Optimization in Data Science and Machine Learning/Decision-Making

SUST, March 23, 2023

Yinyu Ye
Stanford University and CUHKSZ (Sabbatical Leave)

Today's Talk

1. Online Linear Programming Algorithms and Applications

2. Accelerated Second-Order Methods for Nonlinear Optimization and Applications

3. Mixed Integer Linear Programming Solver and Applications

4. Equitable Covering & Partition – Divide and Conquer and Applications

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）
- What is OLP？
  - 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 → 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 → 决策 → 反馈

算法　　潜在模型

需要一边学习，一边优化

From Zizhuo Wang

Linear Programming and LP Giants won Nobel Prize…

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

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

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

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

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

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.

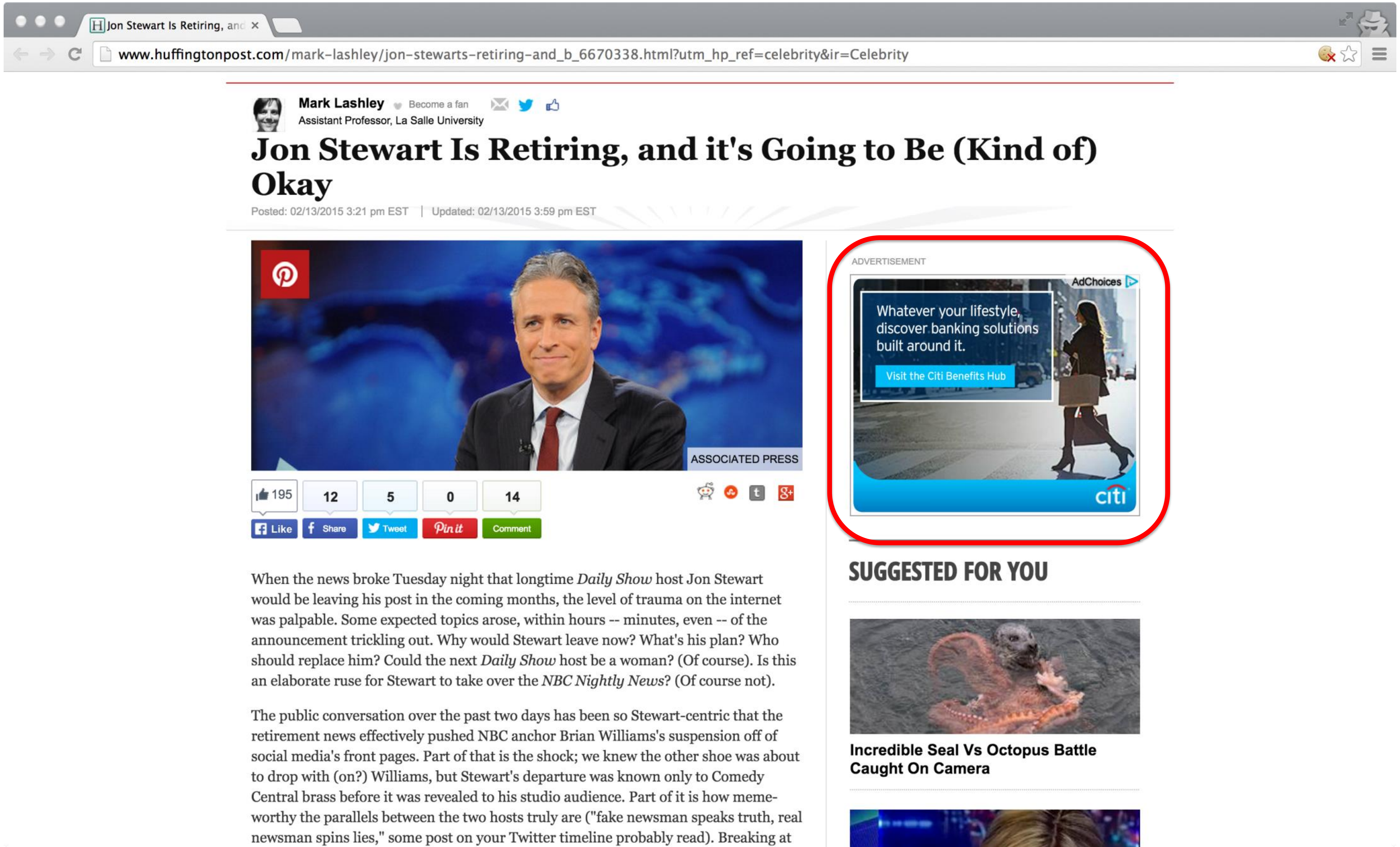
Bid #	$100	$30	….	…	…	Inventory
Decision	x1	x2				
Pants	1	0	….	…	…	100
Shoes	1	0				50
T-Shirts	0	1				500
Jackets	0	0				200
Hats	1	1	…	…	…	1000

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

App. I: Online Matching for Display Advertising



Jon Stewart Is Retiring, and

www.huffingtonpost.com/mark-lashley/jon-stewarts-retiring-and_b_6670338.html?utm_hp_ref=celebrity&ir=Celebrity

Mark Lashley Become a fan
Assistant Professor, La Salle University

Jon Stewart Is Retiring, and it's Going to Be (Kind of) Okay

Posted: 02/13/2015 3:21 pm EST | Updated: 02/13/2015 3:59 pm EST

ASSOCIATED PRESS

195 Like | 12 Share | 5 Tweet | 0 Pin it | 14 Comment

When the news broke Tuesday night that longtime *Daily Show* host Jon Stewart would be leaving his post in the coming months, the level of trauma on the internet was palpable. Some expected topics arose, within hours -- minutes, even -- of the announcement trickling out. Why would Stewart leave now? What's his plan? Who should replace him? Could the next *Daily Show* host be a woman? (Of course). Is this an elaborate ruse for Stewart to take over the *NBC Nightly News*? (Of course not).

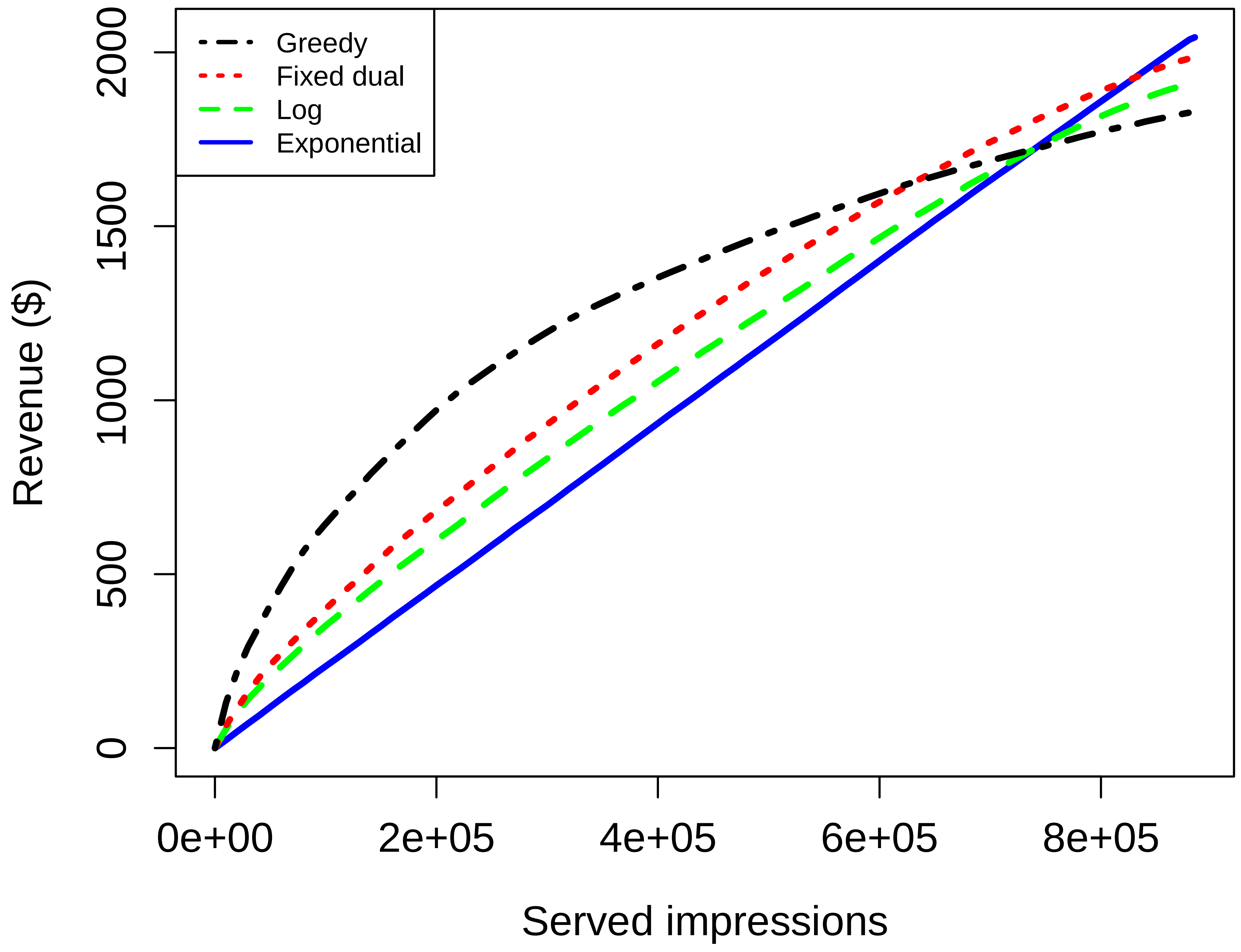
The public conversation over the past two days has been so Stewart-centric that the retirement news effectively pushed NBC anchor Brian Williams's suspension off of social media's front pages. Part of that is the shock; we knew the other shoe was about to drop with (on?) Williams, but Stewart's departure was known only to Comedy Central brass before it was revealed to his studio audience. Part of it is how meme-worthy the parallels between the two hosts truly are ("fake newsman speaks truth, real newsman spins lies," some post on your Twitter timeline probably read). Breaking at

SUGGESTED FOR YOU

Incredible Seal Vs Octopus Battle Caught On Camera

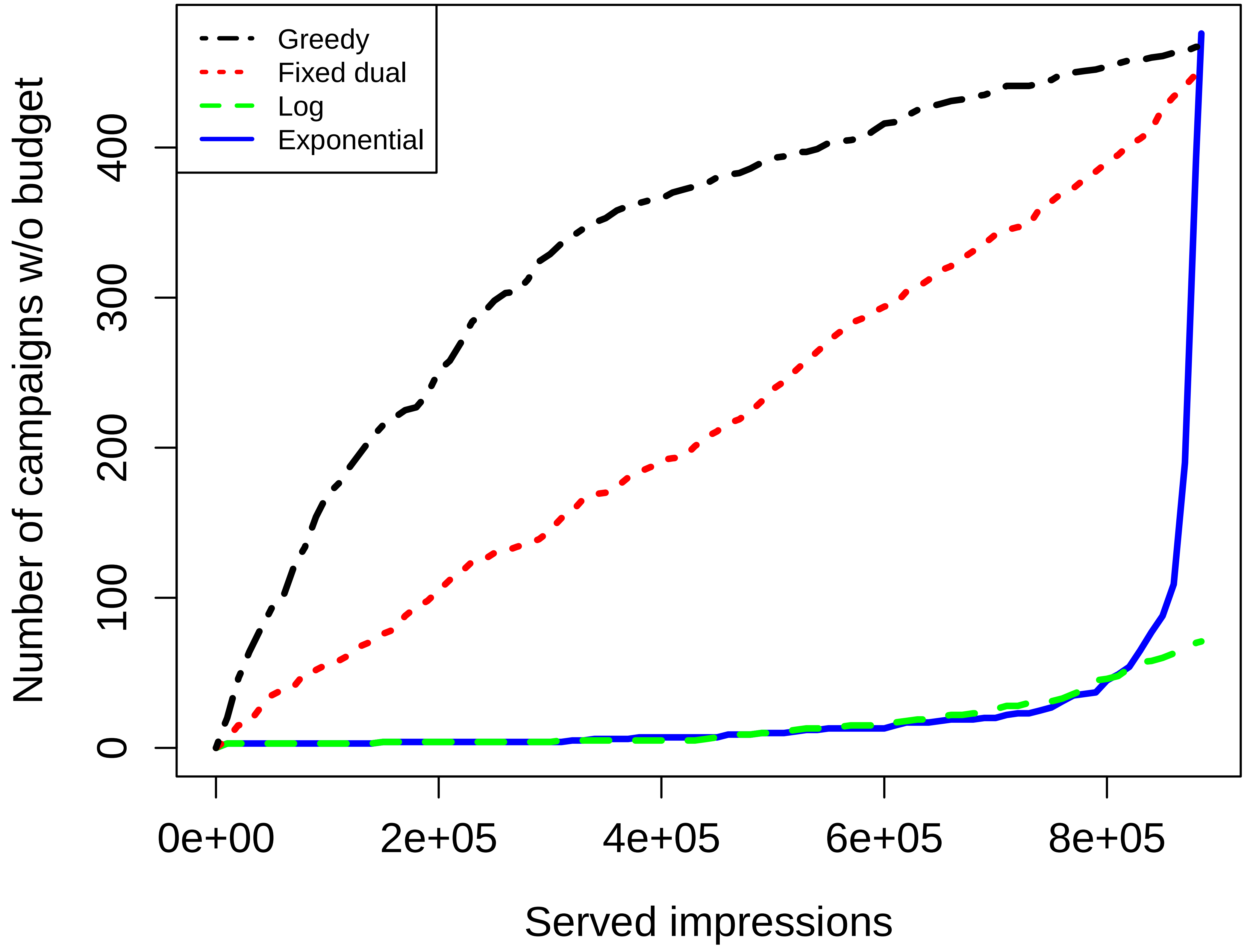
Revenues generated by different methods

- 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.



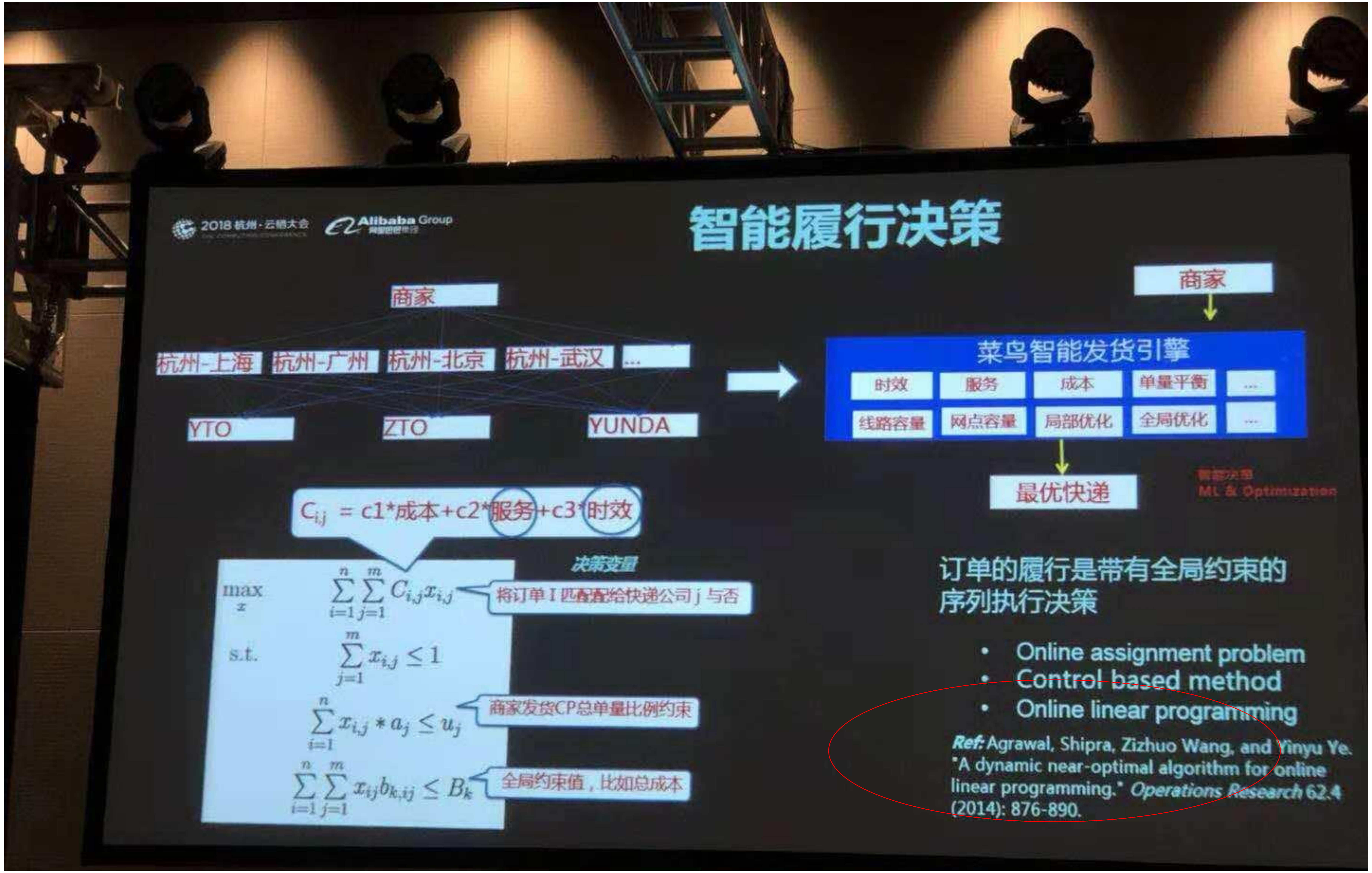
Detailed Performances

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



$$\max_x \sum_{i=1}^{n}\sum_{j=1}^{m} C_{i,j}x_{i,j}$$
$$\text{s.t.} \quad \sum_{j=1}^{m} x_{i,j} \le 1$$
$$\sum_{i=1}^{n} x_{i,j} * a_j \le u_j$$
$$\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} b_{k,ij} \le B_k$$

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

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

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:

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

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

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

- 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 $O(\log T)$ 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

