AI Big-Model and OR Mathematical Optimization

AUGUST 11, 2023

Yinyu Ye
Stanford University and CUHKSZ (Sabbatical Leave)

Today’s Sharing

1. AI Big-Model and OR/Math Optimization
2. Online Training Algorithms and Applications
3. Pre-Trained Mixed Integer Linear Programming Solvers and Applications
4. Big-Model Training Needs Efficient Optimization Algorithms?

ChatGPT的能力边界 | 求解数学优化问题?

- 优化问题根据变量、约束和目标函数的特性，可以分为不同种类，如线性规划、非线性规划和整数规划等
- 小规模整数优化问题，可以通过穷举法求解
- 大规模整数规划问题一般由专业的整数规划软件求解（Gurobi, COPT）

一个简单的数学规划问题

假设有一个最多能承重5kg的背包和5个重量不同的物品，每个物品都具有一定的价值。将哪些物品装入背包，可以使得背包内的物品总价值最大?

物品的重量和价值如下表所示:

物品	1	2	3	4	5
重量 (kg)	2	1	2.5	3.5	1.5
价值 (元)	18	10	20	28	15

使用ChatGPT求解科学决策问题

将上述背包问题向ChatGPT提问：

假设有5件物品和一个最多能承重5kg的背包。第1件物品的重量是2kg，其价值为18元；第2件物品的重量是1kg，其价值为10元；第3件物品的重量是2.5kg，其价值为20元；第4件物品的重量是3.5kg，其价值为28元；第5件物品的重量是1.5kg，其价值为15元。请问将哪些物品装入背包里，物品的价值总和最大？请使用整数规划求解该问题。

这是一个经典的背包问题，可以使用整数规划方法求解。

我们可以定义0/1变量x_i，表示是否将第i件物品放入背包中。同时，我们还需要定义一个非负整数变量z，表示物品的总价值。

然后，我们可以将问题转化为以下的线性规划问题：

最大化 $z=18x_1+10x_2+20x_3+28x_4+15x_5$

满足约束条件：

$2x_1+x_2+2.5x_3+3.5x_4+1.5x_5 \leq 5$

$x_i\in {0,1}, i=1,2,3,4,5$

由于x_i是整数变量，这是一个整数规划问题。可以使用整数规划求解器求解。

求解结果表明，将第1件、第3件、第5件物品装入背包中，可以使得物品的总价值最大，为$18+20+15=53$元。

ChatGPT 在不调用求解器的前提下，能否求解大规模优化问题？

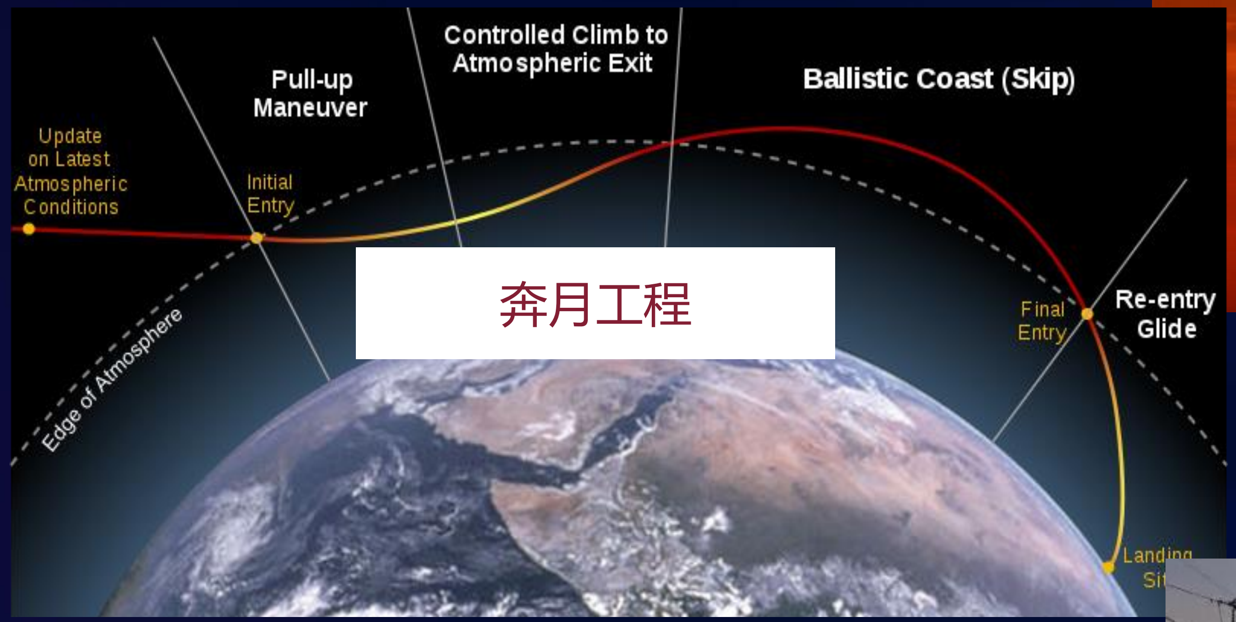
在不调用求解器的前提下，对于大规模的优化问题，通常也比较难直接求解得到最优解，因为现实中的大多数优化问题都非常复杂，存在着多个目标、多个约束、非线性、离散等多种问题，很容易遇到组合爆炸的问题。

