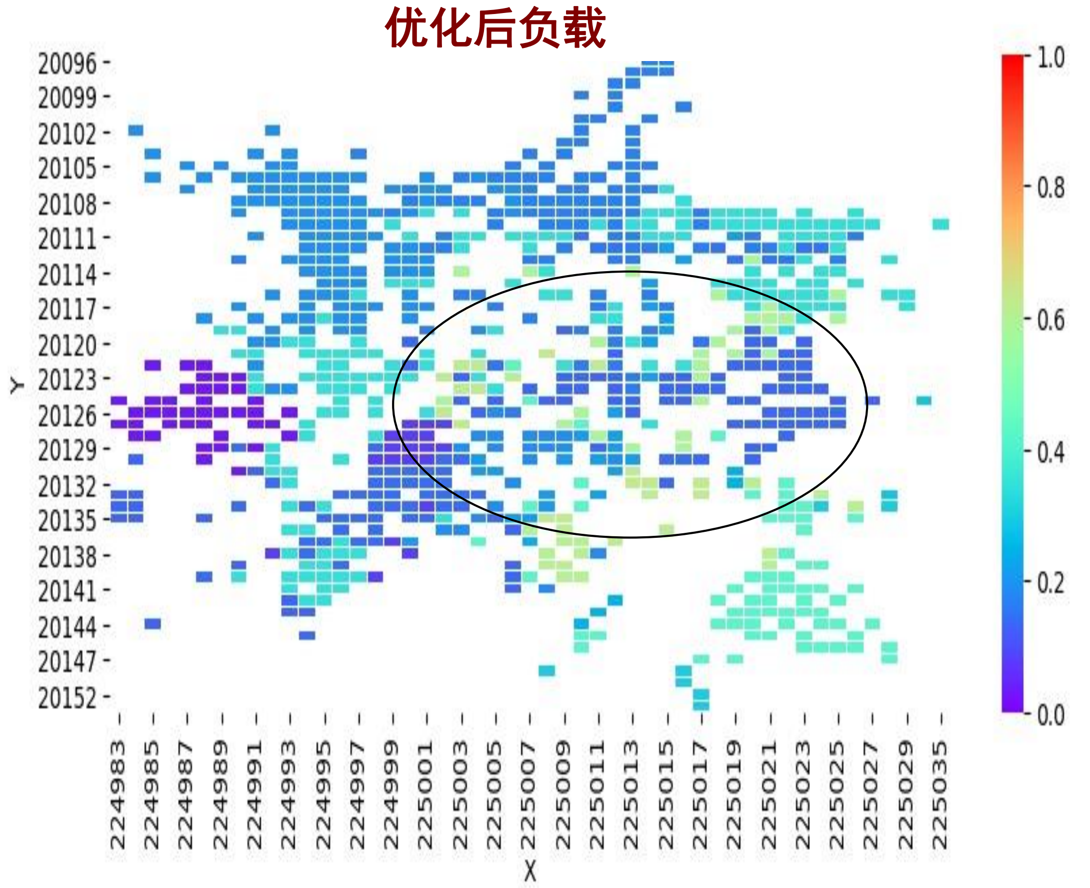
Preliminary Test Result—Effectiveness

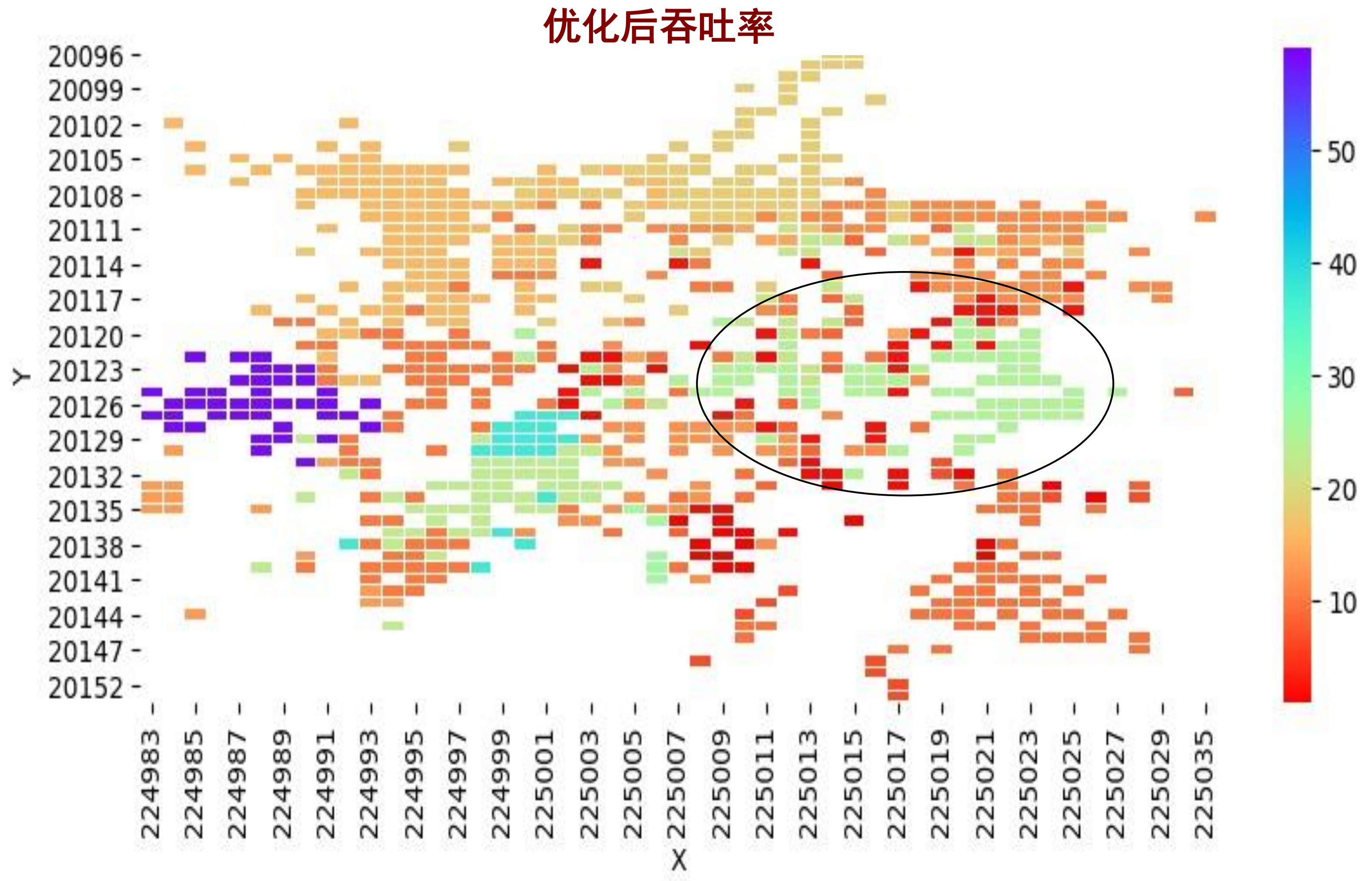
基于真实商用网络进行模型优化效果的测试验证验证统计结果：

	小区数	时段	区域平均负载	区域平均吞吐率(Mb/S)	高负载小区负载	高负载小区吞吐率(Mb/S)
优化前	27	中午及晚共6小时	31%	5.3	68%	2.3
优化后			30%	6.12(提升15%)	66%	2.8(提升22%)
优化前		晚7时话务高峰	37%	3.9	77%	1.6
优化后			33%	5.2(提升33%)	68%	2.1(提升32%)