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

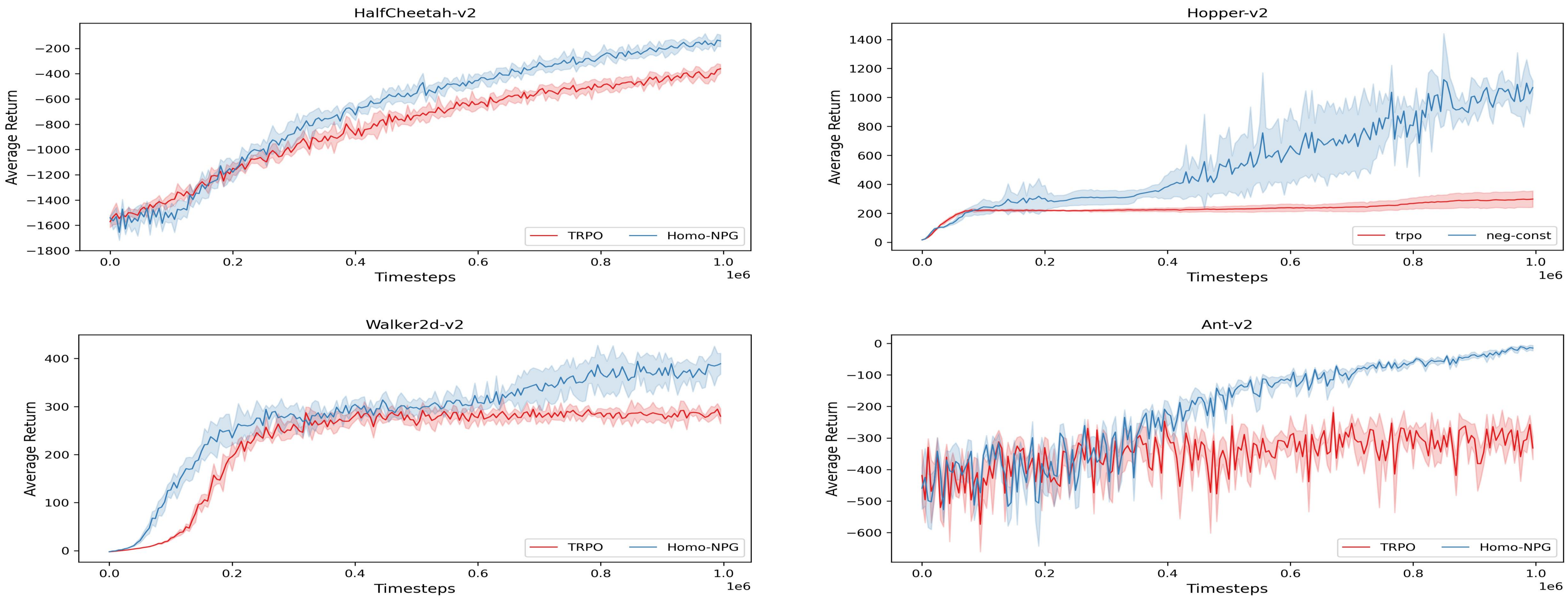
- Based on KL divergence, TRPO (Schulman et al. 2015) uses KL divergence in the constraint:

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

➡ **Homogeneous NPG: Apply HSODM!**

Preliminary Results: HSODM for Policy Optimization in RL

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



- HSODM provides significant improvements over TRPO
- Ongoing: second-order information of L?
- **Further reduce the computation cost per step**

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:

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

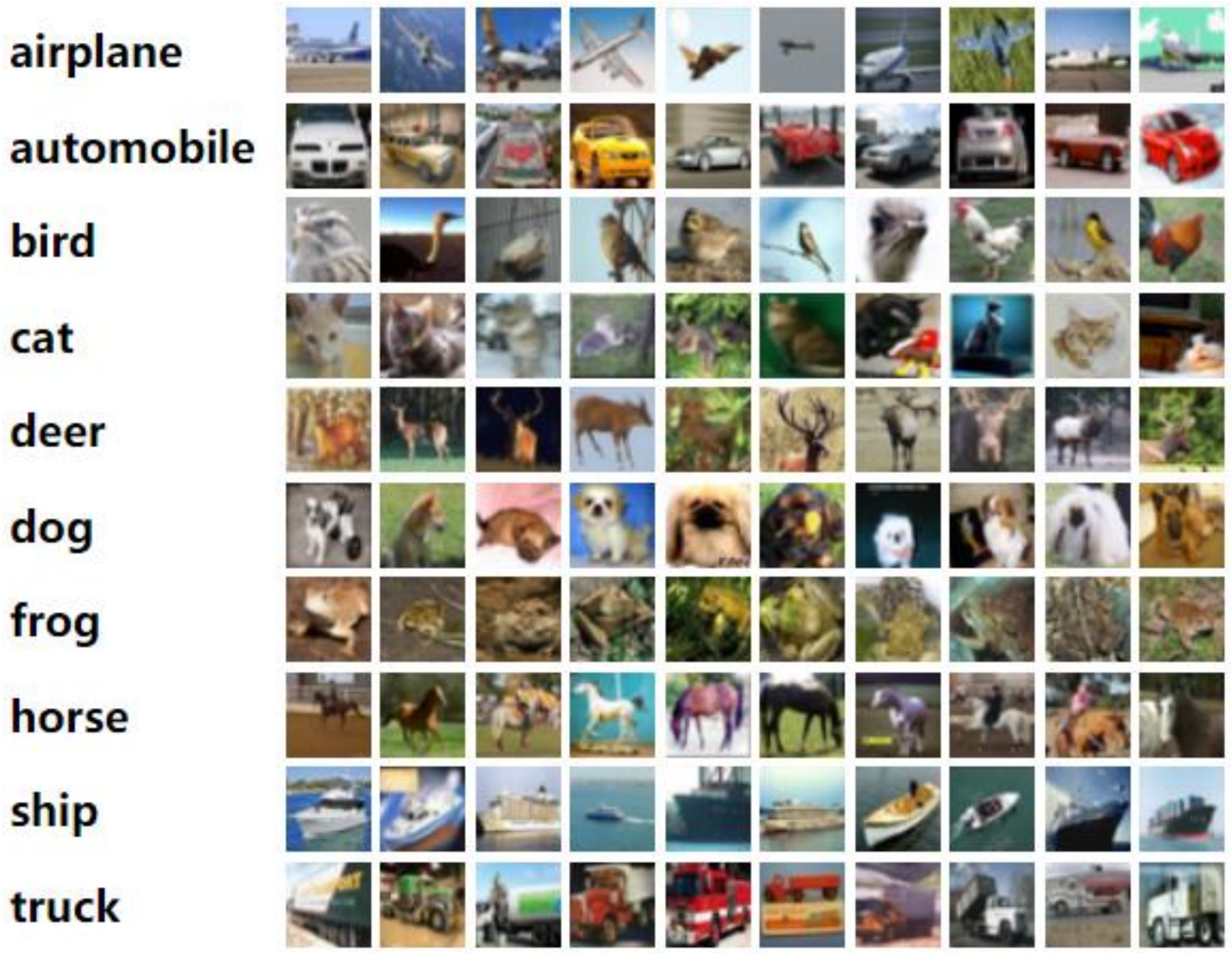
$$||\alpha||_{G_k} \leq \Delta_k$$

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

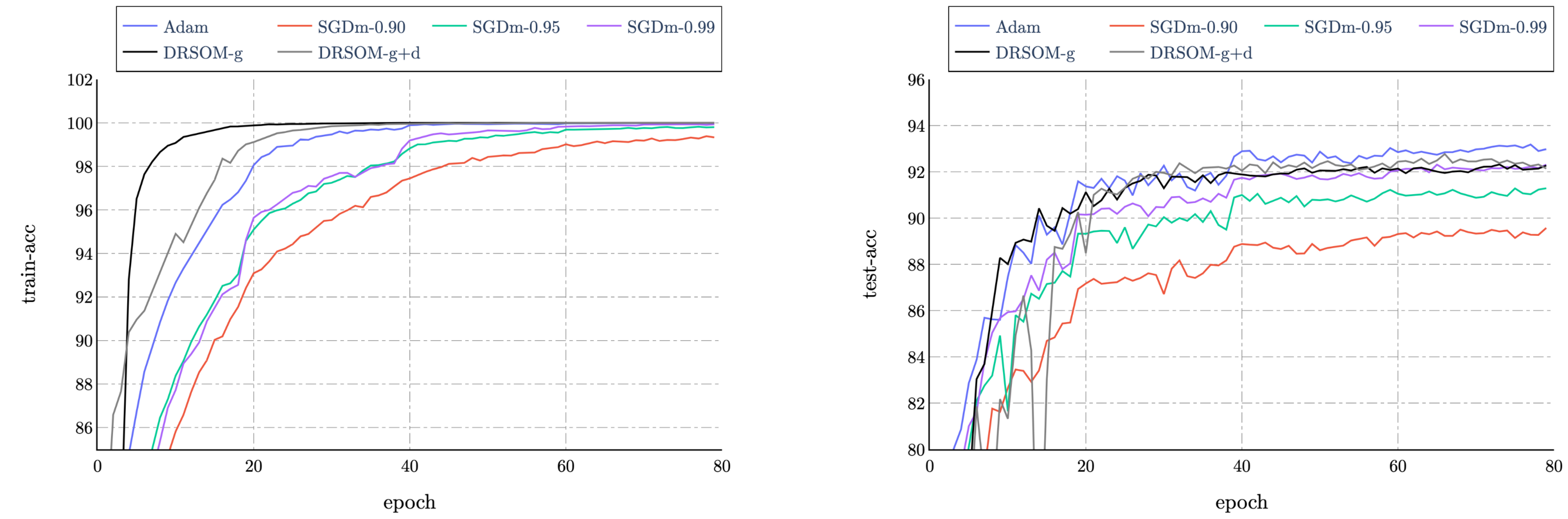
App IV: Neural Networks and Deep Learning

To use DRSOM in machine learning problems

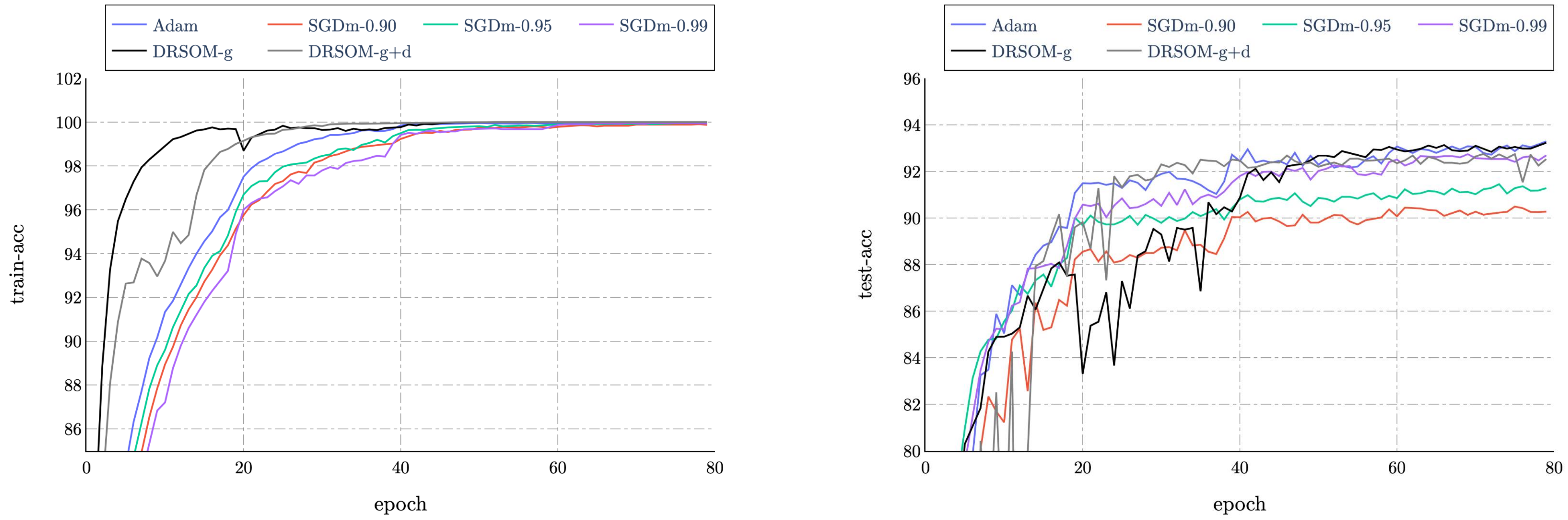
- We apply the mini-batch strategy to a vanilla DRSOM
- Use Automatic Differentiation to compute gradients
- Train ResNet18/Resnet34 Model with CIFAR 10
- Set Adam with initial learning rate 1e-3



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

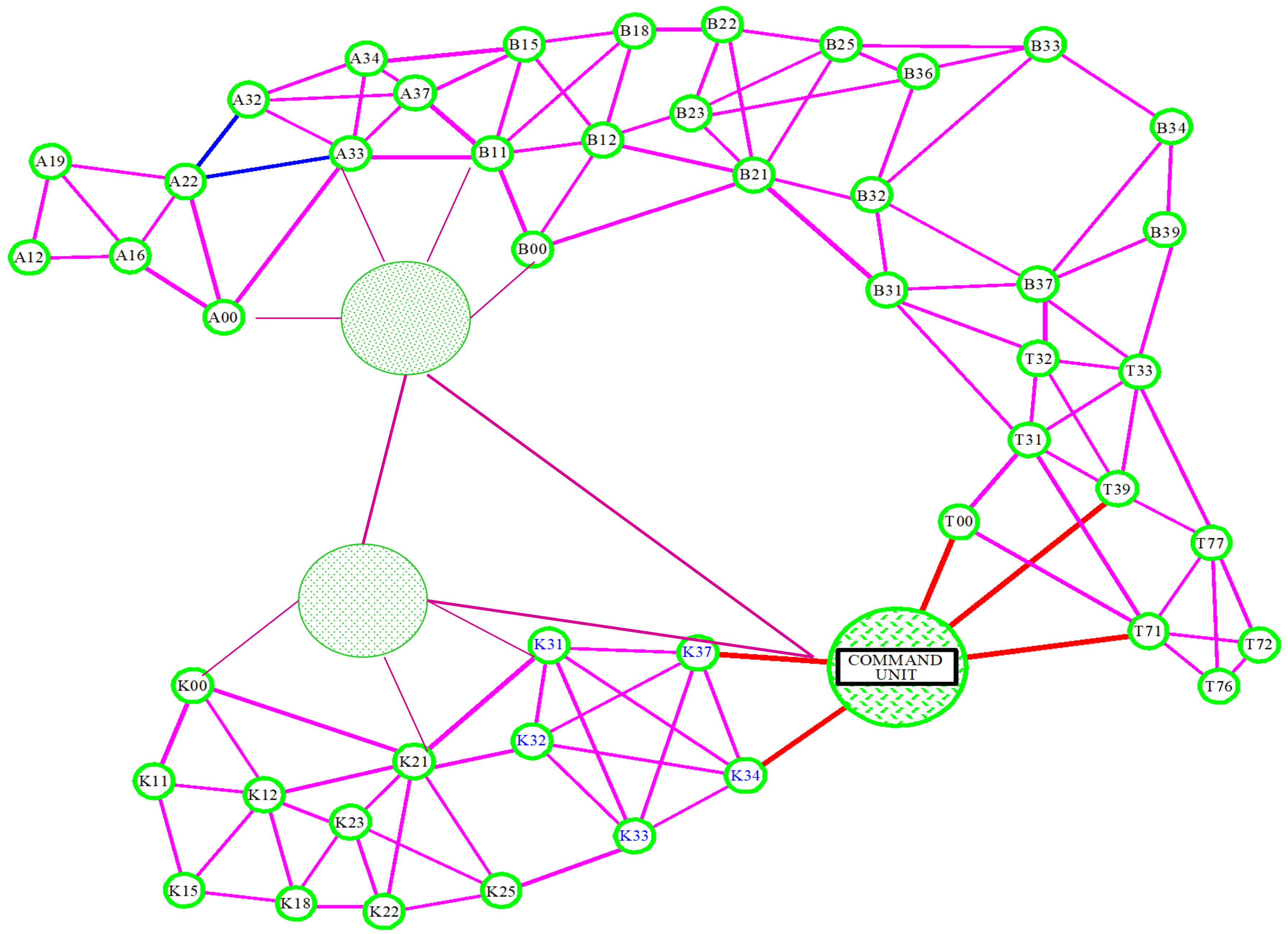
- DRSOM may over-fit the models
- Running time can benefit from Interpolation
- Single direction DRSOM is also good

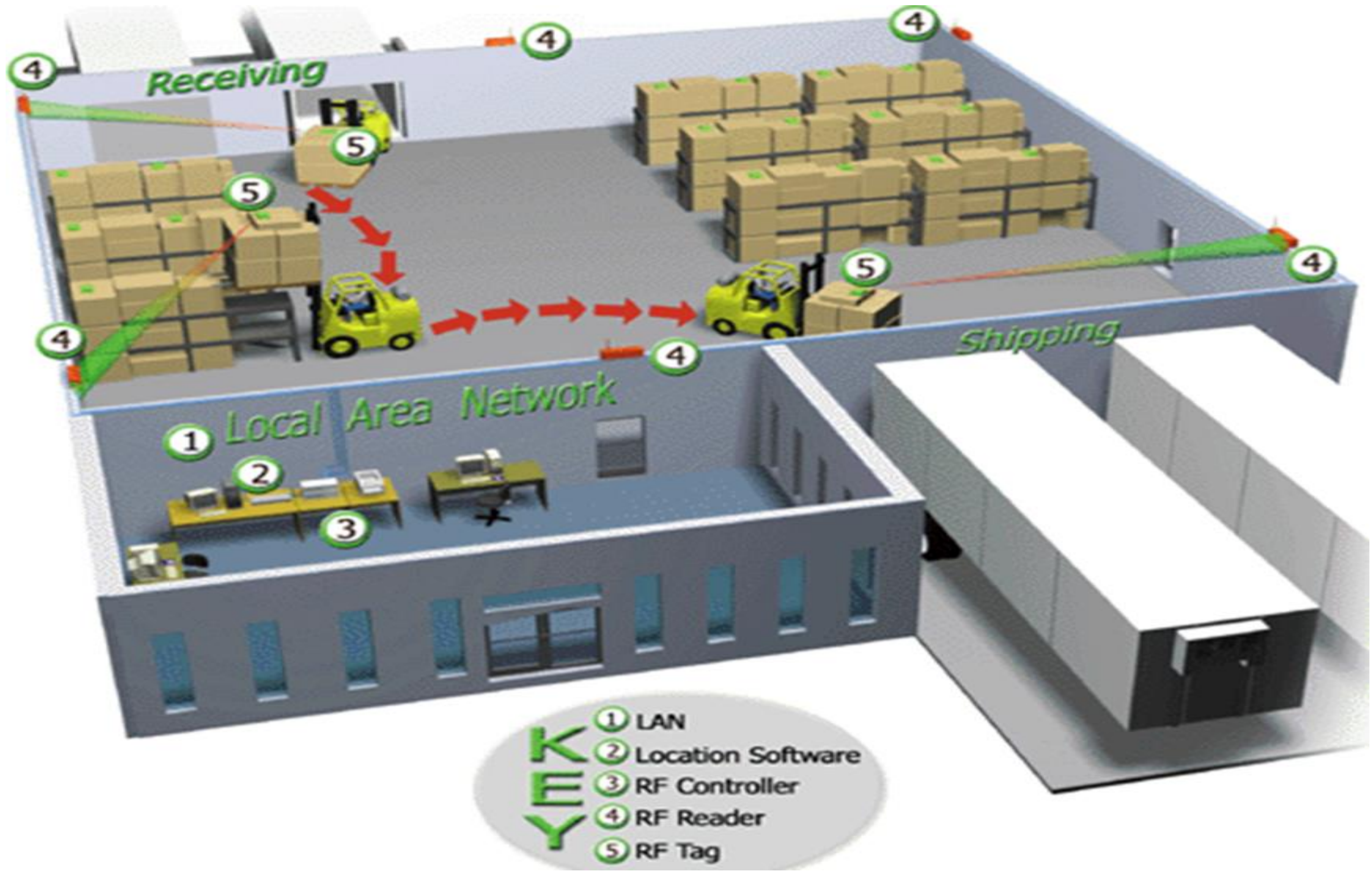
Good potential to be a standard optimizer for deep learning!