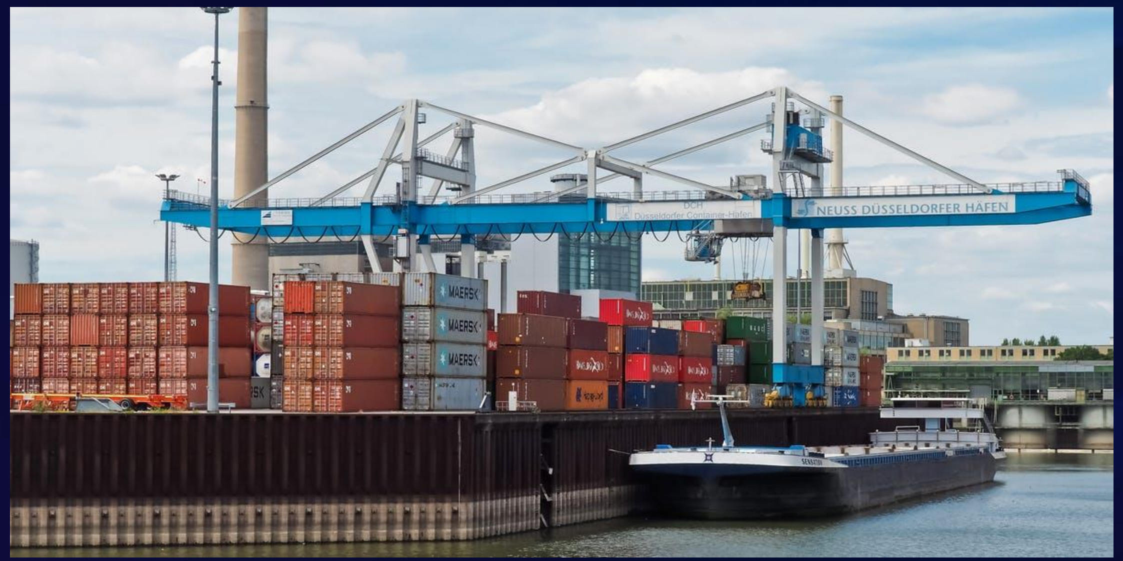
向ChatGPT提问：

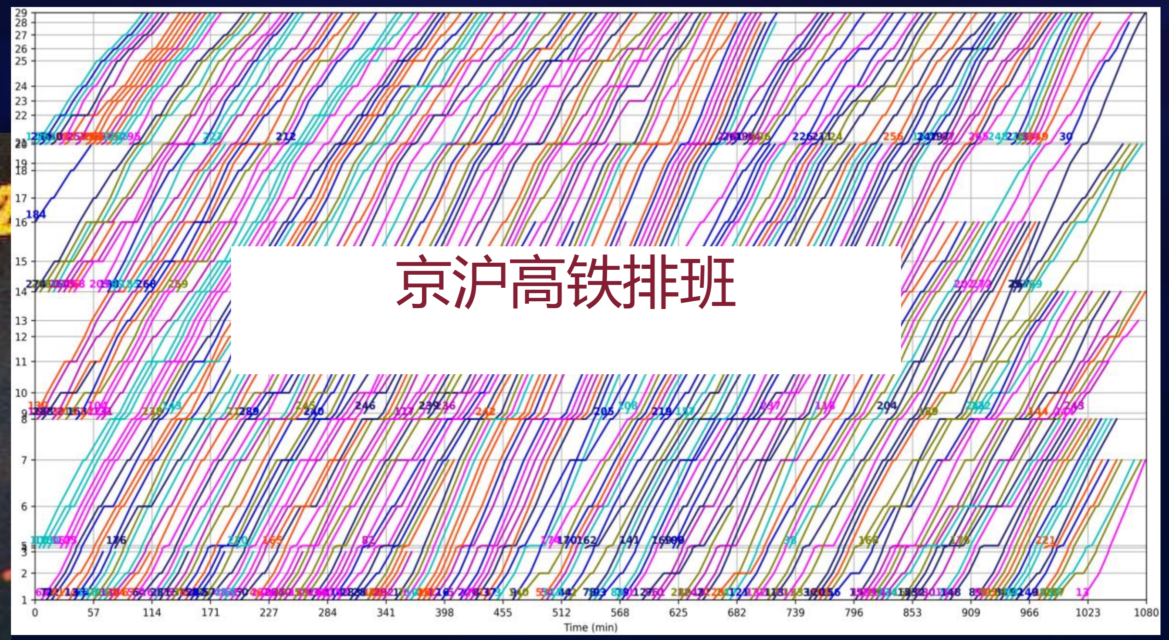
- ChatGPT 会尝试性给出解答，但是不能保证答案的正确性。
- ChatGPT会建议调用求解器求解优化问题：全科vs专科
- ChatGPT 并不是万能的，它是一个语言处理模型，回答复杂的计算问题还需要调用专业化的软件。

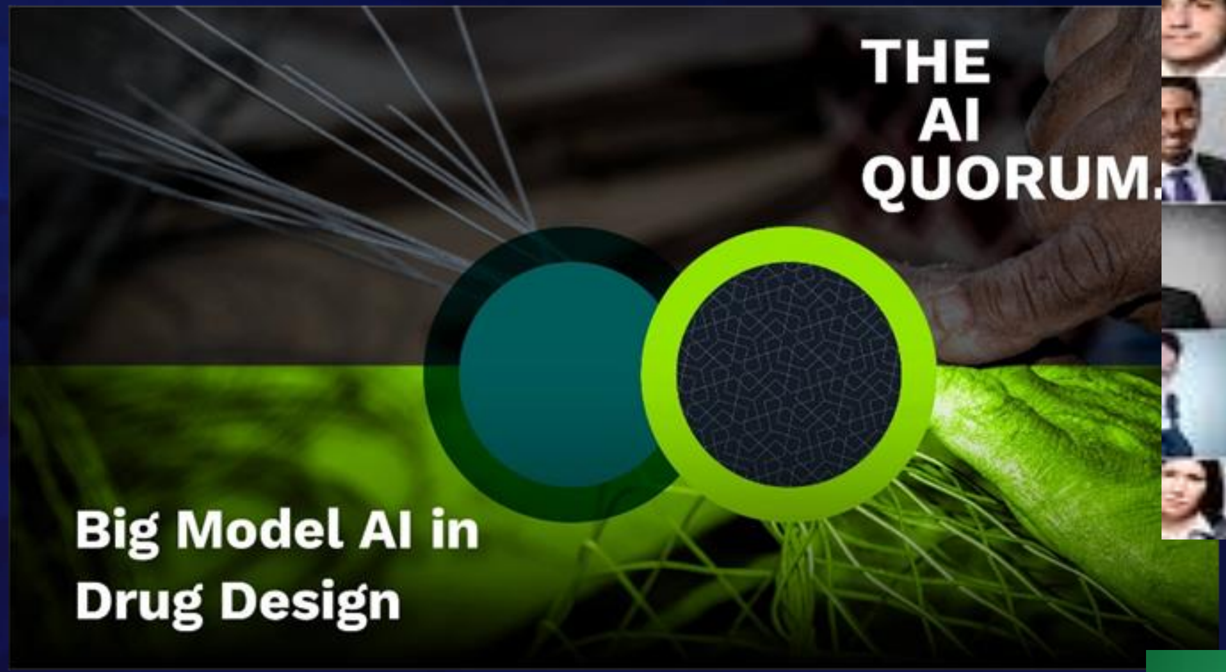
上述问答产生于2023年3月12日

Applications of OR and AI Models

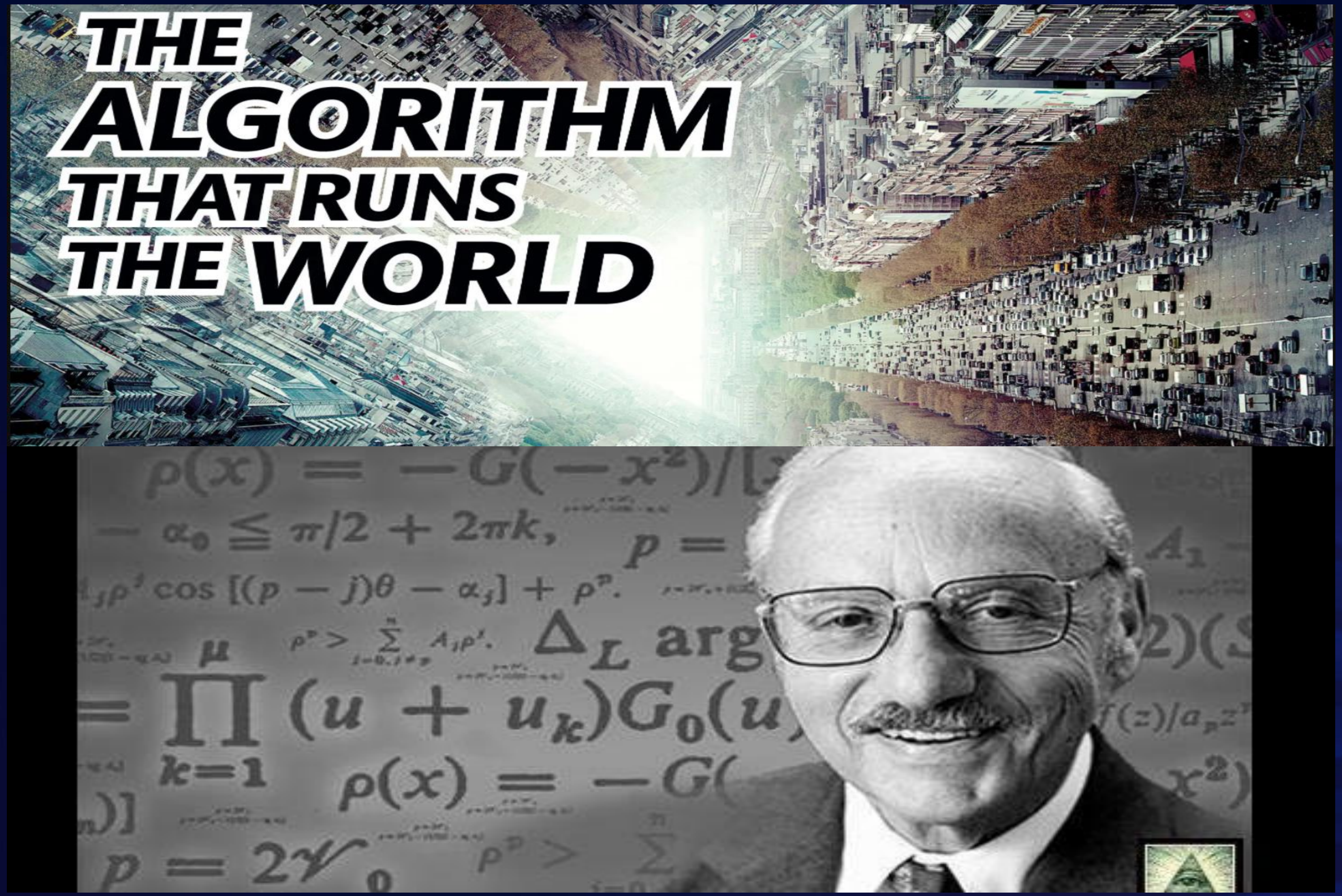








Differences of OR and AI Models



- Based on Science/Logic
- Physical/Economical Principles
- Objective
- Definitive
- Explainable Insights
- Online Training&Decision-Making

- Based on Cases/Experience
- Observation/Behaviour
- Subjective
- Probabilistic
- Black-box
- Offline Training

Today’s Sharing

1. AI Model-Training and OR/Math Optimization
2. Online Training Algorithms and Applications
3. Pre-Trained Mixed Integer Linear Programming Solvers and Applications
4. Big-Model Training Needs Efficient Optimization Algorithms?