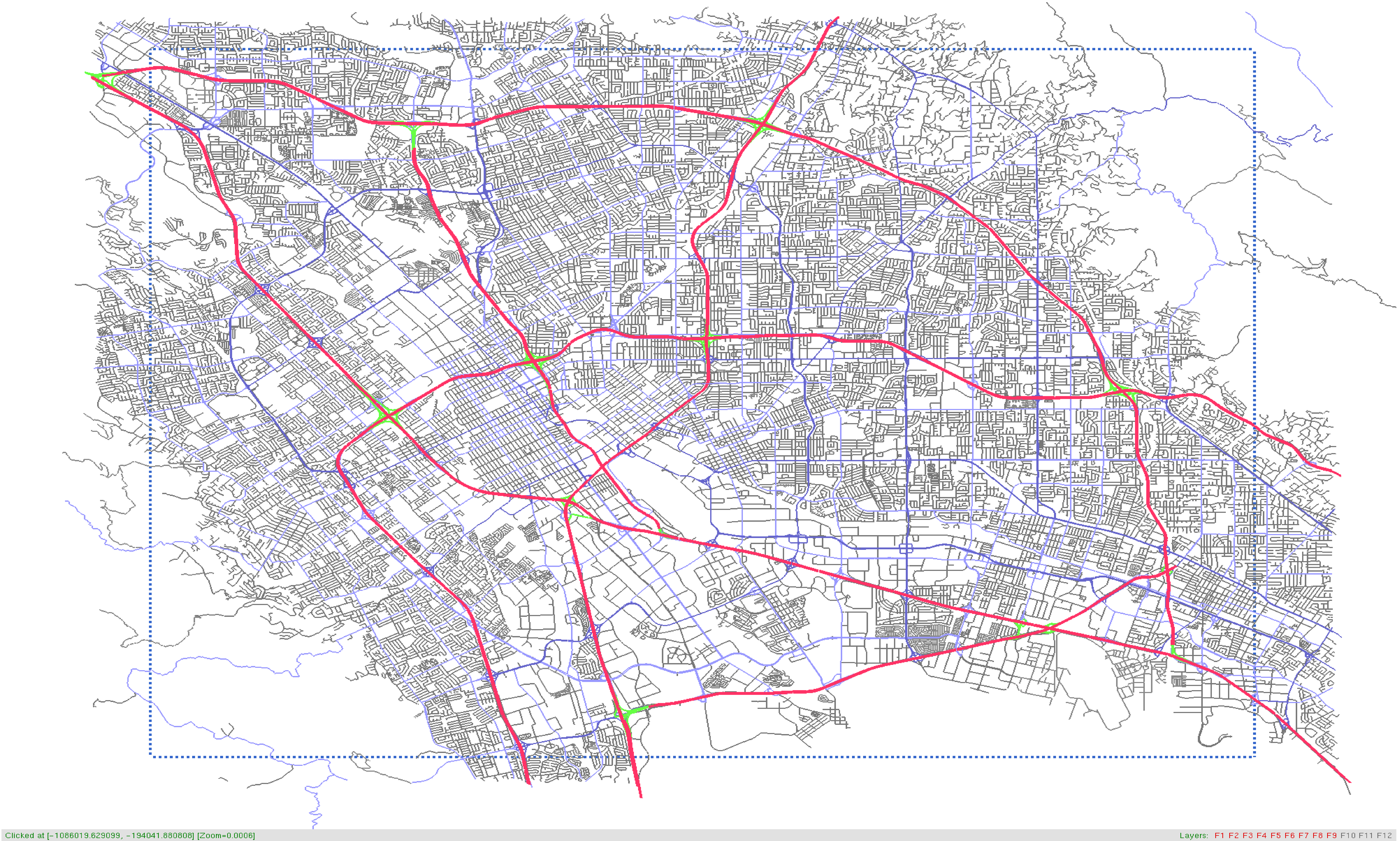








App. VII: Street View Application
Map-Making



Overall Takeaways

It is possible to maker online decisions for quantitative decision models with performance guarantees close to that of the offline decision-making with complete information

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

Decomposition (Divide and Conquer) helps solving large-scale optimization problems

- THANK YOU