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

Step 1: **Linearization**

$$\|x_i - x_j\|^2 = x_i^T x_i - 2x_i^T x_j + x_j^T x_j$$

$\mathbf{Y_{ii}}$ $\mathbf{Y_{ij}}$ $\mathbf{Y_{jj}}$

$$\|a_k - x_j\|^2 = a_k^T a_k - 2a_k^T x_j + x_j^T x_j$$

$\mathbf{Y_{jj}}$

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

Step 2: **Relax**

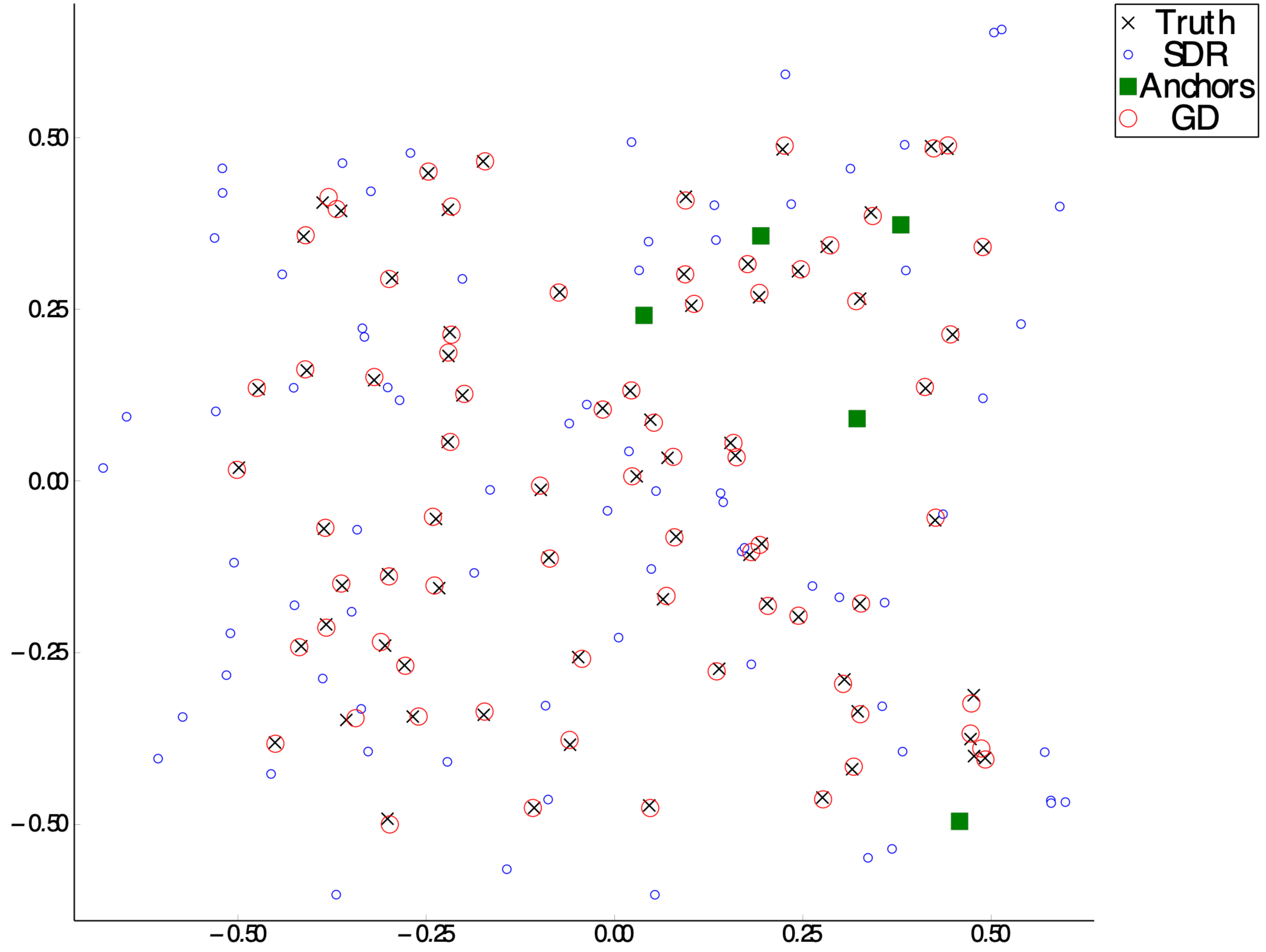
$$Y \geq X^T X \Leftrightarrow Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \geq PSD$$

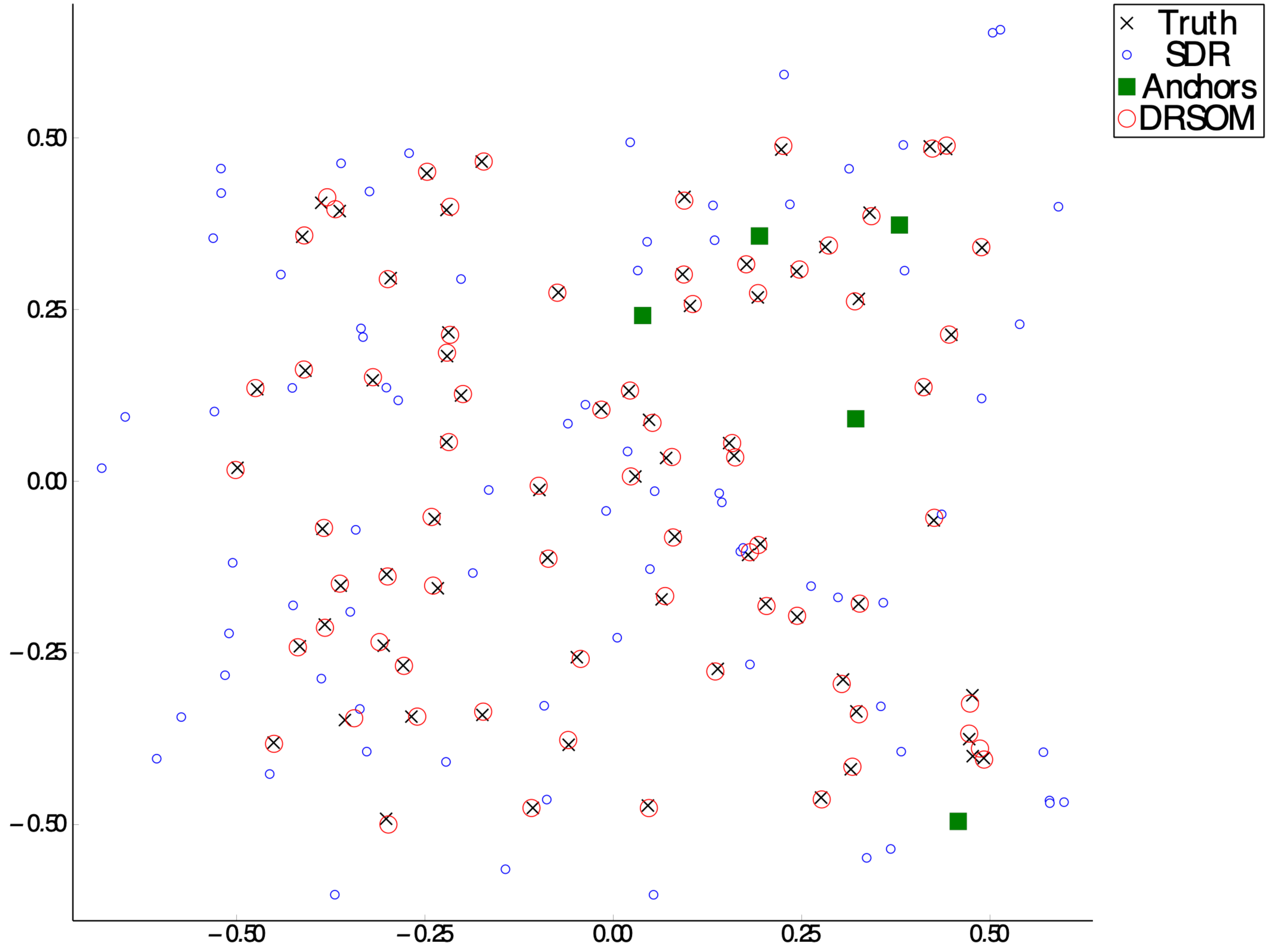
This is a conic linear program which is a convex optimization problem, but $O(n^{3.5}\log(\epsilon^{-1}))$

Biswas and Y 2004, So and Y 2005

Sensor Network Location (SNL) I

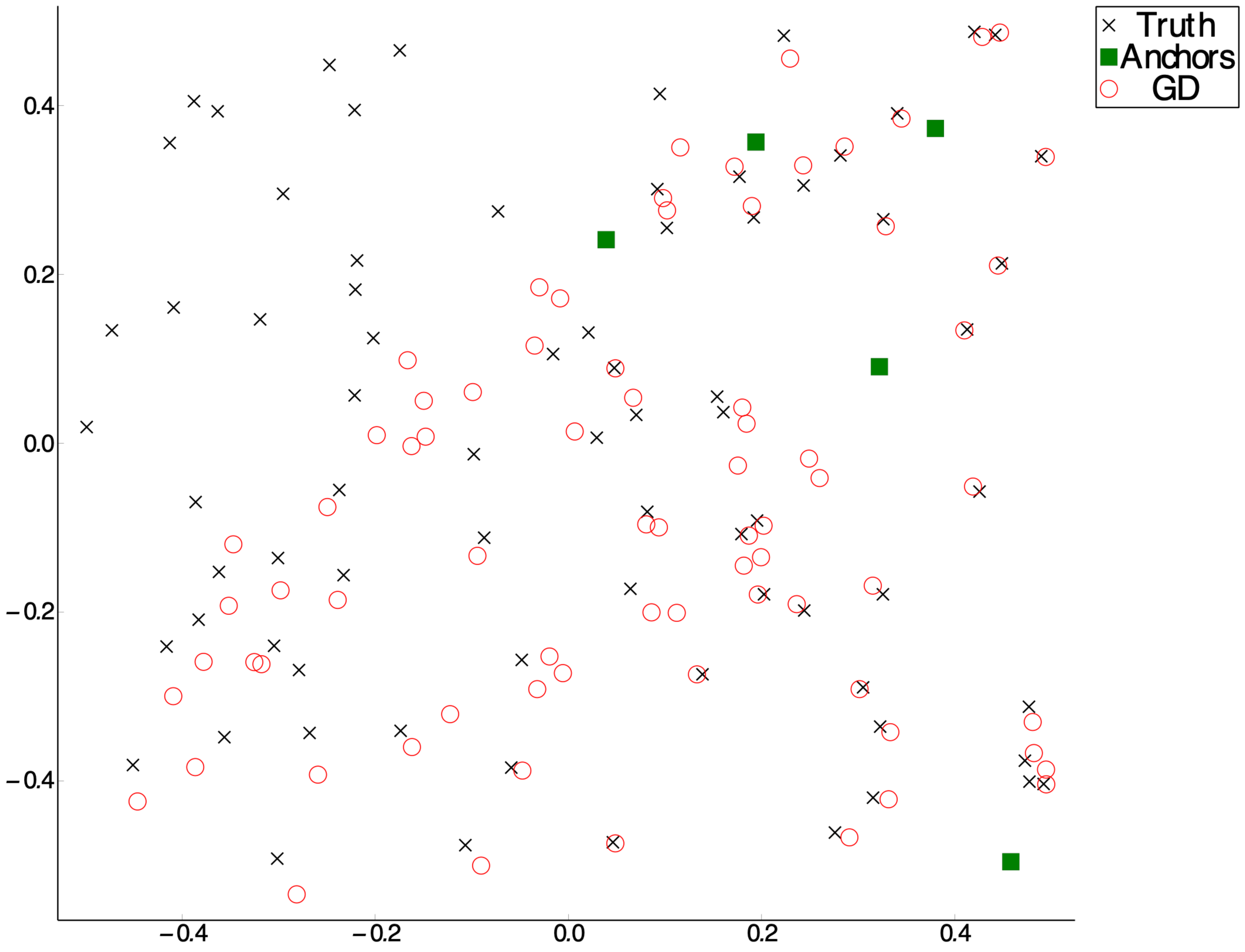
- Graphical results using SDP relaxation (Biswas&Y 2004, SO&Y 2007) to initialize the NLS
- n = 80, m = 5 (anchors), radio range = 0.5, degree = 25, noise factor = 0.05
- Both Gradient Descent and DRSOM can find good solutions !

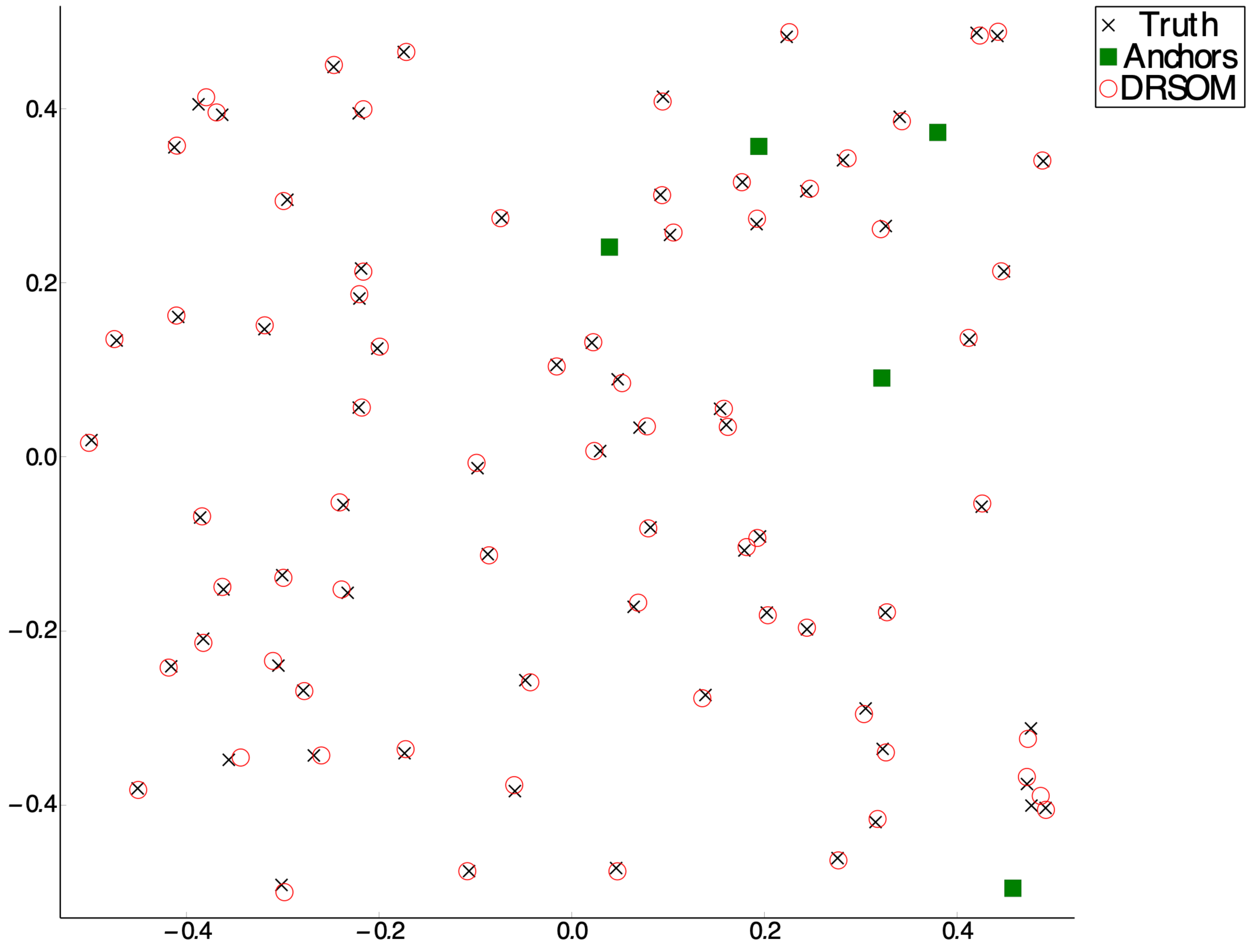




Sensor Network Location (SNL) II

- 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)