Linear Programming and LP Giants won Nobel Prize…

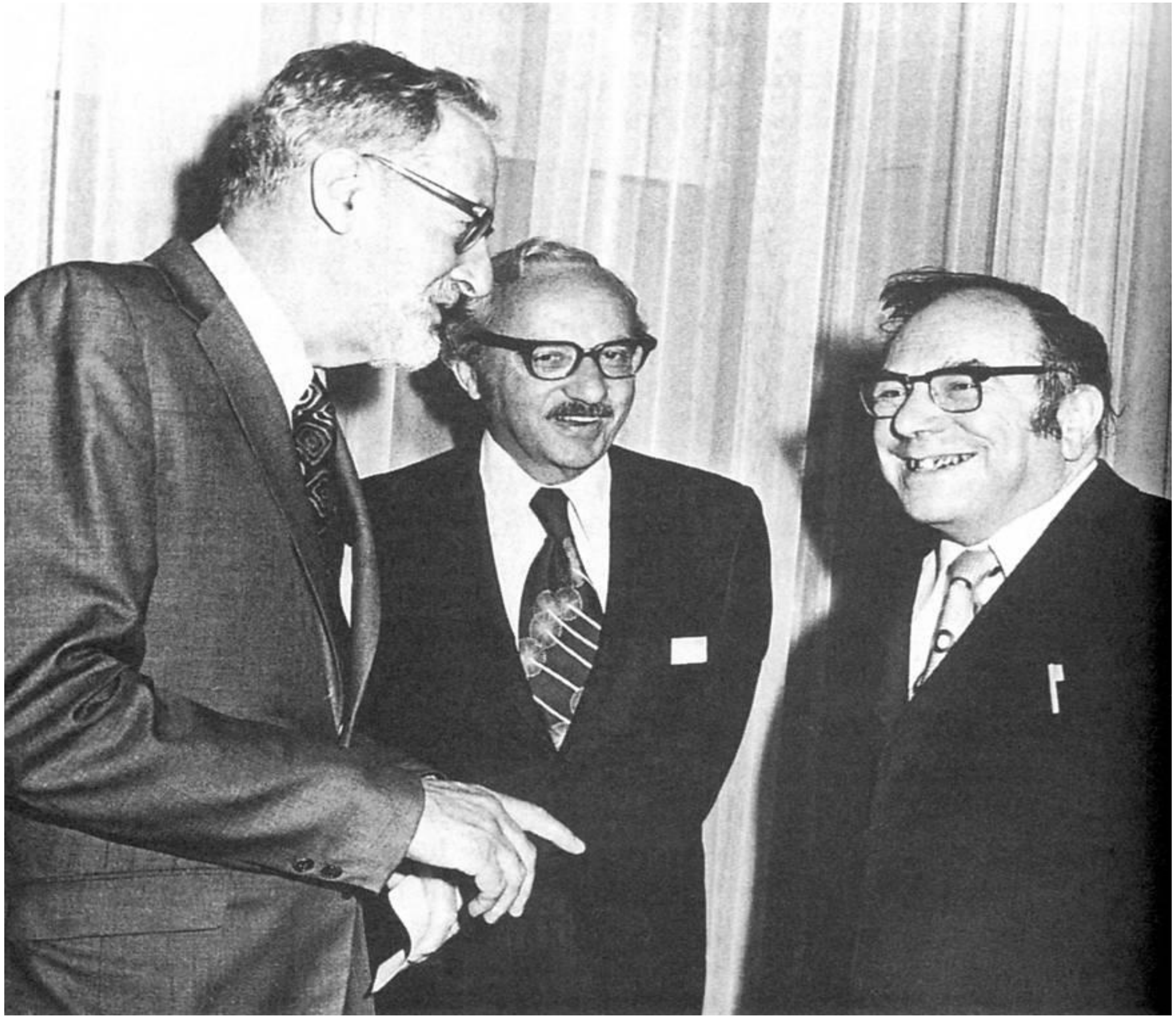
$$\max \quad \sum \pi_j x_j$$

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

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

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

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



Online Linear Programming:
an 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

Online Linear Programming Model and Theory

- OLP theory and practice (Agrawal et al. 2010,14, Li&Y 2022)

- OLP:

$$\begin{aligned} \max \quad & \sum \pi_j x_j \\ \text{s.t.} \quad & \sum_j \boldsymbol{a}_j x_j \le \boldsymbol{b}, \\ & 0 \le x_j \le 1 \quad \forall\, j = 1, \dots, n \end{aligned}$$

  - Variables together with their data points arrive sequentially and decision makers need decide x_j on the fly, that is, before knowing the "future" data points
  - Learning-while-Doing vs Learning-First and Deciding-Second (collect and learn all relevant data, then solve for all ***x***)
  - Offline LP's objective value is a upper bond for the online version
  - Is there an optimal online decision algorithm/mechanism

Price Mechanism for Online Auction

- Learn and compute itemized optimal prices
- Use the prices to price each bid internally
- Accept if it is a over bid, and reject otherwise
- There is an Optimal Online Algorithm to achieve the best you could do!

Massive episodes are transferred into Knowledges that can be stored/reused

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

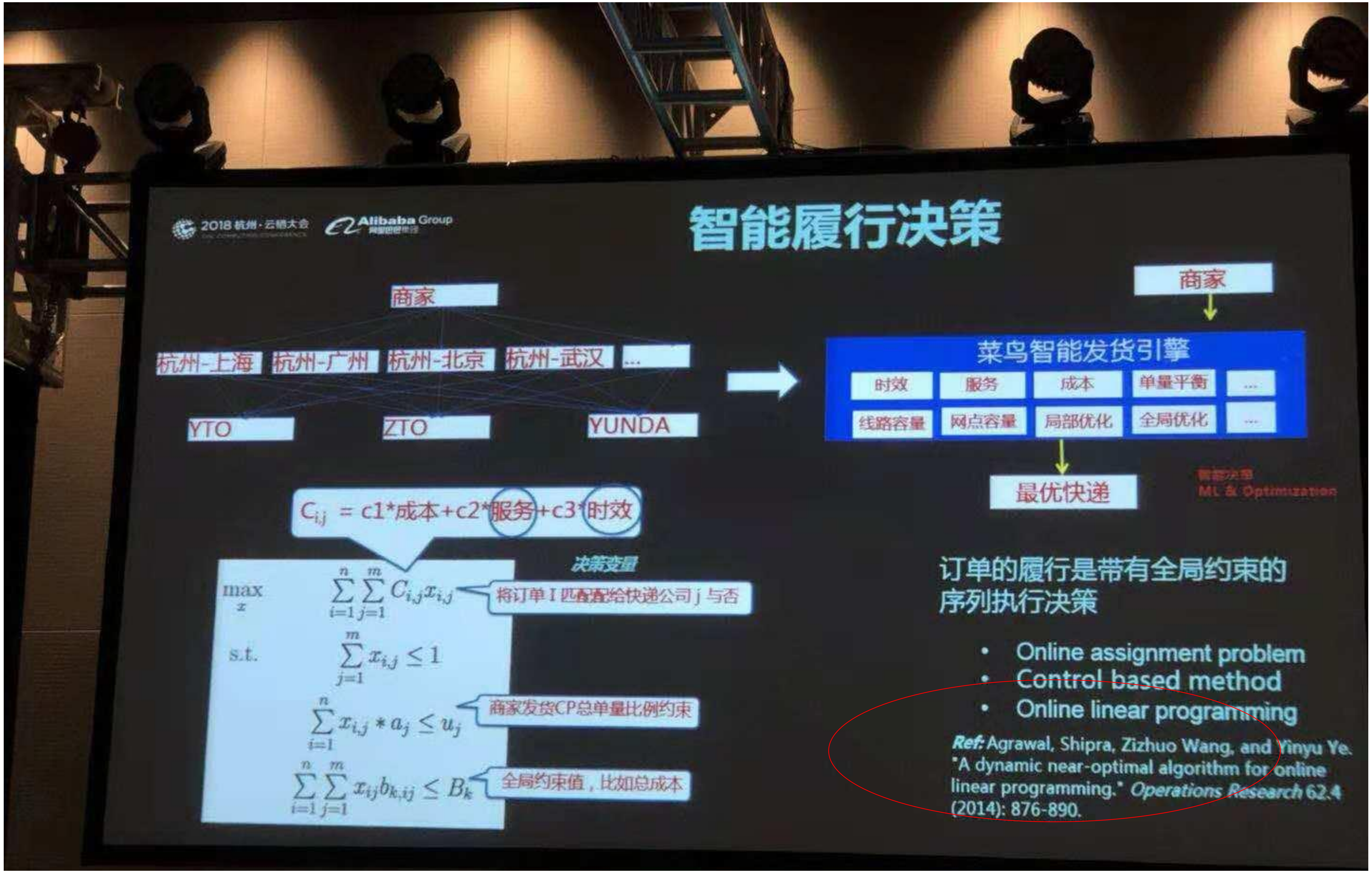
The Online Algorithm can be Applied to Bandits with Knapsack (BwK) Applications

- 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

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

- 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 $\boldsymbol{O}(\log \boldsymbol{T})$ regret/gap bound [Li, Sun & Y 2021 ICML] (**https://proceedings.mlr.press/v139/li21s.html**)

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

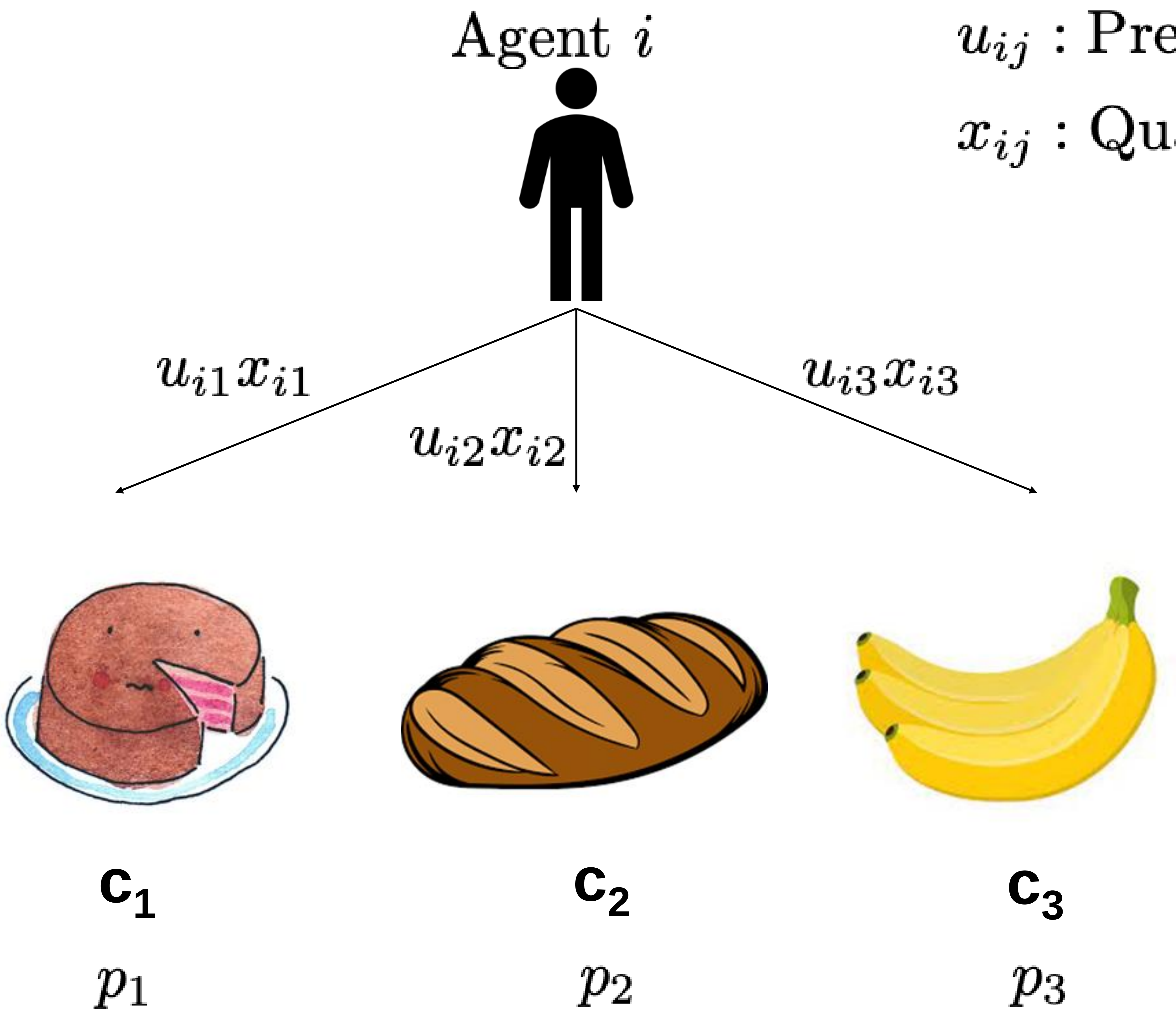
$$
\begin{aligned}
\max \sum_{i=1}^{M}\sum_{j=1}^{N} v_{ij}x_{ij} & \\
s.t. \sum_{i=1}^{M}\sum_{j=1}^{N} c_j x_{ij} \le B, & \\
\sum_{j}^{N} x_{ij} \le 1, \quad \forall i & \\
x_{ij} \ge 0, \quad \forall i, j &
\end{aligned}
\tag{5}
$$

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:

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

A key resource allocation model to achieve efficient allocation is the Fisher Market



M = Total Number of Goods

u_{ij} : Preference of Agent i for one unit of good j

x_{ij} : Quantity of good j purchased by person i

p_j : Price of Good j

w_i : Budget of Agent i

Individual Optimization Problem:

$$\max_{\mathbf{x}_i} \sum_j u_{ij} x_{ij}$$

$$\text{s.t. } \mathbf{p}^T \mathbf{x}_i \leq w_i$$

$$\mathbf{x}_i \geq \mathbf{0}$$

Do Prices exist to CLEAR the market?
(Prices are posted and known to all agents so that they have freedom to choose)

The prices can be derived from a centralized optimization problem with a budget-weighted social objective (Eisenberg-Gale)