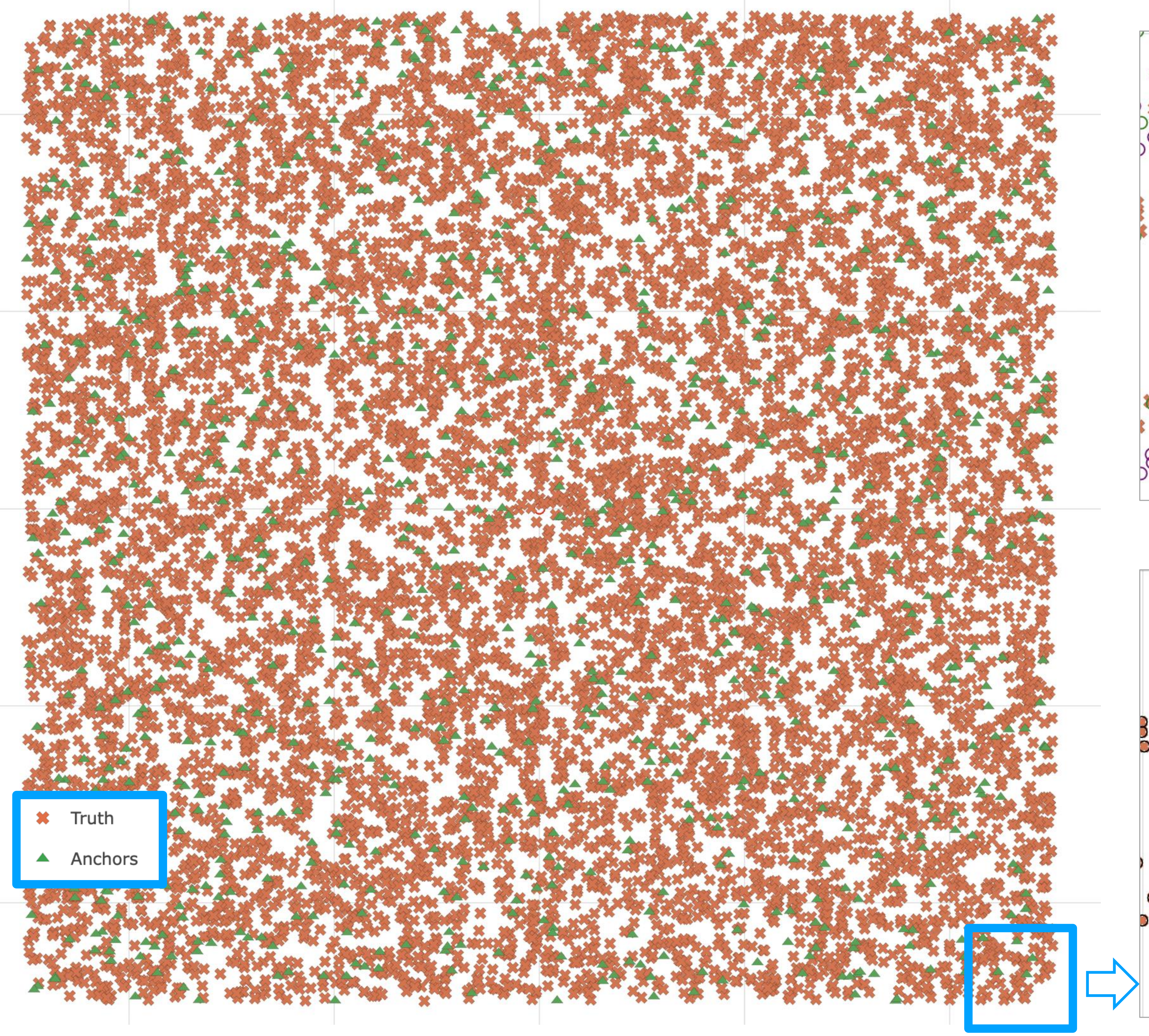
n	m	$\|E\|$	t CG	t DRSOM	t GD
500	50	2.2e+04	1.7e+01	1.1e+01	2.3e+01
1000	80	4.6e+04	7.3e+01	3.9e+01	1.8e+02
2000	120	9.4e+04	2.5e+02	1.4e+02	1.1e+03
3000	150	1.4e+05	6.5e+02	1.4e+02	-
4000	400	1.8e+05	1.3e+03	5.0e+02	-
6000	600	2.7e+05	2.0e+03	1.1e+03	-
10000	1000	4.5e+05	-	2.2e+03	-

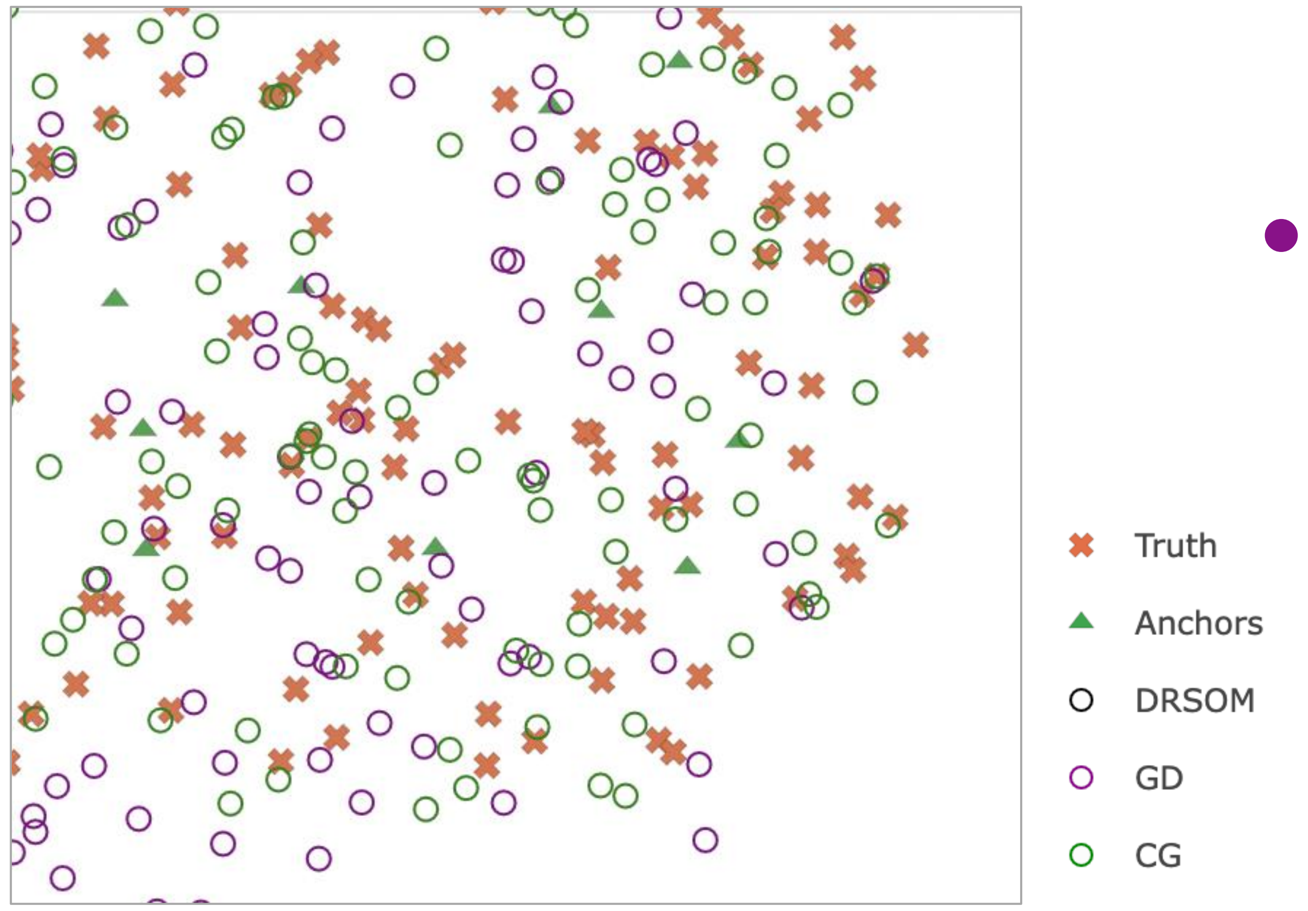
Table 2: Running time of CG, DRSOM, and GD on SNL instances of different problem size, $|E|$ denotes the number of QCQP constraints. "-" means the algorithm exceeds 3,000s.

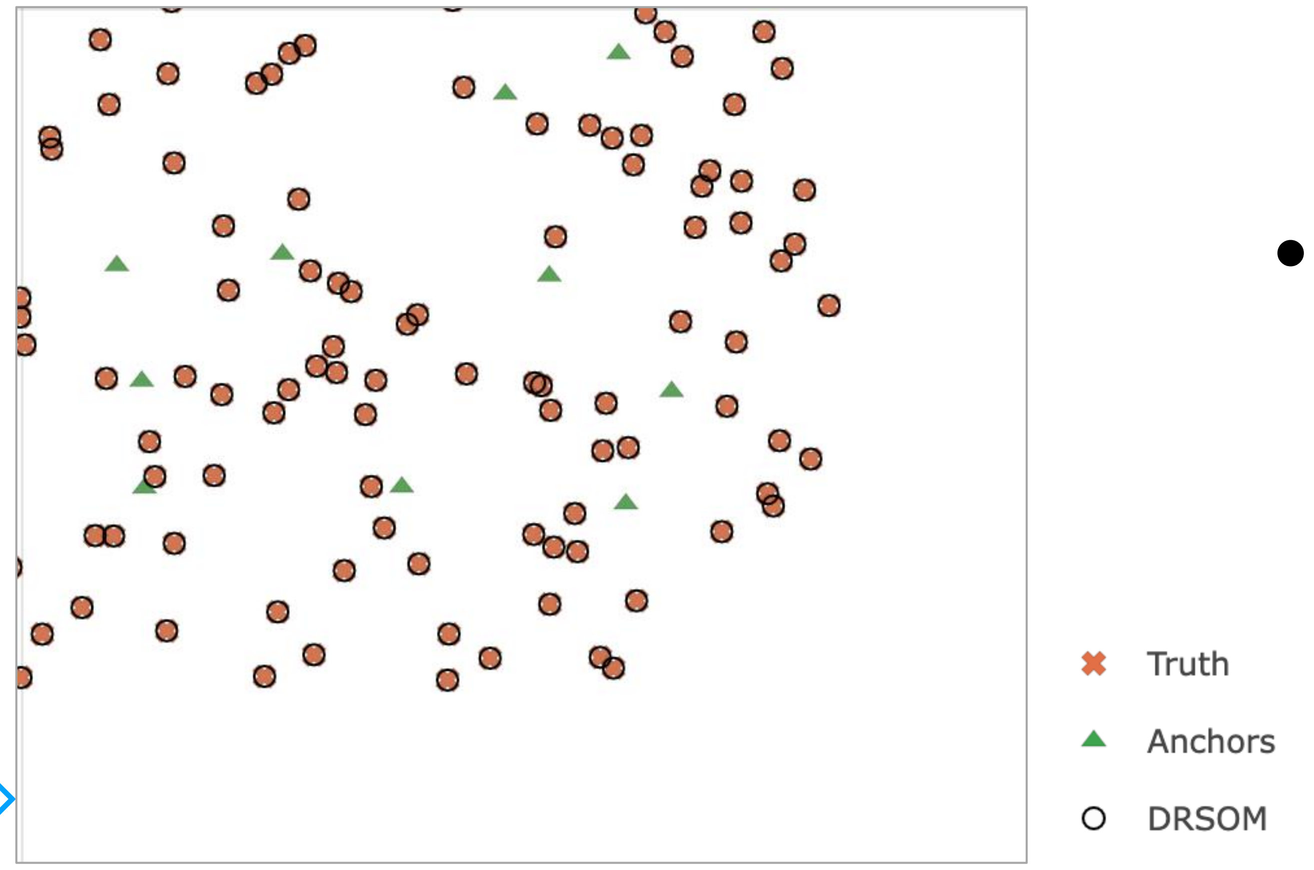
- DRSOM has the best running time (benefits of 2nd order info and interpolation!)

Sensor Network Location, Large-Scale Instances 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**

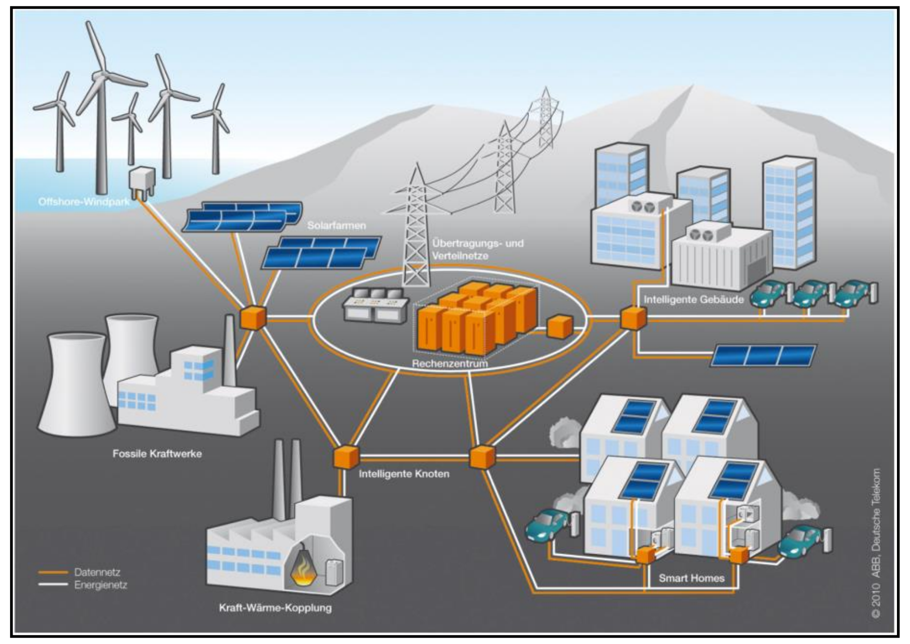
- **DRSOM can converge to $|g_k| \le 1e^{-5}$ in 2,200s**

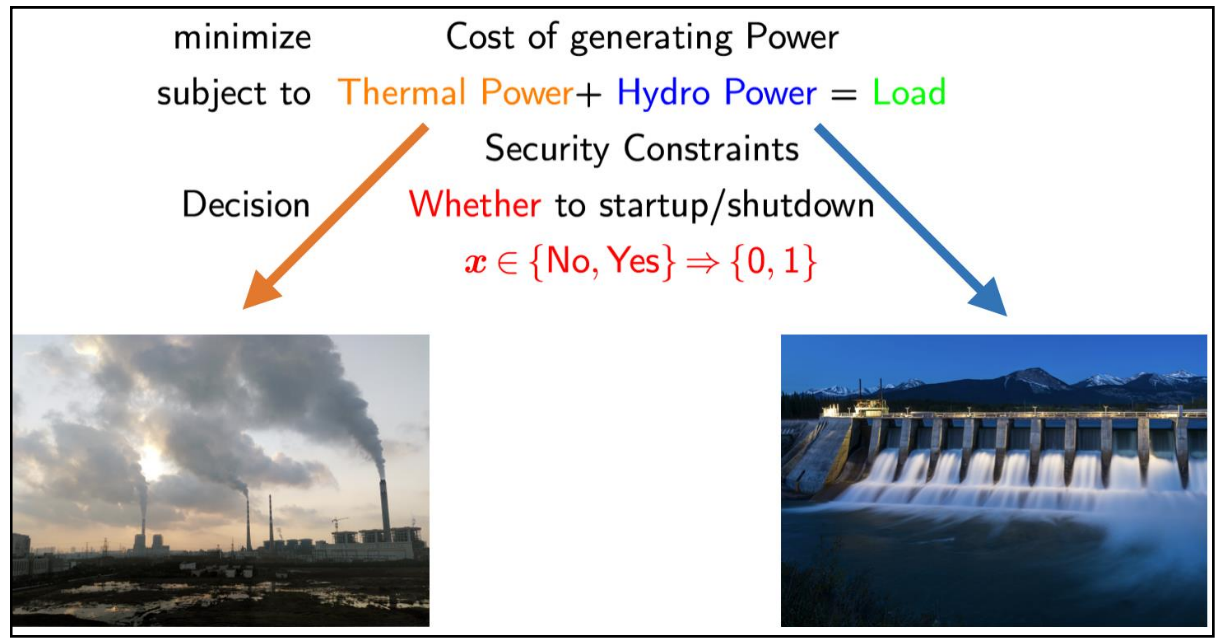
Sensor Network Online Tracking, 2D and 3D

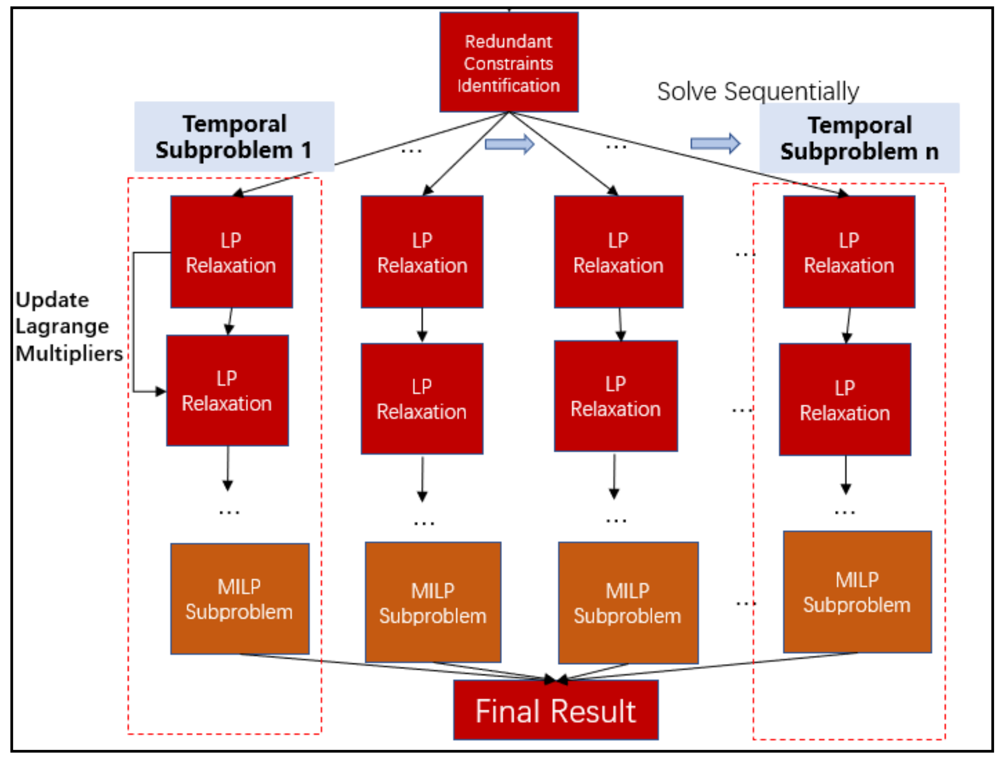
Topic 3: Mixed Integer Linear Programming Solver