Individual Optimization Problem:

$$\max_{\mathbf{x}_i} \sum_j u_{ij} x_{ij}$$

$$\text{s.t. } \mathbf{p}^T \mathbf{x}_i \leq w_i$$

$$\mathbf{x}_i \geq \mathbf{0}$$

$$\Longleftrightarrow$$

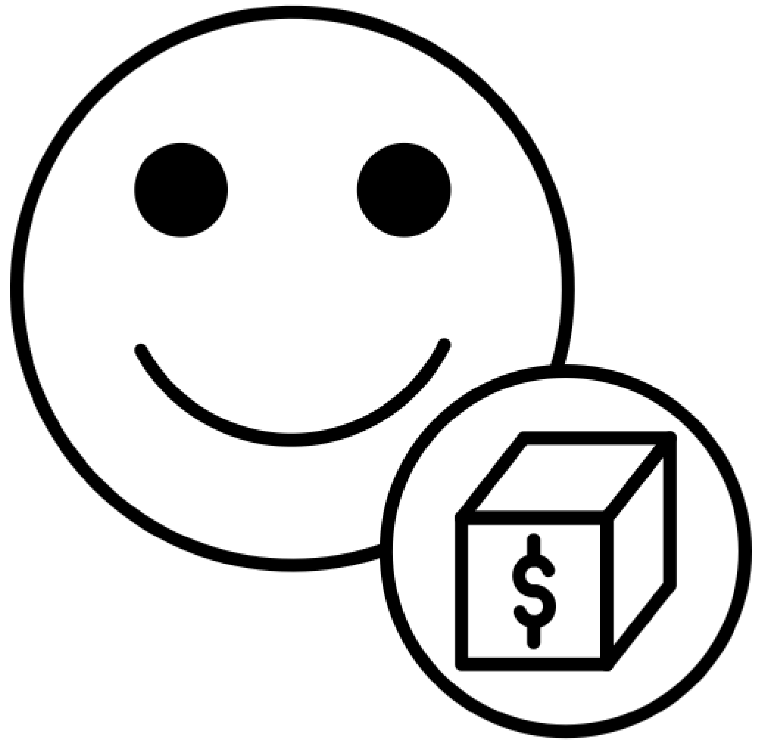
Social Optimization Problem:

$$\max_{\mathbf{x}_i, \forall i \in [N]} \sum_i w_i \log \left(\sum_j u_{ij} x_{ij} \right)$$

$$\text{s.t. } \boxed{\sum_i x_{ij} \leq c_j, \forall j \in [M]}$$

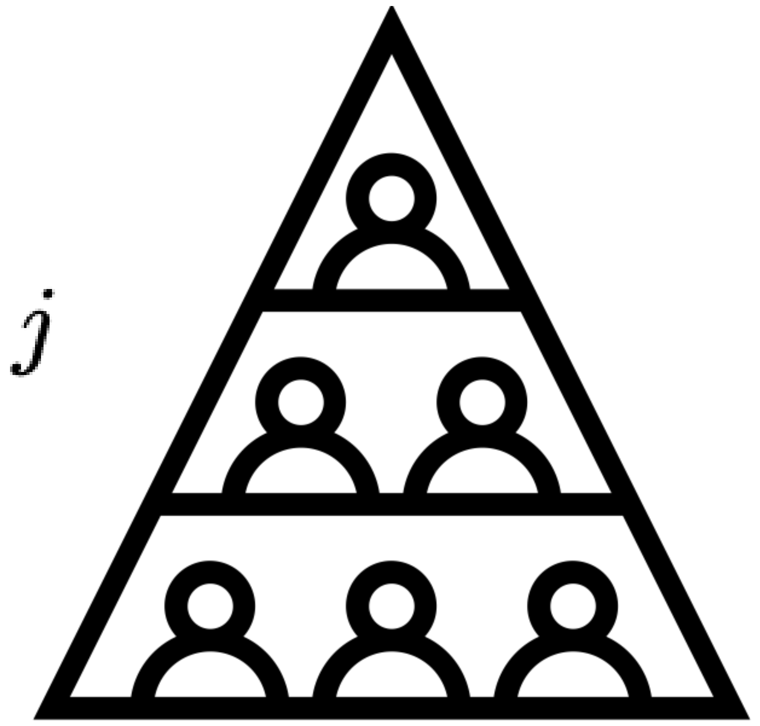
Capacity Constraints

$$x_{ij} \geq 0, \forall i, j$$

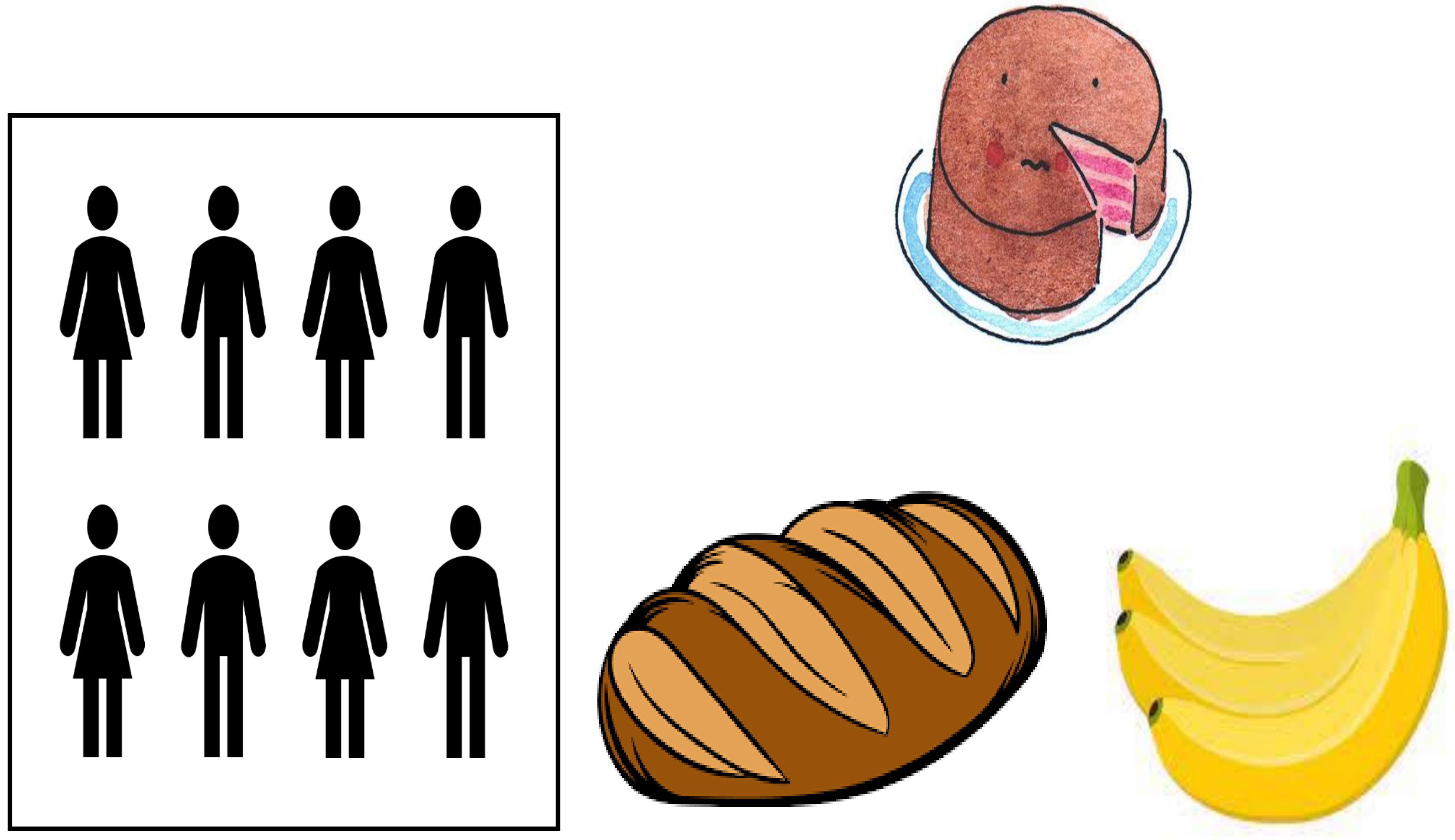


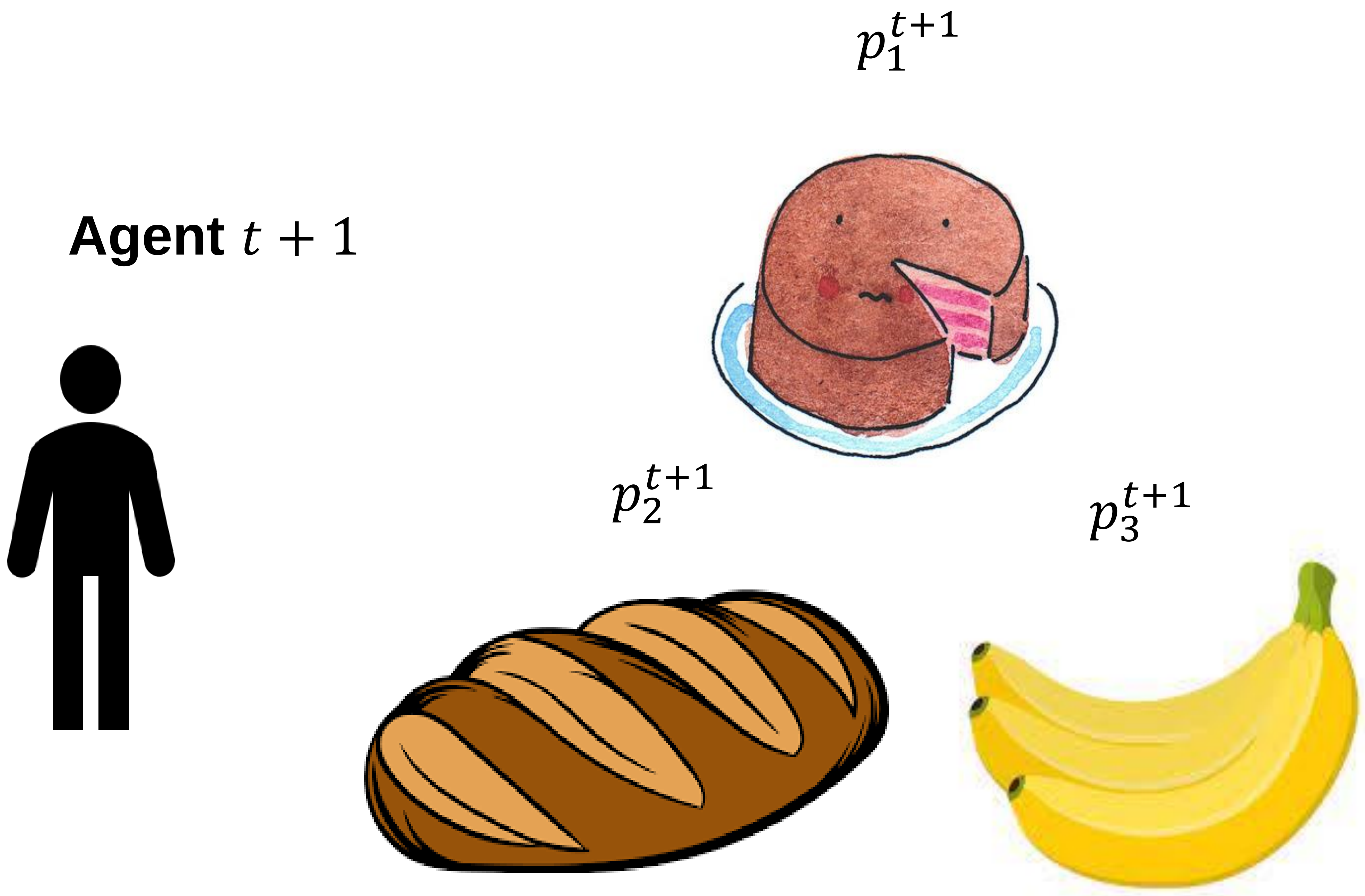
p_j : Price of Good j = Dual Variable of Constraint j

C_j can be decision variables subject to other resource constraints



Online Market Pricing: How to update posted-prices to minimize regret of the Eisenberg/Gale social welfare while achieving market clearness





Static Fisher Market Price Equilibrium Theory and Limit

Now Agents Arrive Online

The price at time $t+1$ is updated and reposted based on observed consumption x^t at time t

Jalota and Y

https://arxiv.org/abs/2205.00825

We develop a revealed preference algorithm with sub-linear regret and constraint violation guarantees

Algorithm 2: Revealed Preference Algorithm for Online Fisher Markets

Input : Number of users n, Vector of good capacities per user $\mathbf{d} = \frac{\mathbf{c}}{n}$

Initialize $\mathbf{p}^1 > \mathbf{0}$;

for $t = 1, 2, ..., n$ **do**

  Phase I: ;

  User purchases an optimal bundle of goods $\mathbf{x}_t$ given the price $\mathbf{p}^t$;

  Phase II (Price Update): ;

  $\mathbf{p}^{t+1} \leftarrow \mathbf{p}^t - \gamma_t \boxed{(\mathbf{d} - \mathbf{x}_t)}$;

end

Difference between market share of each agent and goods purchased

Step-size: $O\left(\frac{1}{\sqrt{n}}\right)$

Only requires knowledge of user consumption (and not their budgets or utilities) to update prices

Theorem: Under i.i.d. budget and utility parameters with strictly positive support and when good capacities are $O(n)$, Algorithm 2 achieves an expected regret of $R_n(\boldsymbol{\pi}) \leq O(\sqrt{n})$ and expected constraint violation of $V_n(\boldsymbol{\pi}) \leq O(\sqrt{n})$, where n is the number of arriving users.

Today’s Sharing

1. AI Model-Training and OR/Math Optimization
2. Online Training Algorithms and Applications
3. Pre-Trained Mixed Integer Linear Programming Solvers and Applications
4. Big-Model Training Needs Efficient Optimization Algorithms?