Application VI: Unit Commitment and Power Grid Optimization

COPT, Cardinal Operations 2022







Unit Commitment Problem

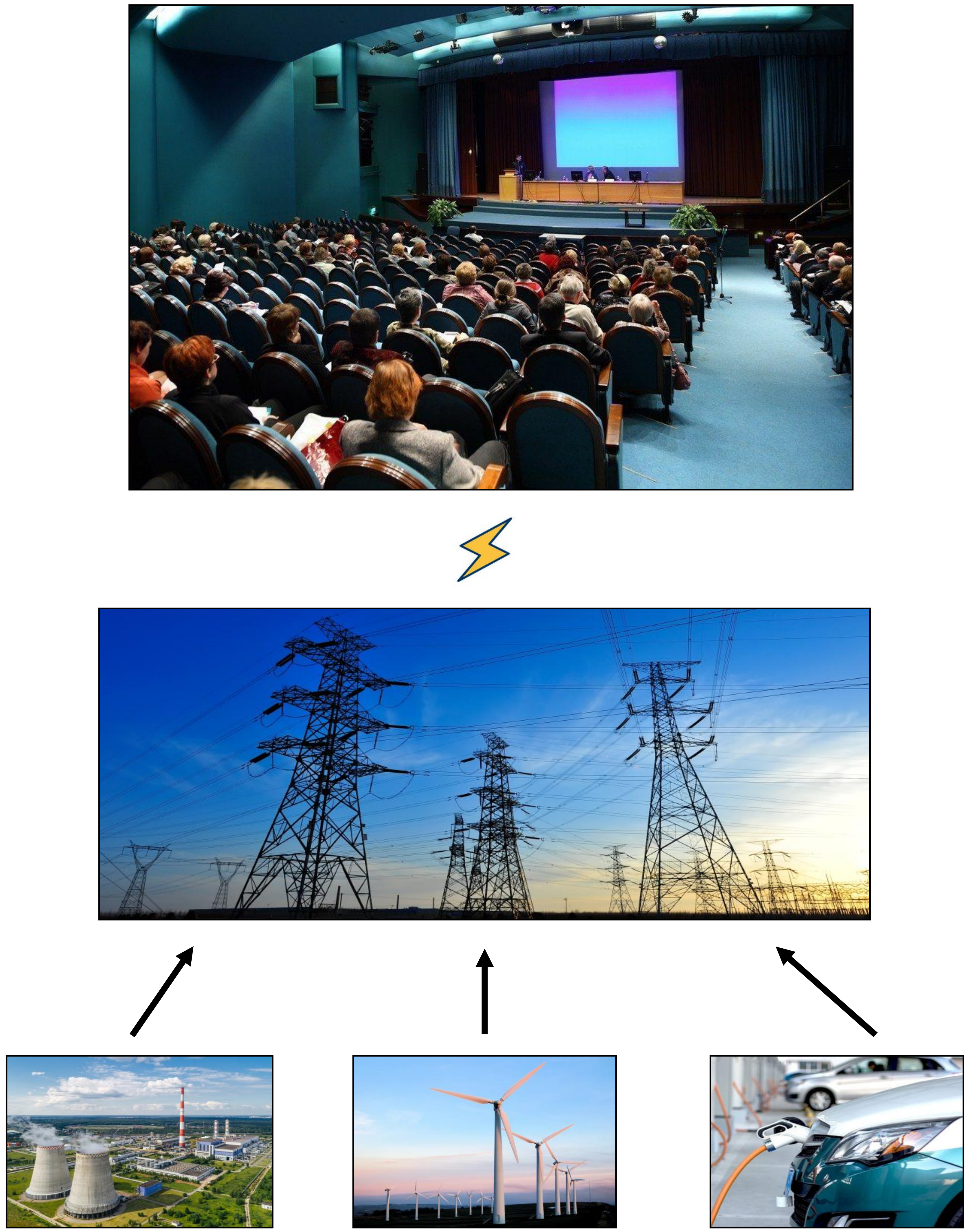
- Electricity is generated from units (various generators)
- Transmitted *safely* and *stably* through power grids
- Consumed at minimum (reasonable) price

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

Case Study: Sichuan Thermal-Hydro Hybrid Model

- **A UC problem from real-life background (Sichuan Province)**
- **With 20 thermal and 230 hydro units**
- **Hydro units involve no decision (binary variables)**

minimize Cost of electricity

subject to Thermal Power + Hydro Power = Load

Other Constraints

Decision Unit Operation Decision

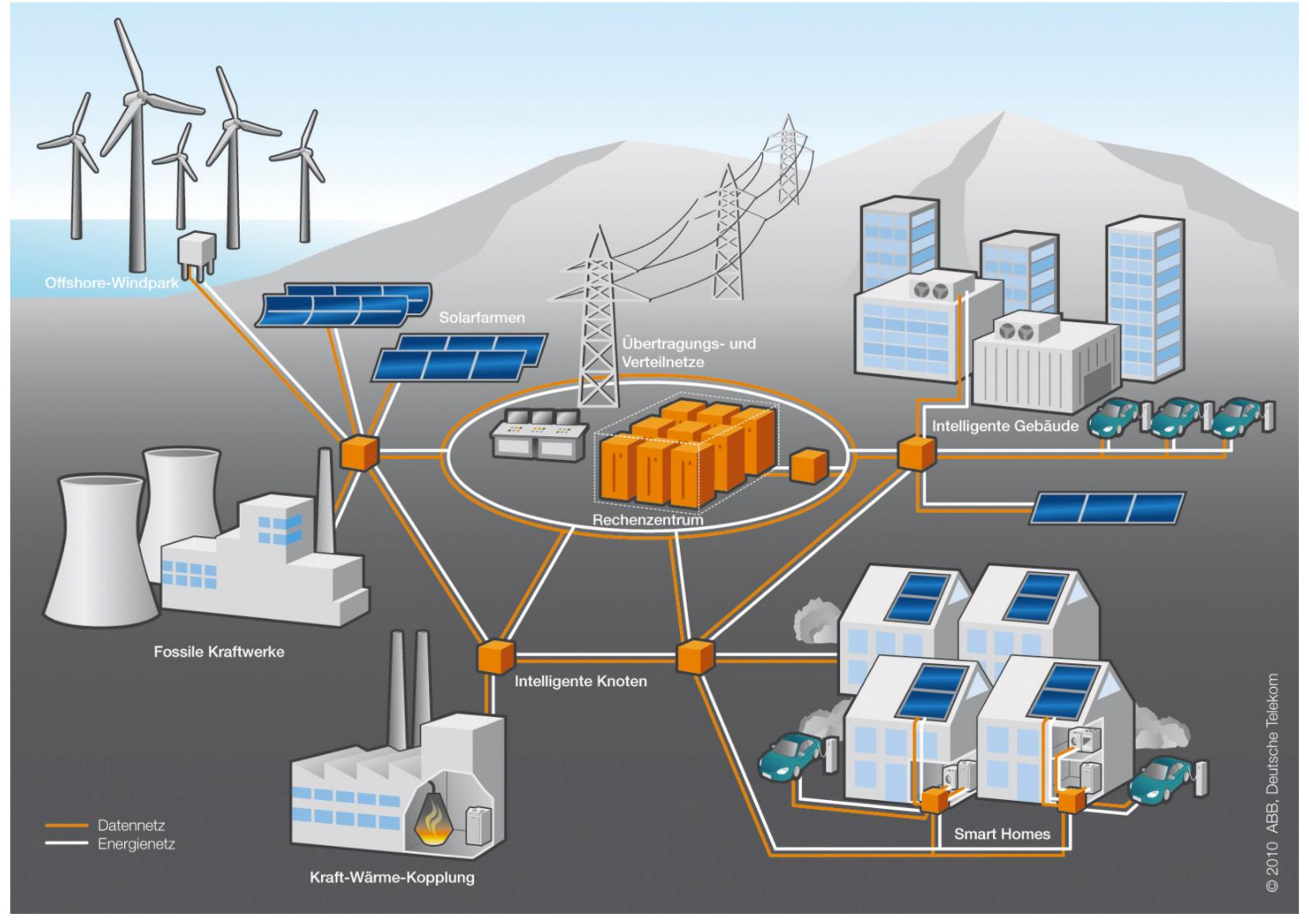
Hardness

- Costs are piecewise in generated power
- All the units are coupled by the Load balancing constraint
- A much larger and harder MILP model, *but*

Better Modeling + Algorithm Makes it Easier!

Successively Implemented in a Much Larger Region

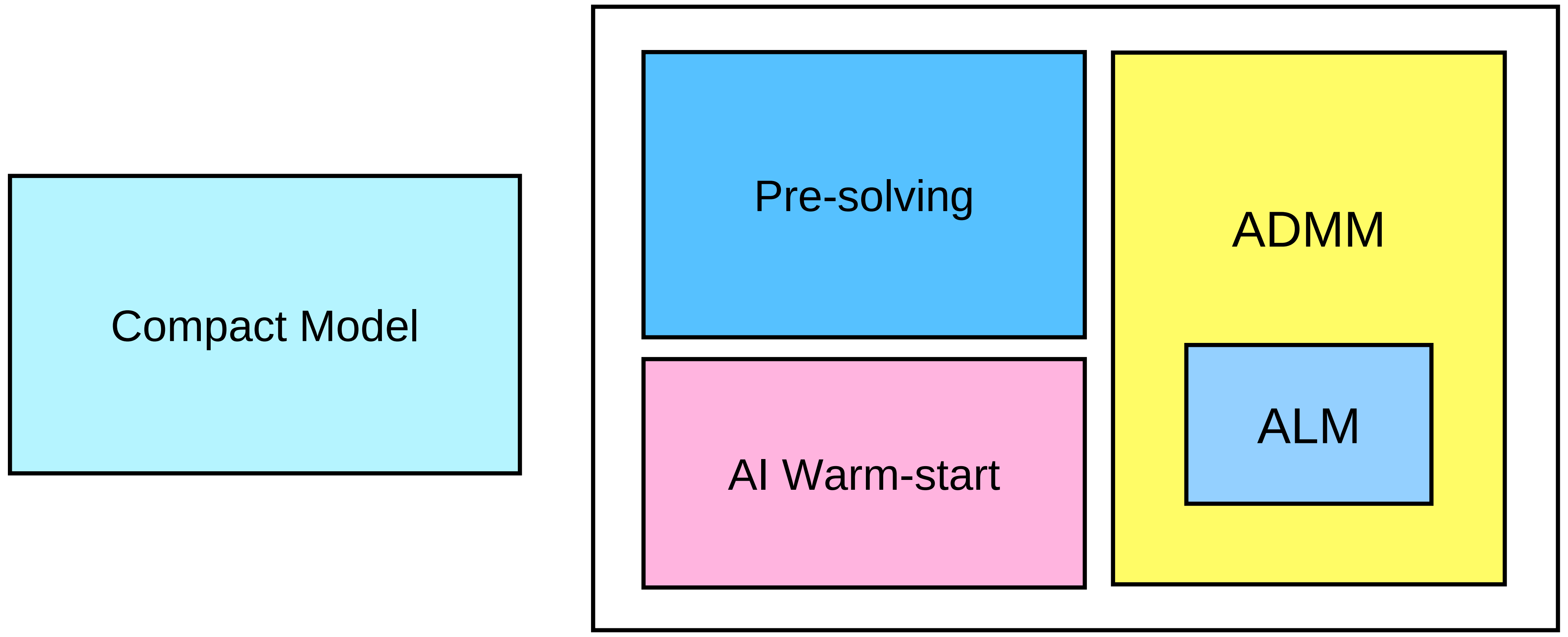
- A much larger UC problem with security constraint
- With much more (millions of) constraints and variables
- More than 1000 units of Thermal, Hydro and New energy
- Consider interaction between regions and time periods



Huge size + Various business logic + Complicated coupling constraints

- Intractable without exploring structure
- Accurate and succinct model helps
- Domain specific algorithms matter a lot
- ML/AI has a big role to play

Model, Algorithm and ML/AI together make it tractable



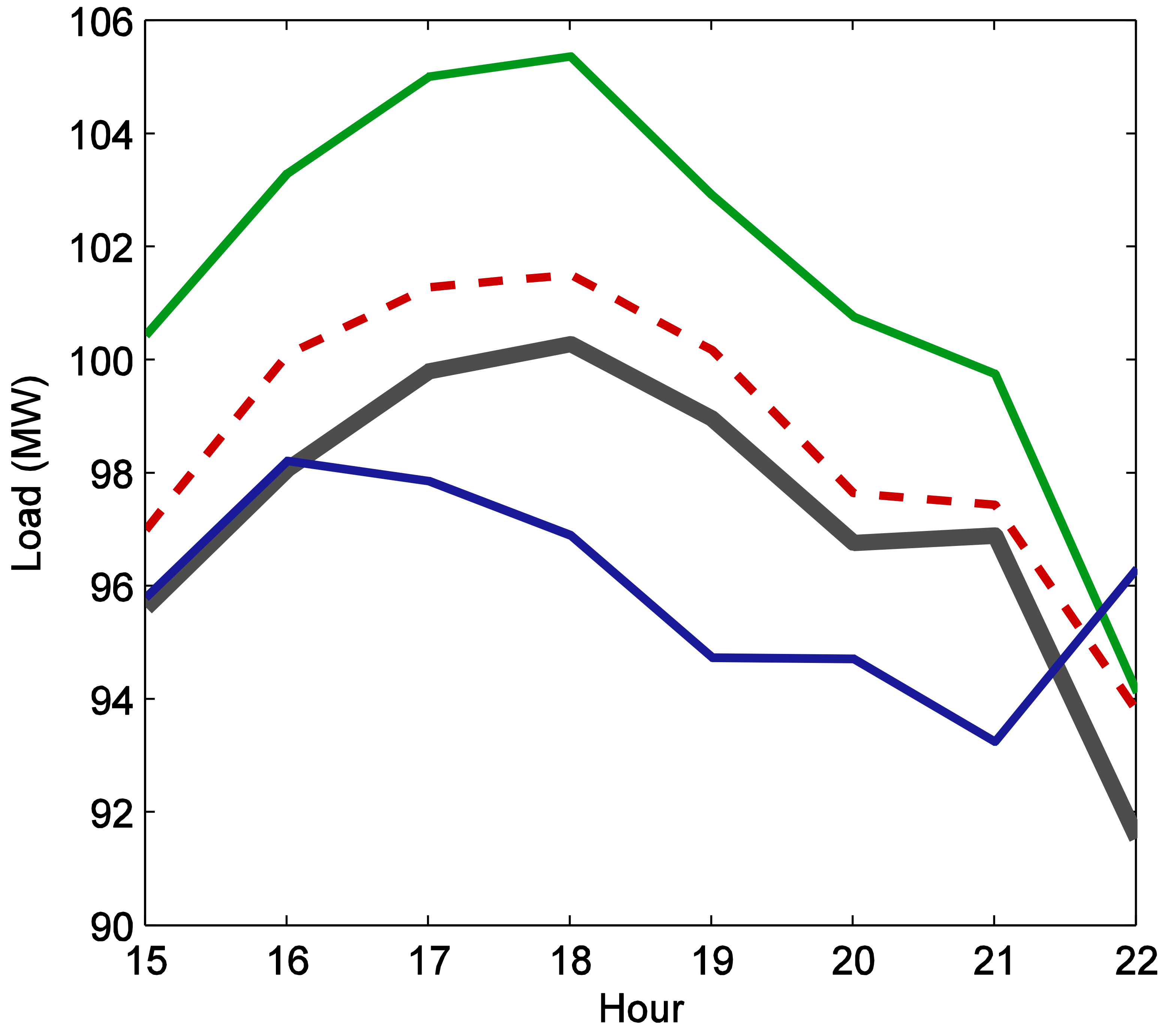
App. VII: Beijing Public Transport Intelligent Urban Bus Operations Management with Mixed Fleet Types and Charging Schedule

北京公交 Beijing Public Transport

杉数科技 Cardinal Operations

Kickoff 2022.8

Peak Reduction due to Smart Charging and Discharging

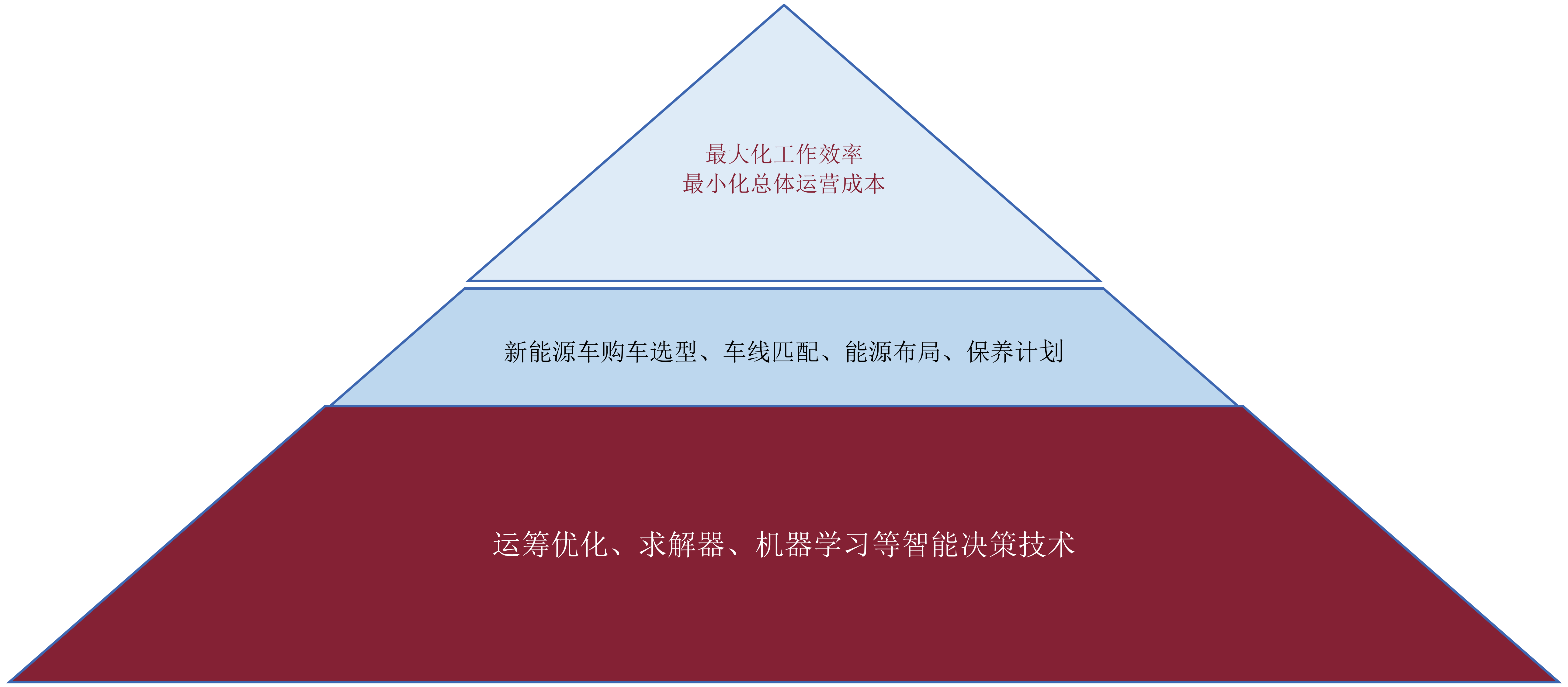


	Standard	Low PGE	Linear Progr.
Total Fleet ($)	97,678	83,695	65,349
Mean Cost / Mile	0.068	0.044	0.0054
Increase in Peak	5.1%	1.4%	-0.25%

Background: Decision Intelligence in the case of Beijing Public Transport

北京公交 Beijing Public Transport

杉数科技 Cardinal Operations



北京市“十四五规划”目标

加快构建“综合、绿色、智能、安全”的立体化现代化城市交通系统

- 到2025年，中心城区绿色出行比例提高至 76.5%
- 全面推进智慧城市建设，重点发展智慧交通
- 围绕轨道交通优化地面公交线网，减少长距离、长时间运行线路，提高车辆利用率

加快建立科学、高效的“城市智能运行决策管理体系”

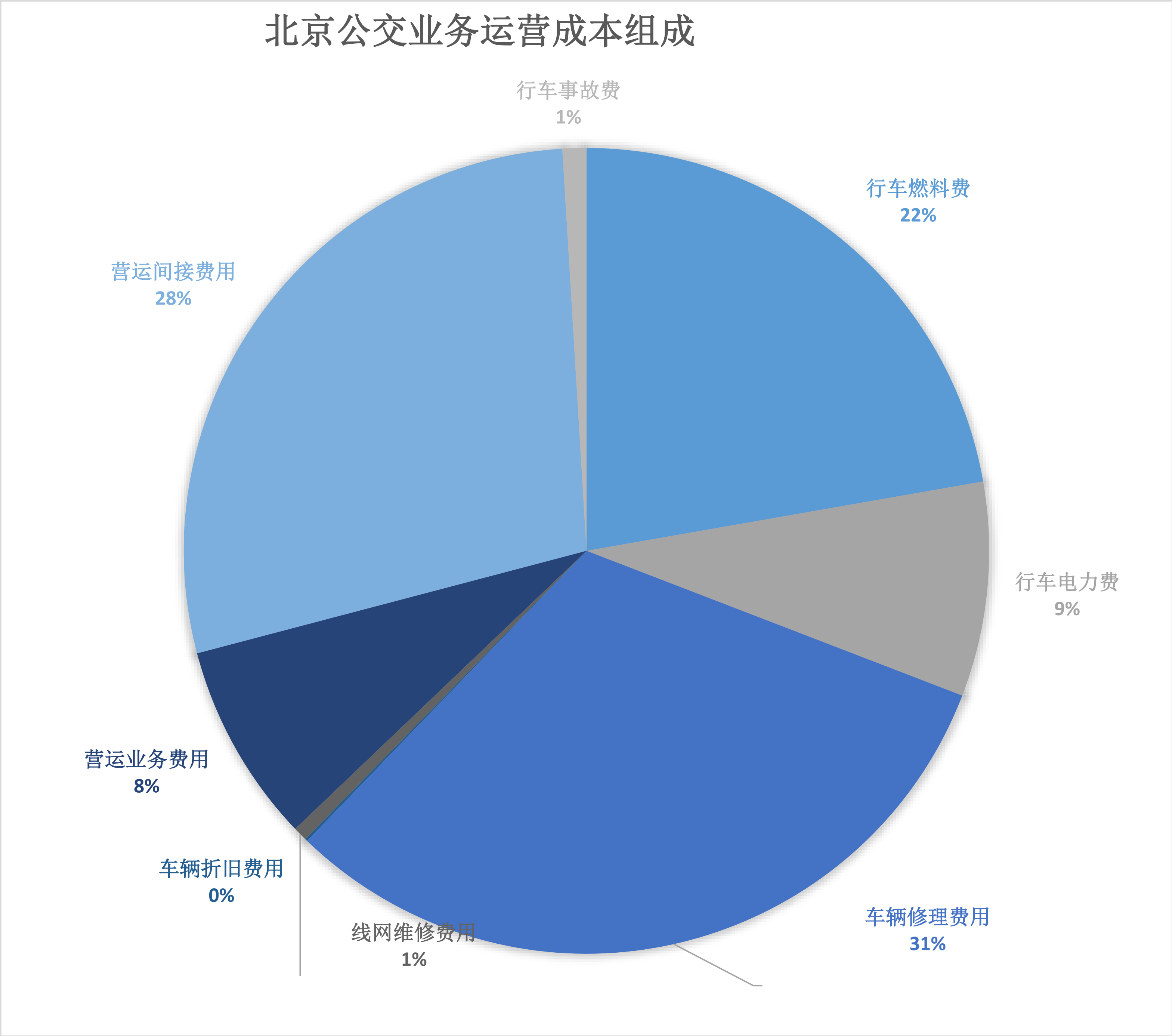
- 高水平推动城市交通的数字转型和智慧升级，形成城市交通整体解决方案
- 加快建设公共交通网络化智能调度体系，让公交出行越来越可靠，时间有保证

More efficient and intelligent decision-making to achieve 14th Five-Year Plan goals

Beijing Public Transport Line 7 is selected as the Key Pilot Unit of the intelligent transformation of Beijing Public Transport

Intelligent Transformation Empowered by Cardinal Operations

北京公交 Beijing Public Transport 杉数科技 Cardinal Operations



Beijing Public Transport's total operational costs reached 6.65 billion Yuan in 2020, of which fuels, electricity, maintenance, repair and other indirect costs accounted for over 90%. Preliminary analysis shows various potential use cases for optimization in cost reduction.

北京公交 Beijing Public Transport

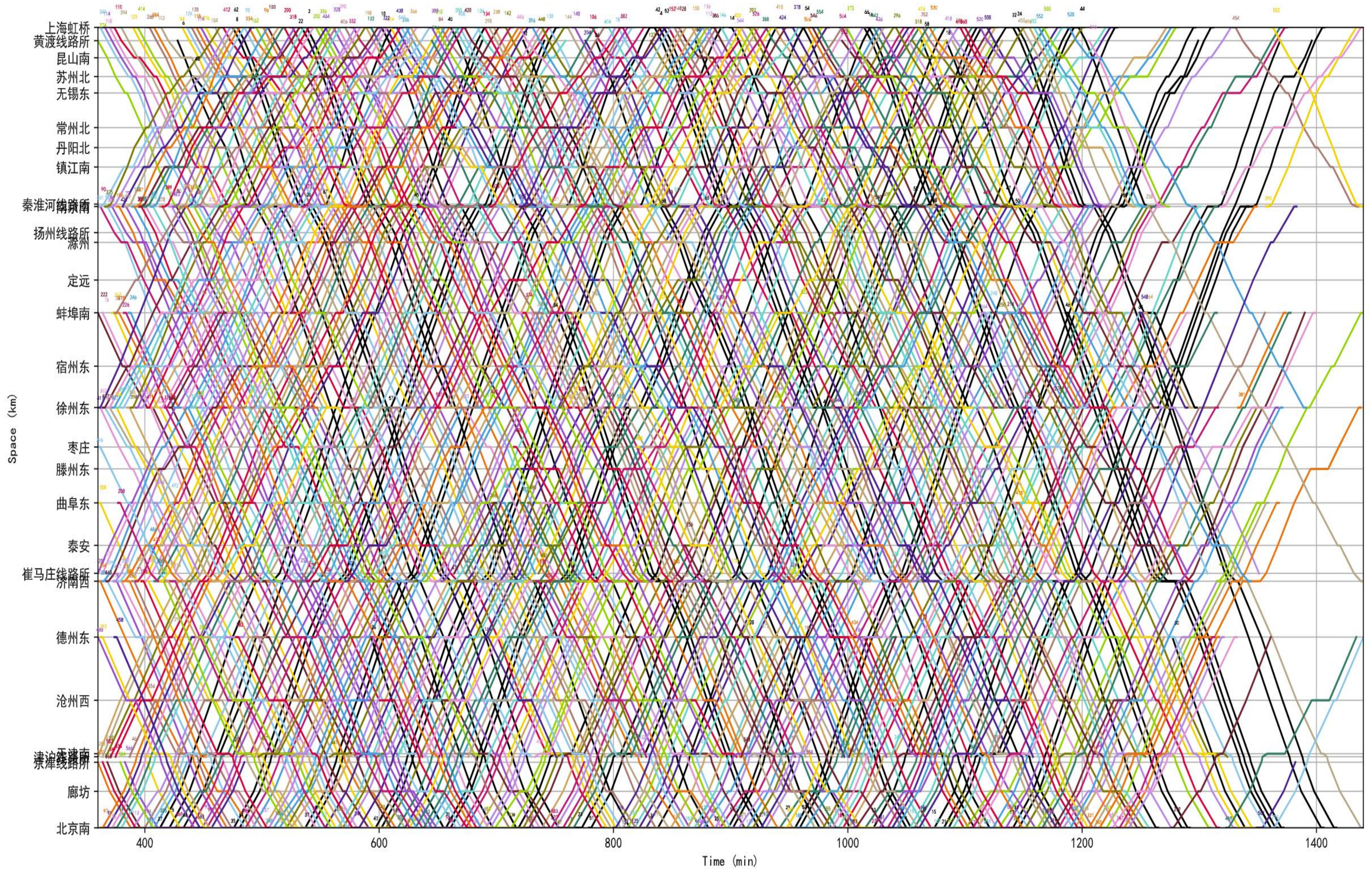
X

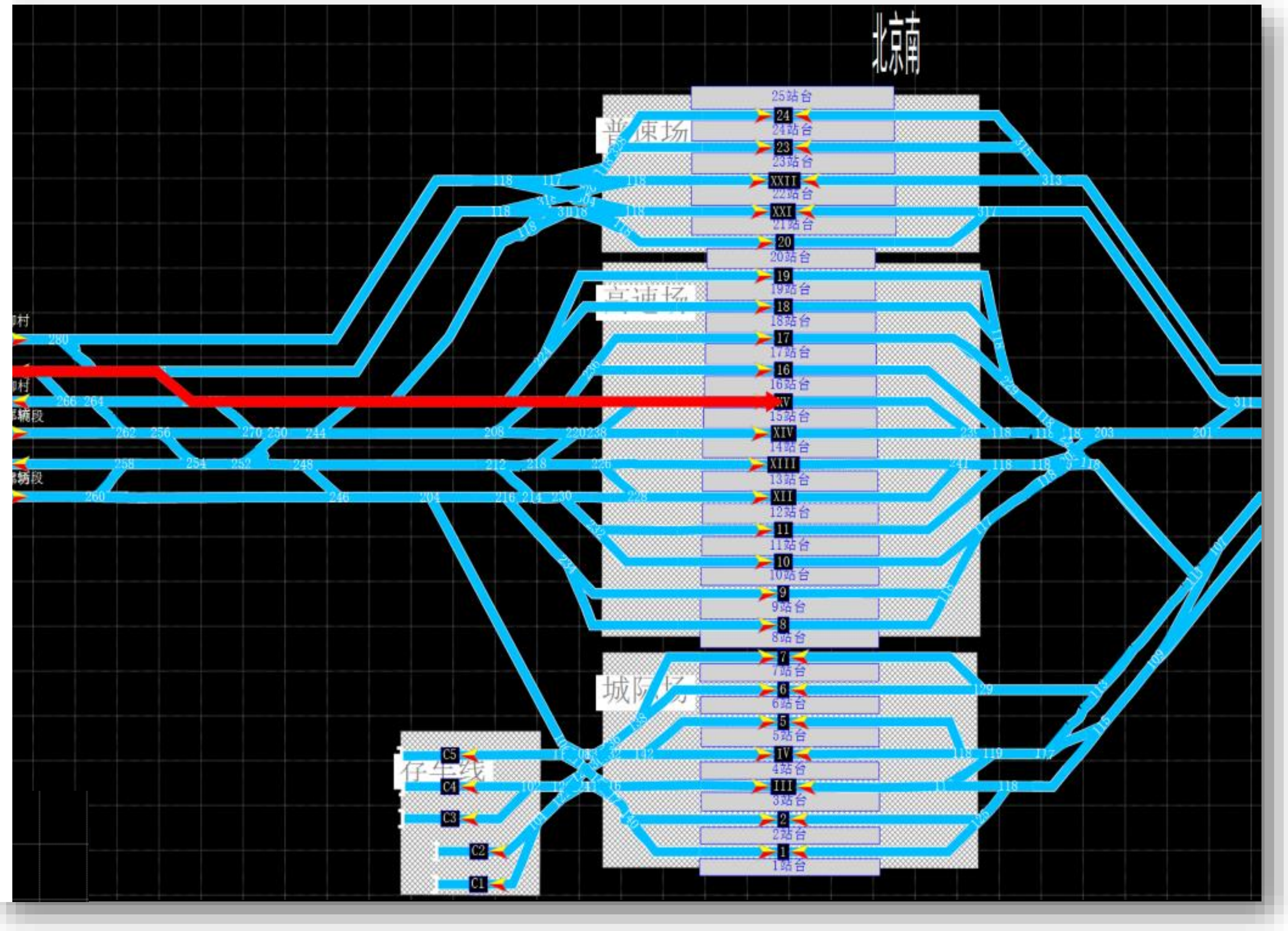
杉数科技 Cardinal Operations

Beijing Public Transport, in partner with Cardinal Operations, aims to build an innovative integrated system for smart operations in urban public transportation operations, and explore larger markets in the future.

App. VIII: Beijing-Shanghai High-speed Railway Scheduling Optimization

COPT, Cardinal Operations 2022





Background

- China High-speed Railway has been committed to providing high-quality transportation services to passengers, and the formulation of train scheduling is a key link in the operation.
- **At present, train scheduling is based on human experience**, which becomes increasingly difficult to handle the growing network. Therefore, both industry and academia are seeking ways to **automate train scheduling**.
- The train scheduling problem can be divided into **Train Timetabling Problem (TTP)** and **Train Platforming Problem (TPP)**.

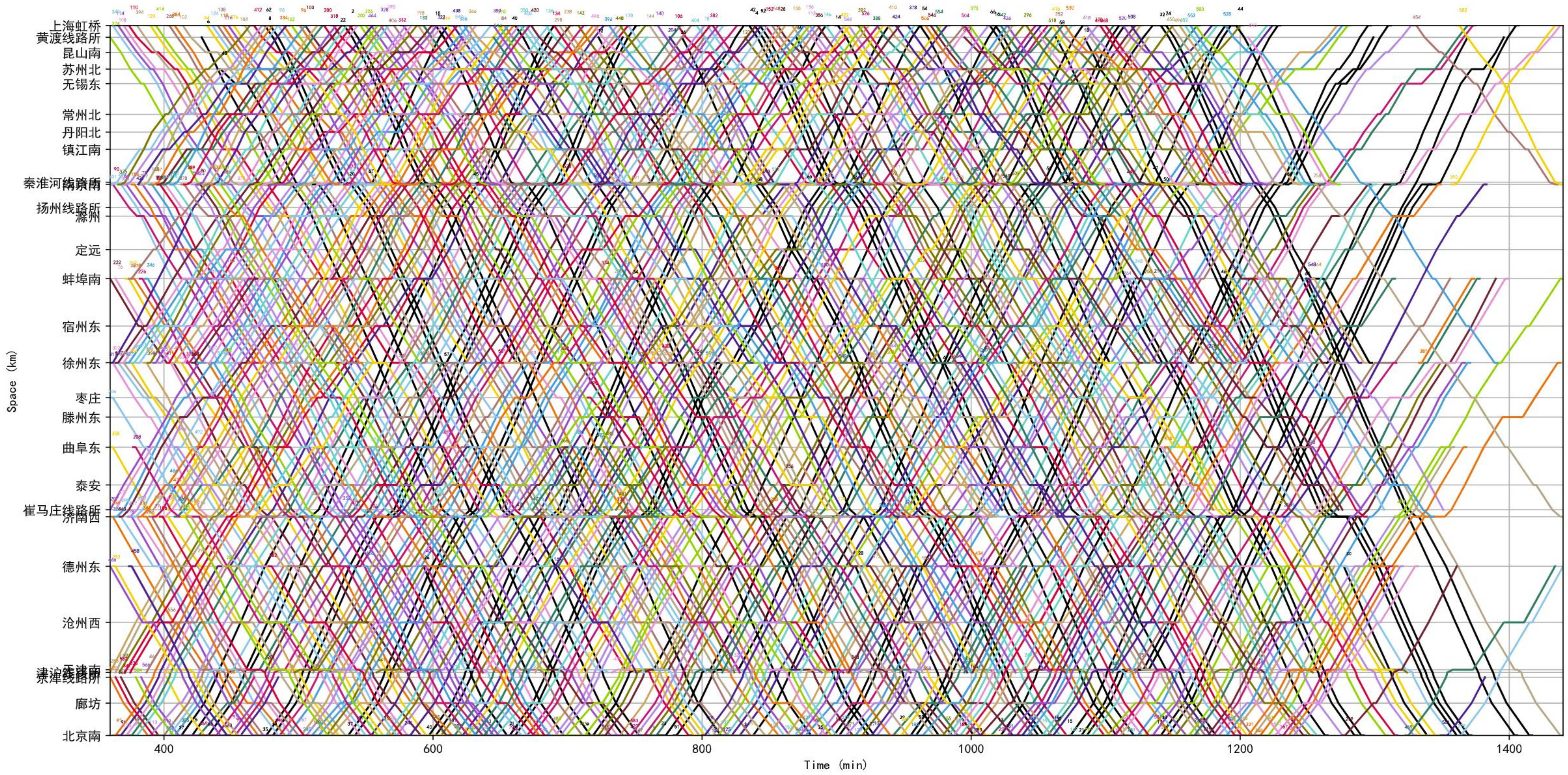
Optimization Model:

- **Objective:** maximize the number of trains placed in the train scheduling, thereby maximizing operating revenue；
- **Constraints:** describe the running behavior of trains and prevent train collisions；

- The project mainly solves **TTP for Beijing-Shanghai High-speed Railway** and **TPP at Beijingnan Railway Station**.
  - **Beijing-Shanghai High-speed Railway** is the busiest high-speed railway with the largest number of passengers in China. It is 1,318 km in total and passes 29 stations.
  - **Beijingnan Railway Station** is the largest railway station in Beijing, with the largest area and the largest number of trains.
  - Both problems are challenging scheduling tasks, which can be formulated as Mixed Integer Programming (MIP).