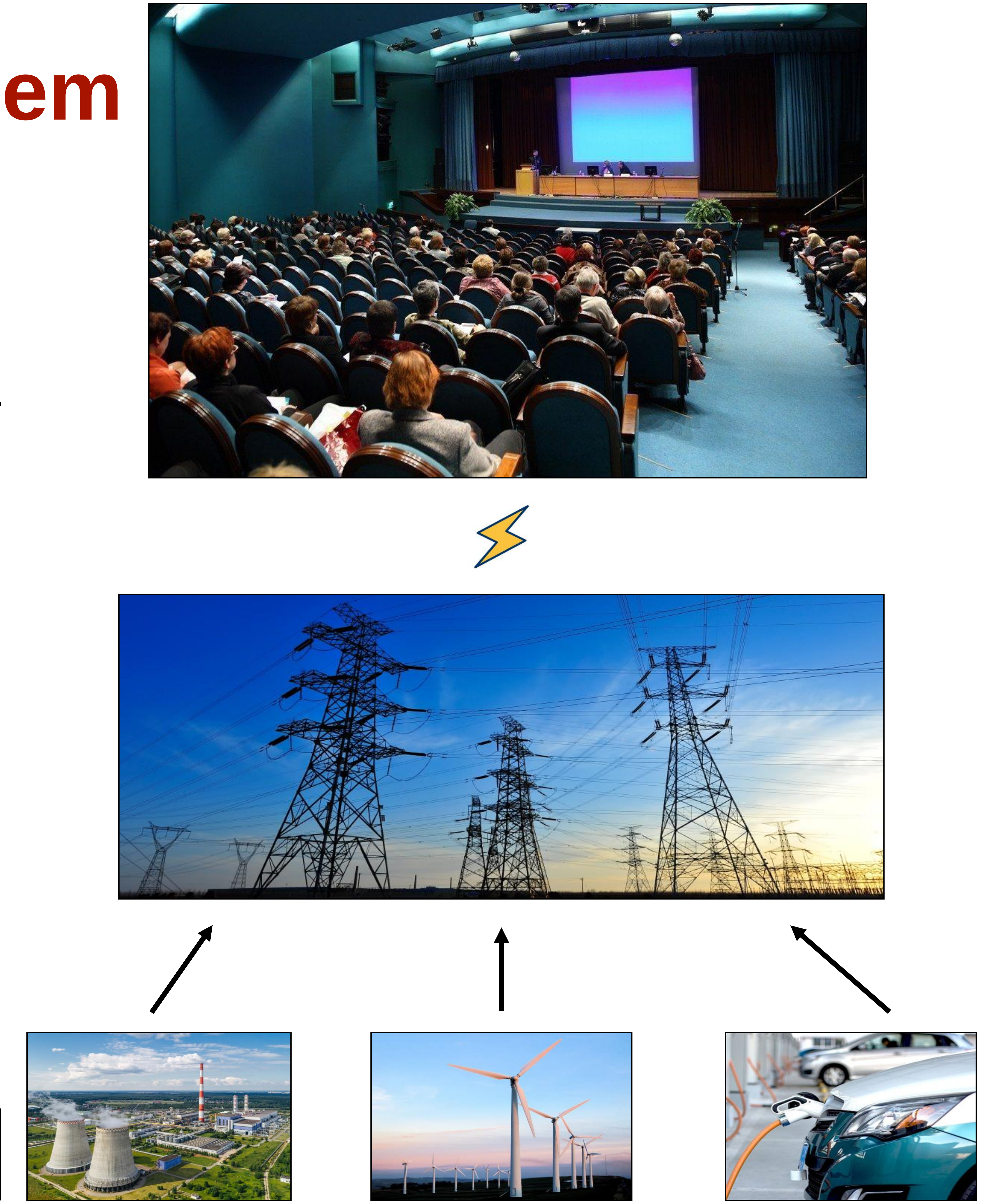
MILPI: Unit Commitment Problem

- 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

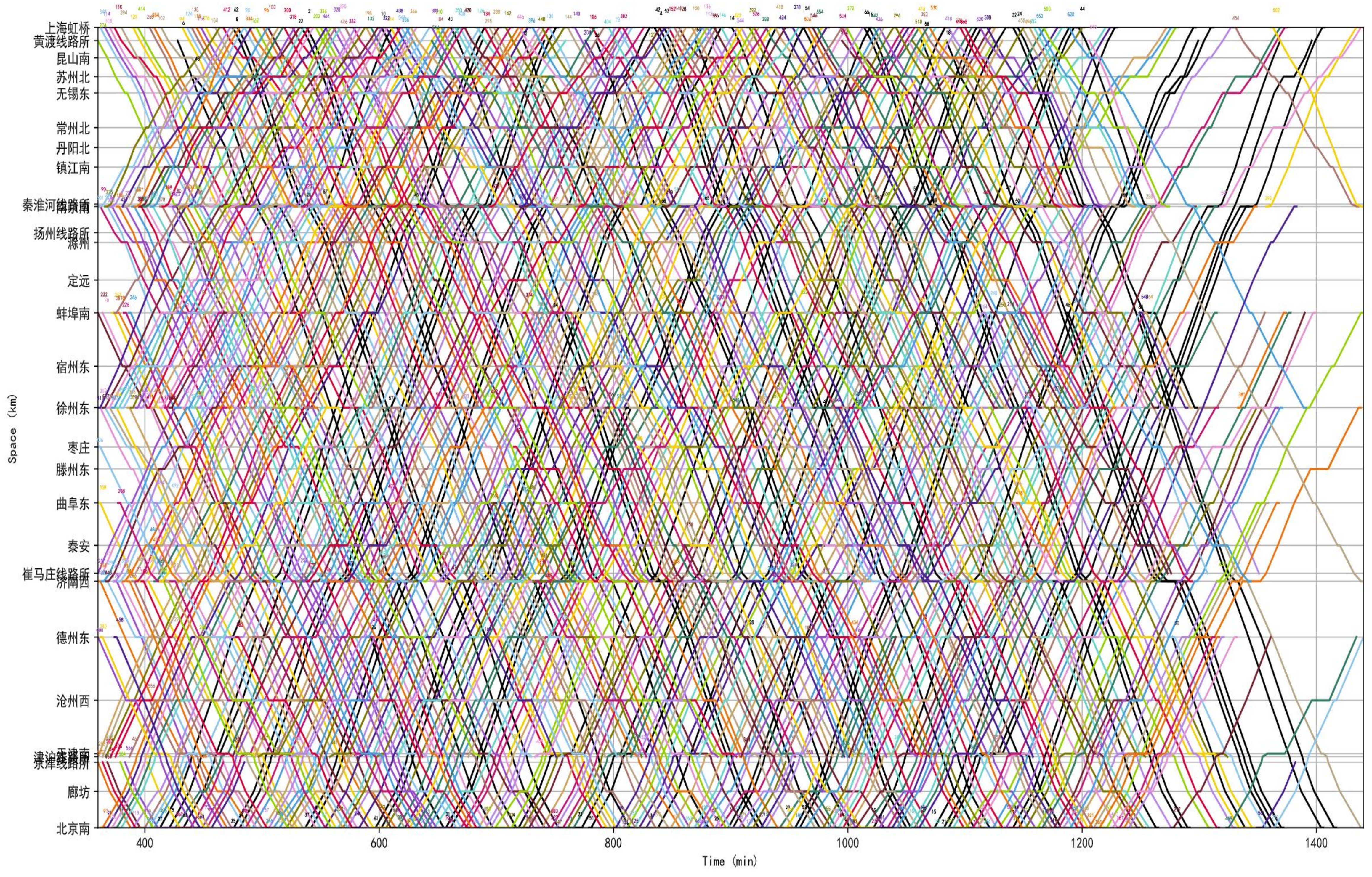
minimize	Cost of electricity
subject to	Safety and Stability
	Adaptivity to various units

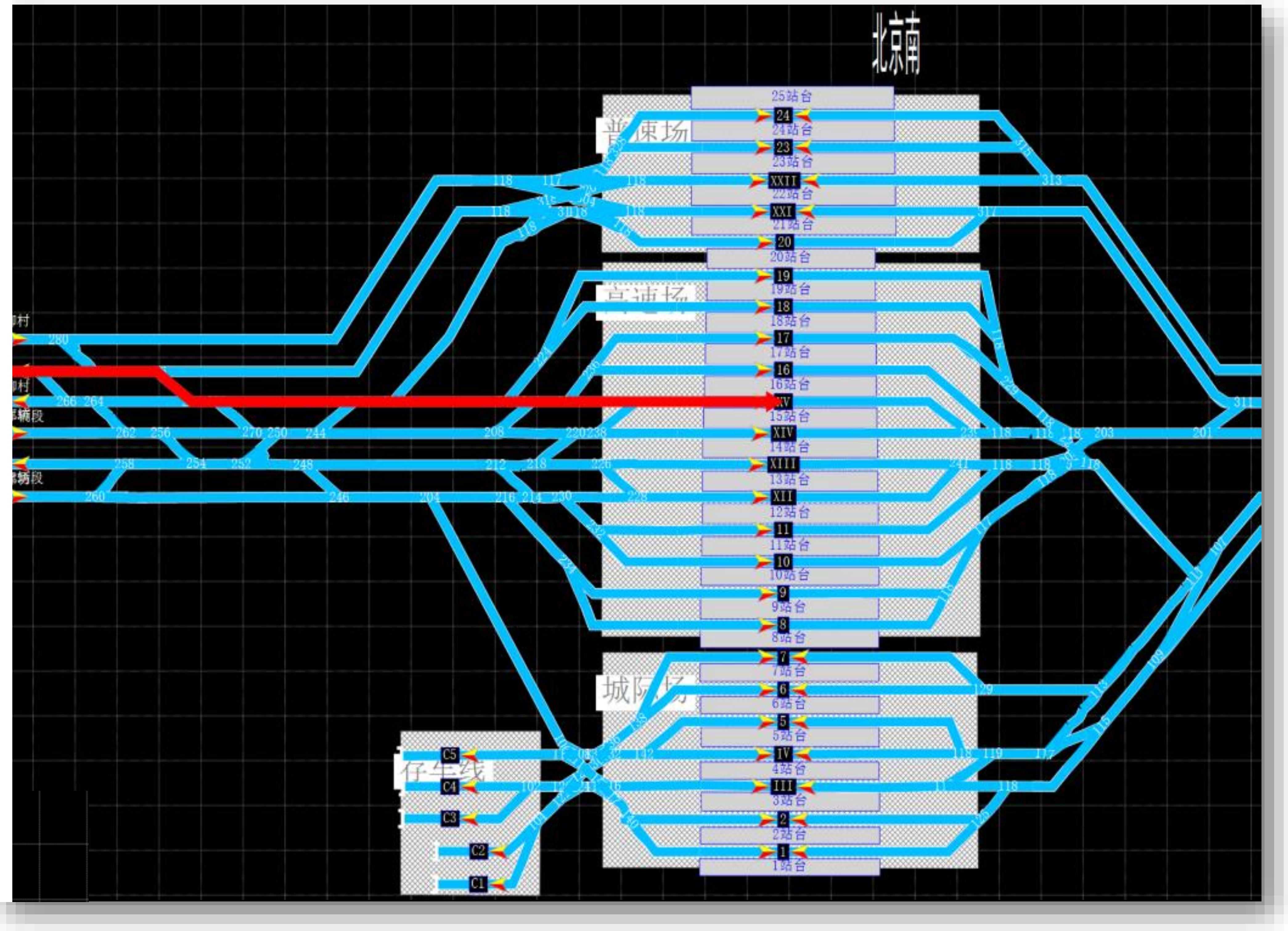
Unit commitment problem dispatches the units
safely and stably at minimum cost