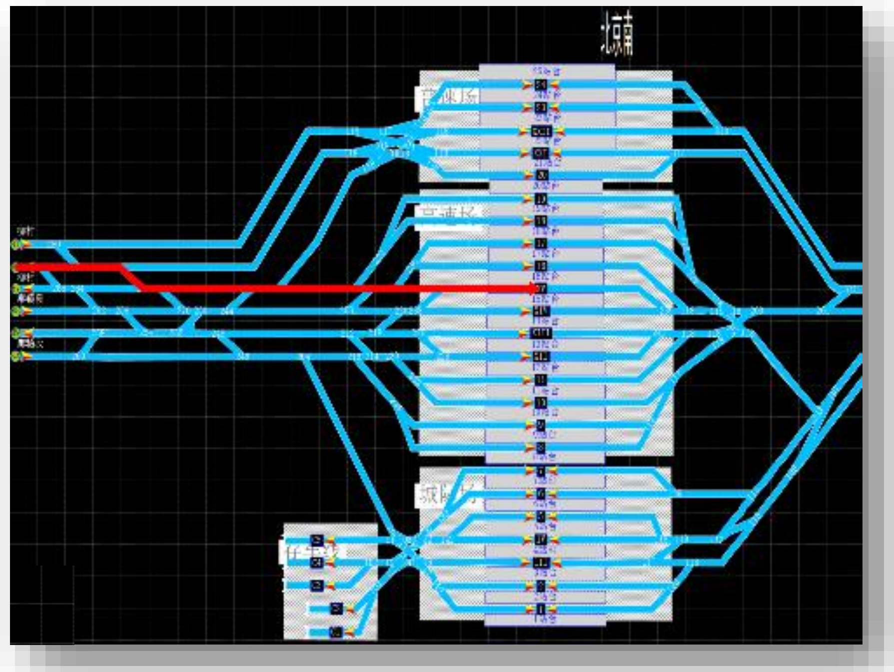
Numerical Results: TTP for Beijing-Shanghai

- We solve the TTP for Beijing-Shanghai high-speed railway using Cardinal Optimizer (COPT).
- COPT is the first fully independently developed mathematical programming solver in China with strong solving ability of MIP problem. It also has excellent performance in solving this problem.
- The result is presented in the following figure. We only need about **1000 seconds** to schedule 584 train in two directions.



Numerical Results: TPP at Beijingnan Station

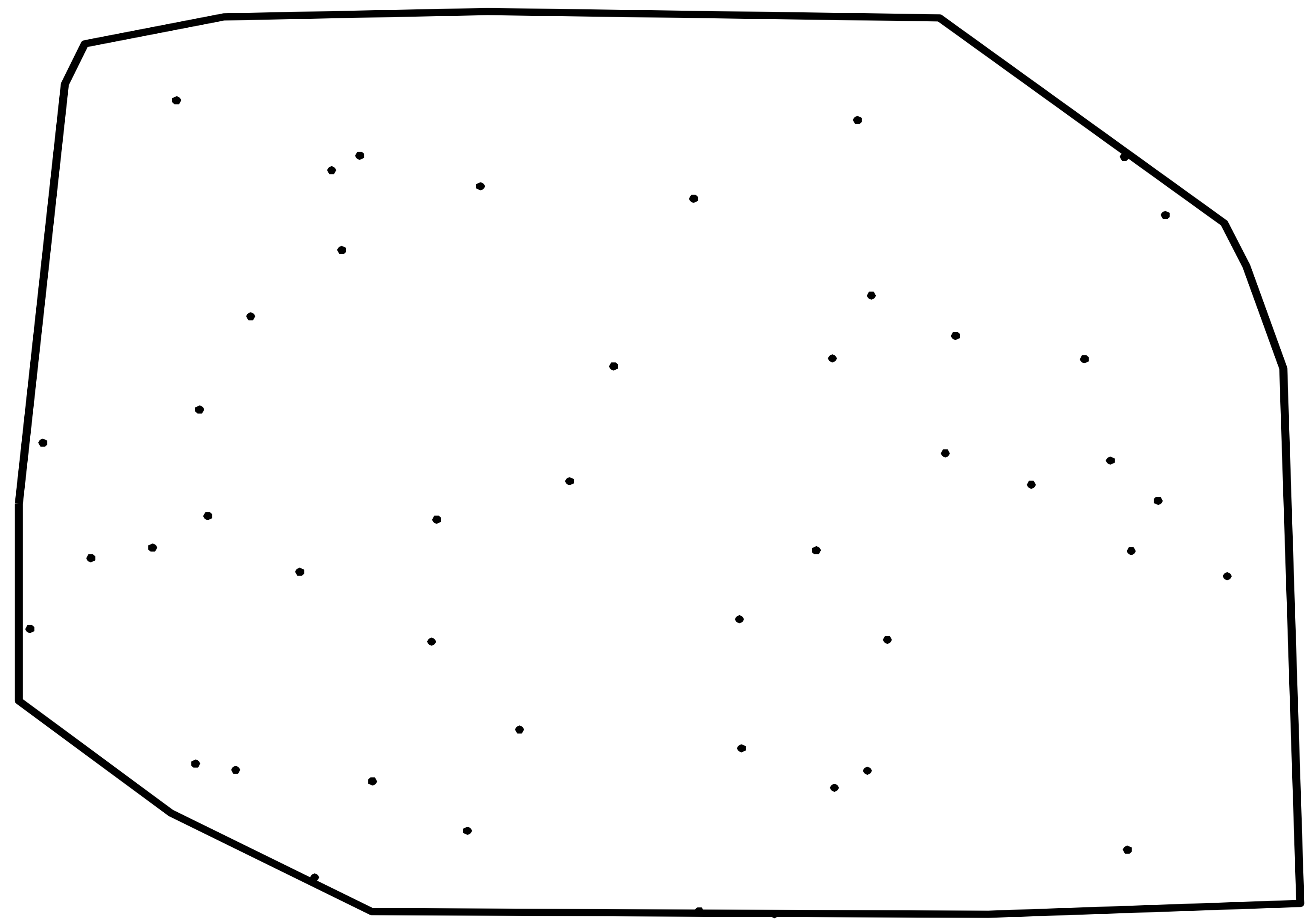
- We solve the TPP at Beijingnan Railway Station using Cardinal Optimizer (COPT).
- Considering the connection pairs and ensuring the feasibility, we solve the model within **2 hours**, which is much less than manual scheduling.
- The result is presented in the following table, including time nodes about occupation at boundaries and tracks.



列车编号	前序车站	进入站界	进站路径	停靠站线	离开站界	出站路径	后序车站	进入站界时间	进入站线时间	离开站线时间	离开站界时间
361		站界:B10		站线:XIV	站界:B9	站线:10:XIV	廊坊		12:00:00	12:06:00	12:10:00
74	廊坊	站界:B8	站线:16:8	站线:8	站界:B7			11:57:00	12:02:00	12:17:00	
125		站界:B10		站线:11	站界:B9	站线:13:11	廊坊		12:06:00	12:13:00	12:17:00
114	廊坊	站界:B8	站线:7:17	站线:17	站界:B7			12:10:00	12:14:00	12:29:00	
251		站界:B10		站线:8	站界:B9	站线:16:8	廊坊		12:17:00	12:27:00	12:32:00
20	廊坊	站界:B8	站线:7:17	站线:17	站界:B7	站线:7:17		12:19:00	12:23:00	12:25:00	12:29:00
96	廊坊	站界:B8	站线:13:11	站线:11	站界:B7			12:25:00	12:29:00	12:44:00	
223		站界:B10		站线:17	站界:B9	站线:7:17	廊坊		12:29:00	12:44:00	12:48:00
8	廊坊	站界:B8	站线:8:16	站线:16	站界:B7			12:33:00	12:37:00	12:42:00	
23		站界:B10		站线:16	站界:B9	站线:8:16	廊坊		12:42:00	12:57:00	13:01:00
127		站界:B10		站线:11	站界:B9	站线:13:11	廊坊		12:44:00	12:49:00	12:53:00
572	廊坊	站界:B8	站线:5:19	站线:19	站界:B7			12:43:00	12:48:00	13:03:00	
124	廊坊	站界:B8	站线:6:18	站线:18	站界:B7			12:47:00	12:52:00	12:57:00	
102	廊坊	站界:B8	站线:15:9	站线:9	站界:B7			12:51:00	12:56:00	13:07:00	
225		站界:B10		站线:18	站界:B9	站线:6:18	廊坊		12:57:00	13:12:00	13:17:00
51		站界:B10		站线:17	站界:B9	站线:7:17	廊坊		12:59:00	13:01:00	13:05:00
116	廊坊	站界:B8	站线:13:11	站线:11	站界:B7			12:56:00	13:00:00	13:15:00	
169		站界:B10		站线:19	站界:B9	站线:5:19	廊坊		13:03:00	13:18:00	13:23:00
133		站界:B10		站线:9	站界:B9	站线:15:9	廊坊		13:07:00	13:22:00	13:27:00
161		站界:B10		站线:11	站界:B9	站线:13:11	廊坊		13:15:00	13:26:00	13:30:00
138	廊坊	站界:B8	站线:5:19	站线:19	站界:B7			13:13:00	13:18:00	13:33:00	
118	廊坊	站界:B8	站线:8:16	站线:16	站界:B7			13:27:00	13:31:00	13:36:00	
109		站界:B10		站线:19	站界:B9	站线:5:19	廊坊		13:33:00	13:41:00	13:46:00
100	廊坊	站界:B8	站线:8:16	站线:16	站界:B7			13:31:00	13:35:00	13:40:00	
229		站界:B10		站线:16	站界:B9	站线:8:16	廊坊		13:36:00	13:51:00	13:55:00
2	廊坊	站界:B8	站线:16:8	站线:8	站界:B7			13:34:00	13:39:00	13:47:00	
131		站界:B10		站线:16	站界:B9	站线:8:16	廊坊		13:40:00	13:55:00	13:59:00
3		站界:B10		站线:8	站界:B9	站线:16:8	廊坊		13:47:00	14:02:00	14:07:00
98	廊坊	站界:B8	站线:10:XIV	站线:XIV	站界:B7			13:43:00	13:47:00	14:02:00	
108	廊坊	站界:B8	站线:13:11	站线:11	站界:B7			13:47:00	13:51:00	14:06:00	

Topic 4: Equitable Covering & Partition – Divide and Conquer (Carlsson et al. 2009)

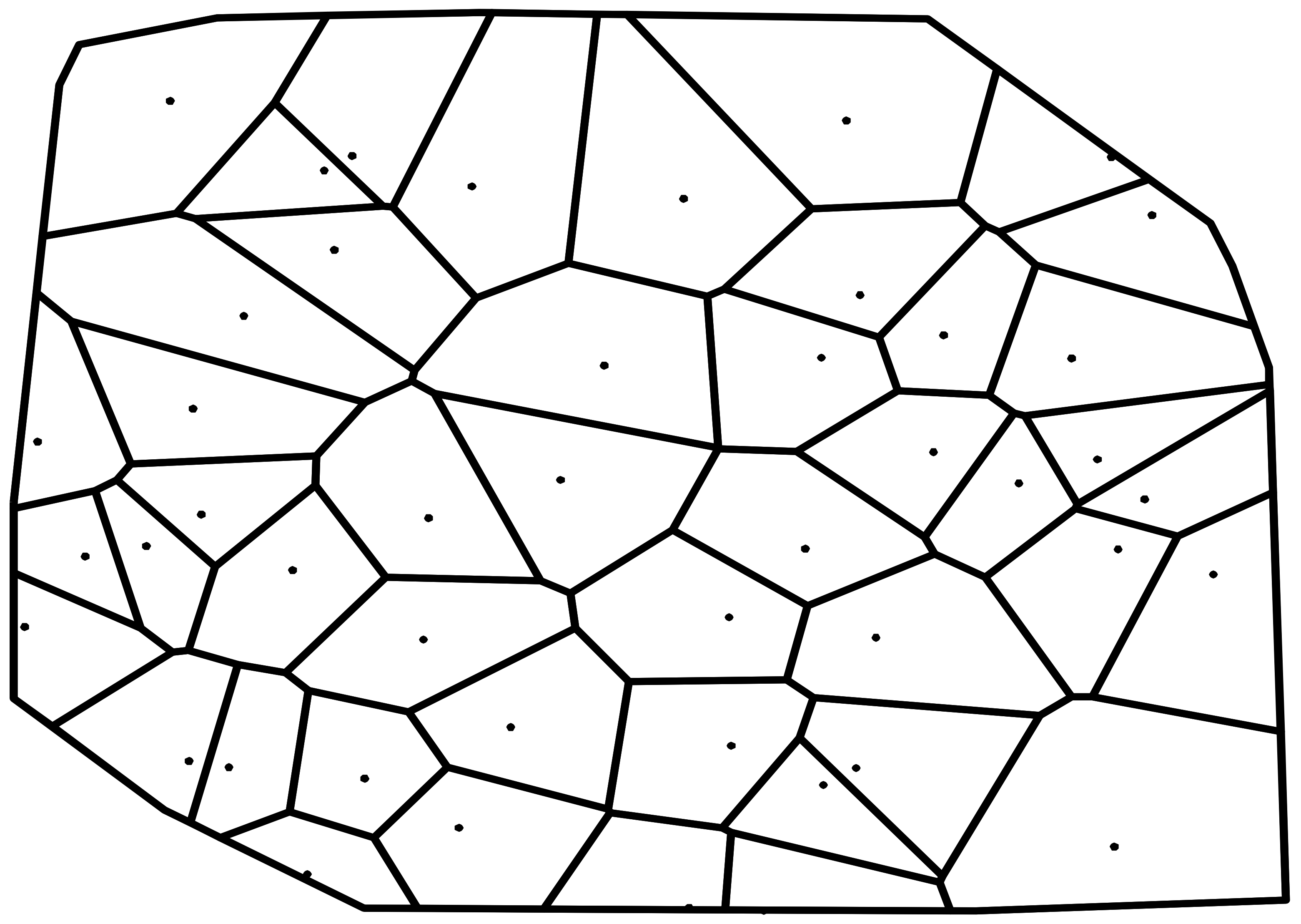
Problem Statement: Divide-Conquer



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

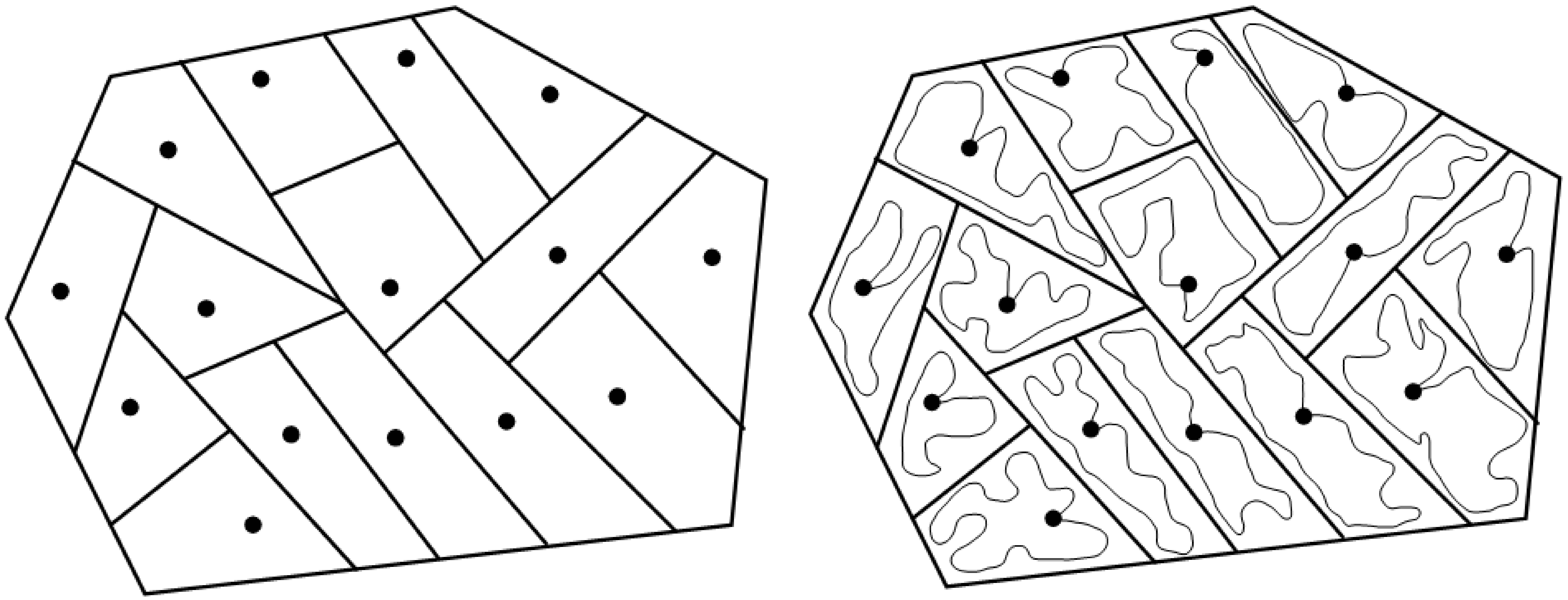
Related ML Problem: Voronoi Diagram



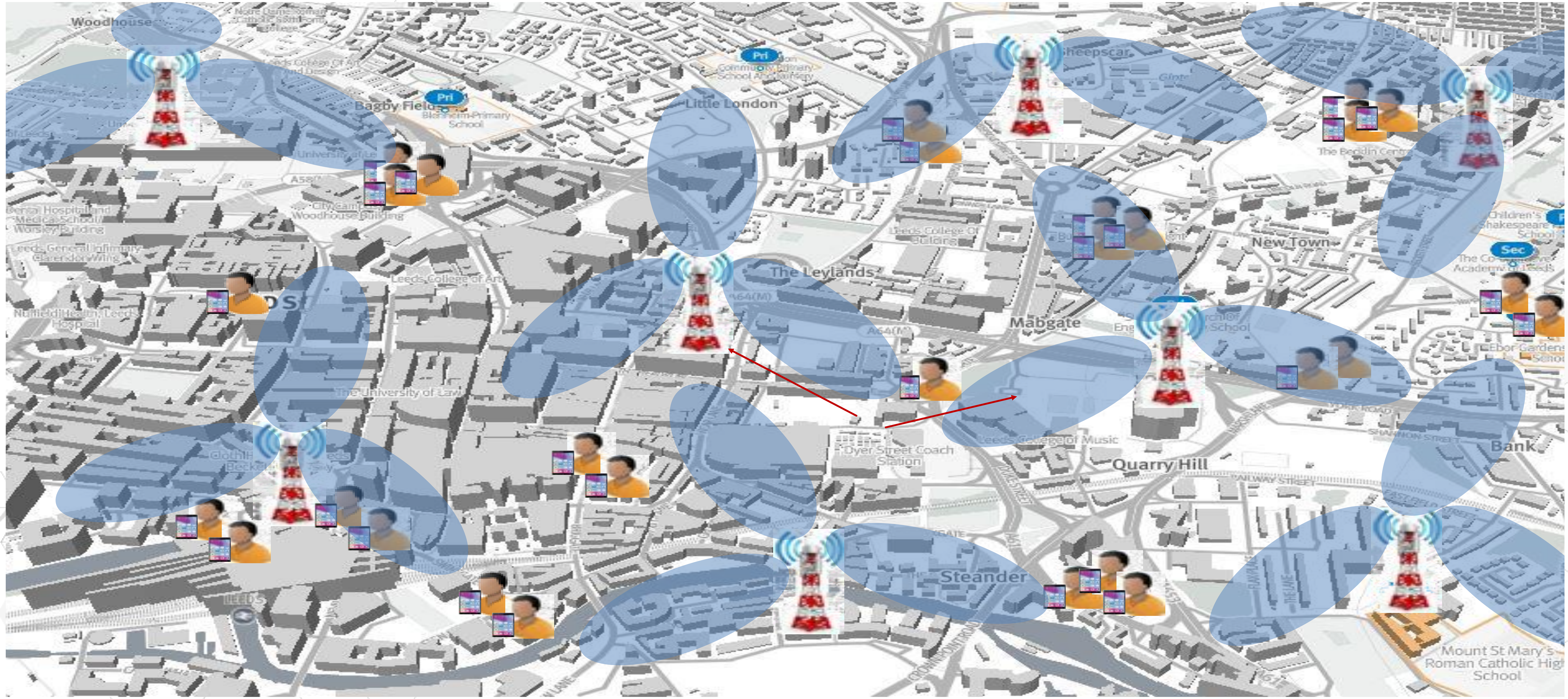
In the *Voronoi Diagram*, we satisfy the first two properties (each sub-region is convex and contains one point), but the sub-regions have different areas.

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. IX: Wireless Tower – Dynamic Resource Allocation

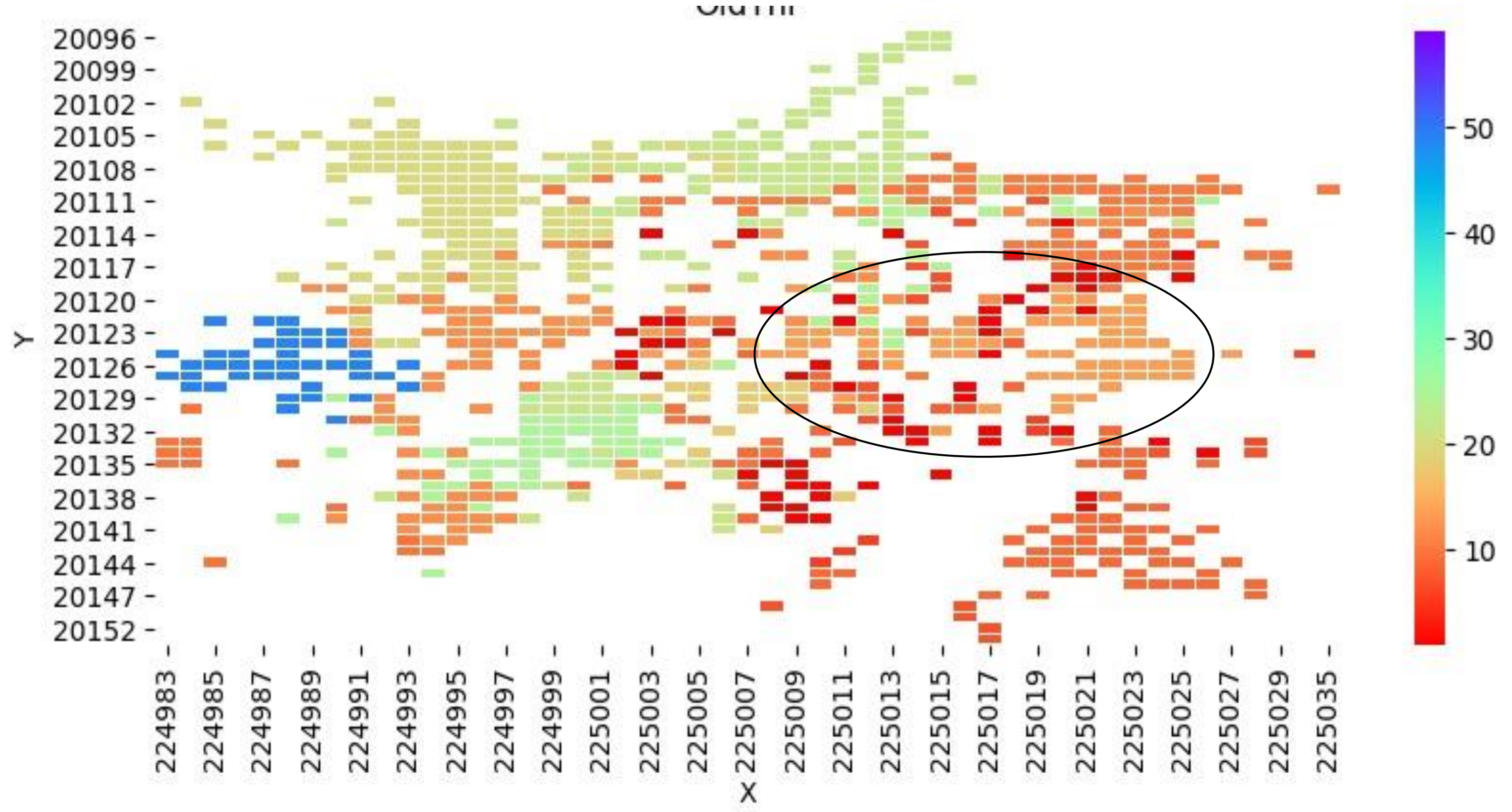


Preliminary Test Result—Effectiveness

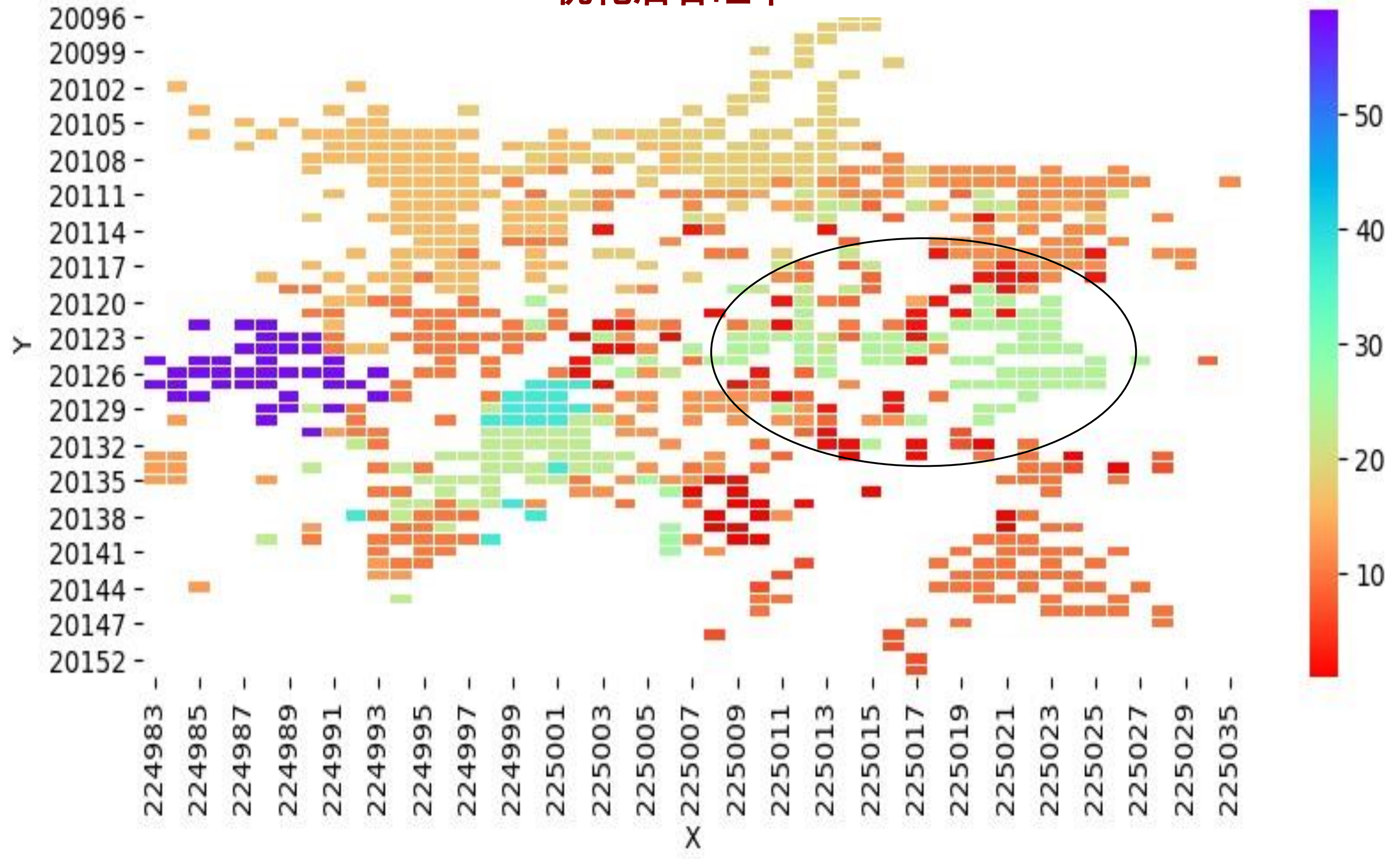
基于真实商用网络进行模型优化效果的测试验证验证统计结果：

	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%)

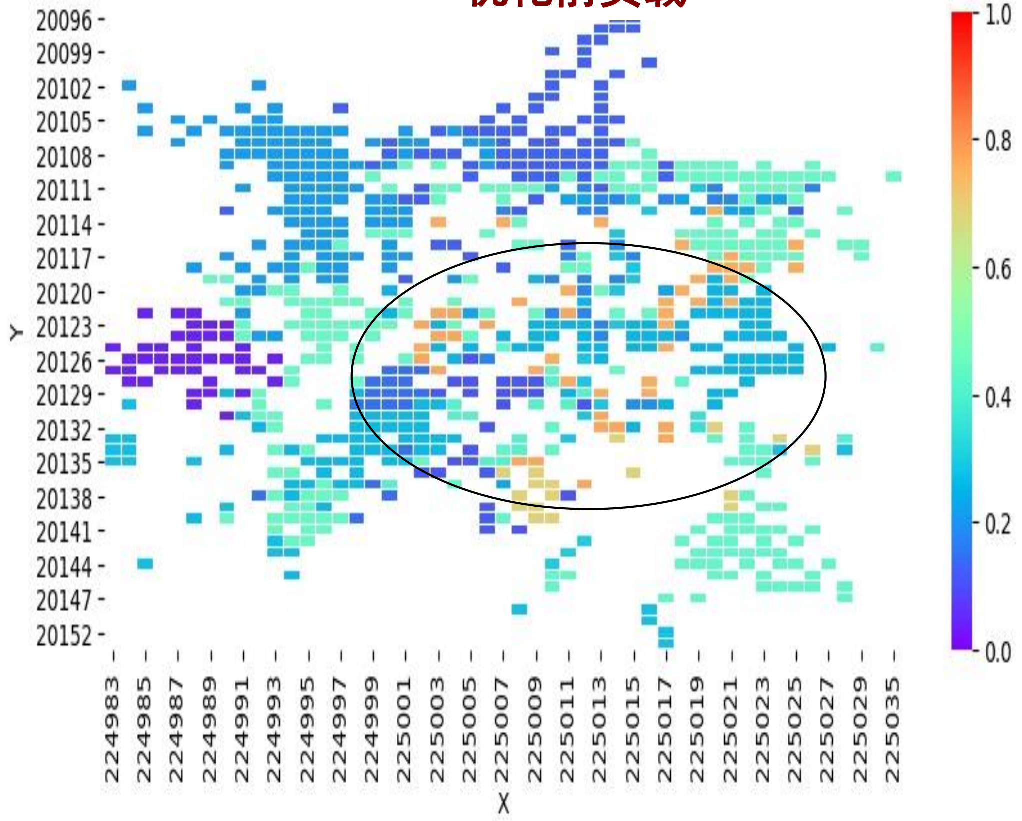
优化前吞吐率



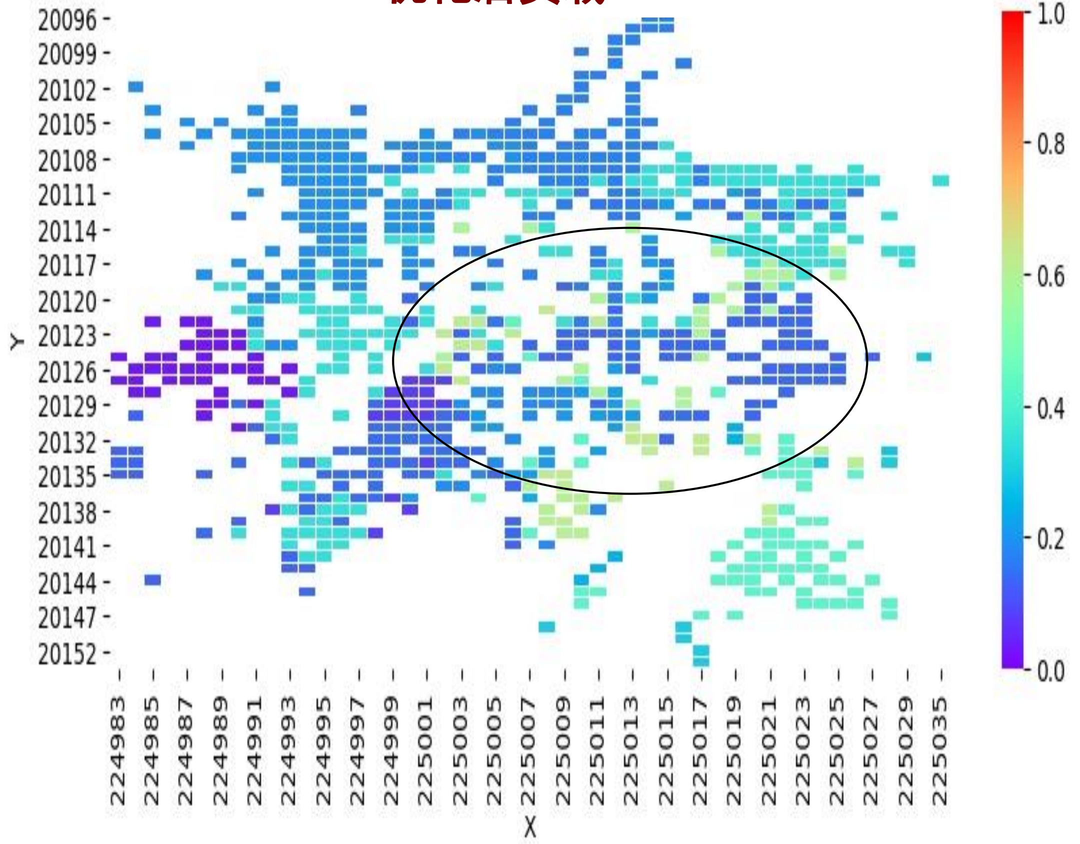
优化后吞吐率



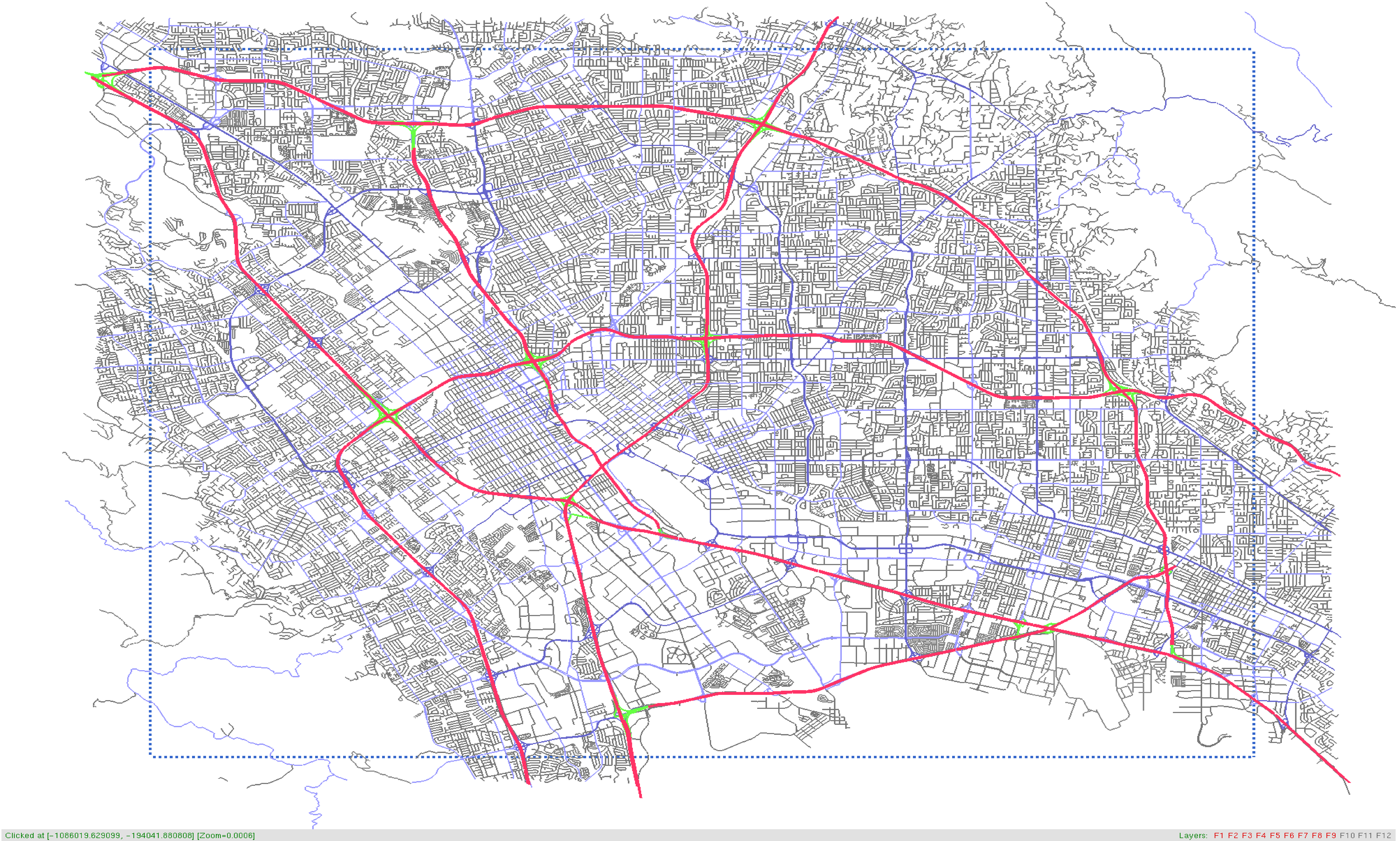
优化前负载



优化后负载



App. X: 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!

Mixed Integer LP solvers benefit real economy

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

- THANK YOU