MILPII: Beijing-Shanghai High-speed Railway Scheduling Optimization

COPT, Cardinal Operations 2022





Risk-Pooling Cuts in MILP

- Given an MILP, the fraction solution 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 Cuts via Moment Ambiguity of DRO

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

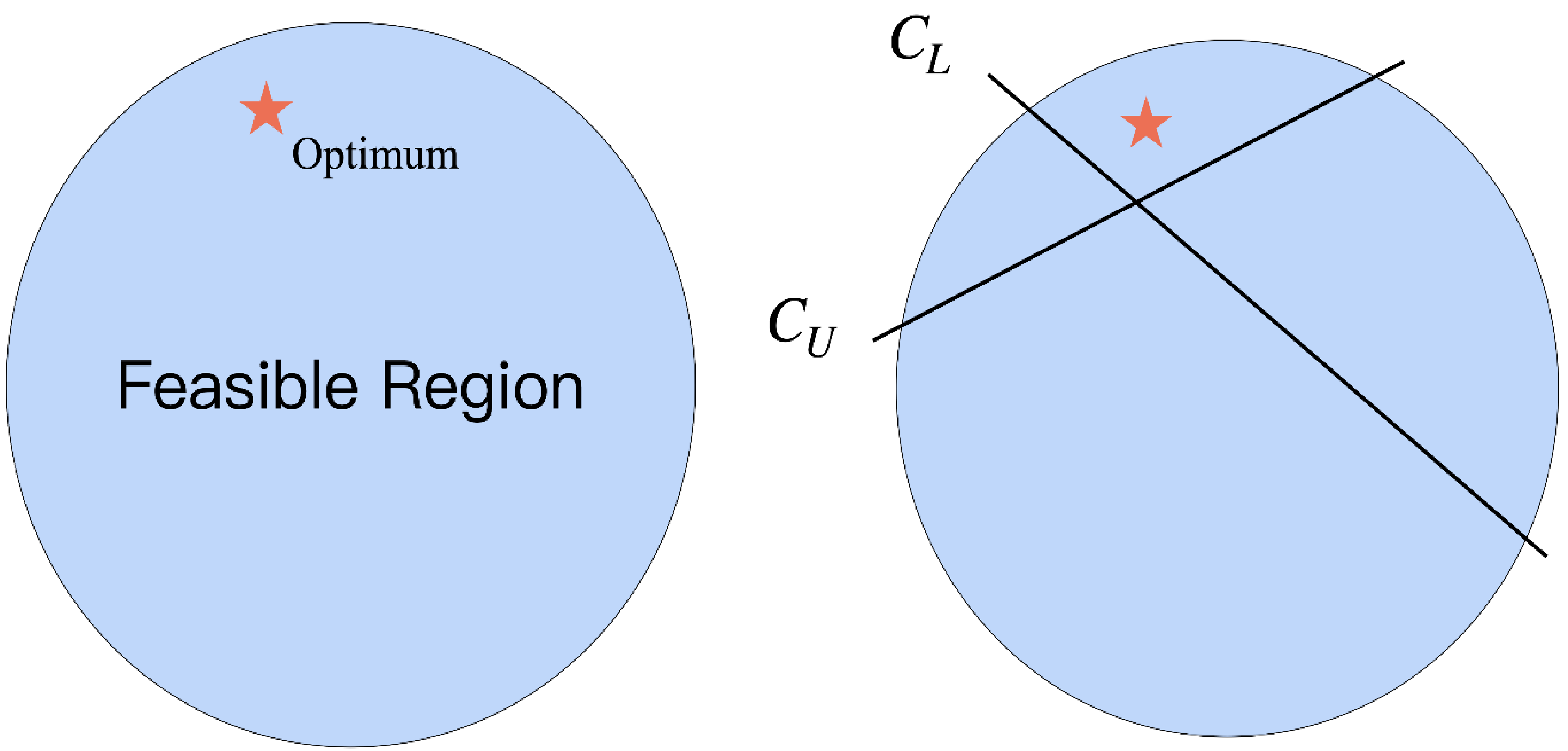
$$\sum_{i \in \mathcal{U} = \{j : \hat{y}_j(\xi) \geq 0.9\}} y_i^*(\xi) \geq \alpha \cdot |\mathcal{U}| \qquad \sum_{i \in \{j : \hat{y}_j(\xi) \leq 0.1\}} y_i^*(\xi) \leq \beta \cdot |\mathcal{L}|$$

- **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**
- **Choose α, β to increase the confidence level:**

Interpret y_j^* as some random variables with expectation $\hat{y}_j$, then justify it

Massive data are transferred into Knowledges that can be stored/reused

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



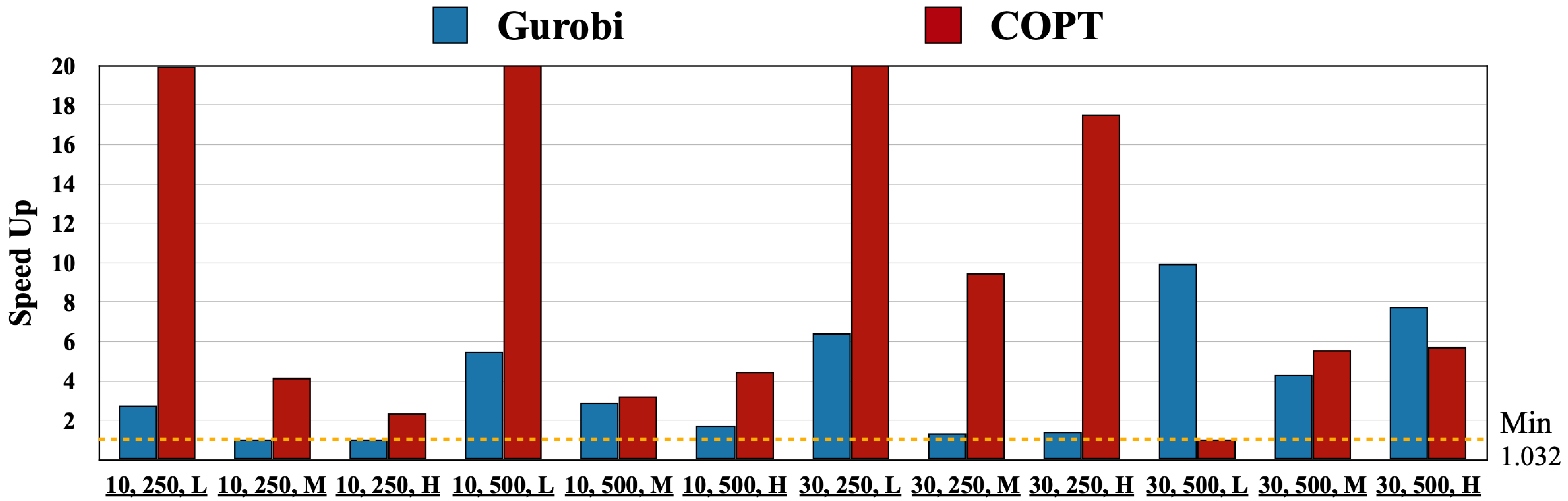
- **Overall, the two cuts (and their complement) split the whole feasible region into four regions**
- **Solving the most likelihood region of two cuts often gives a satisfying solution with confidence**
- **Branching over all four regions independently will not miss the optimal solution**

Numerical 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

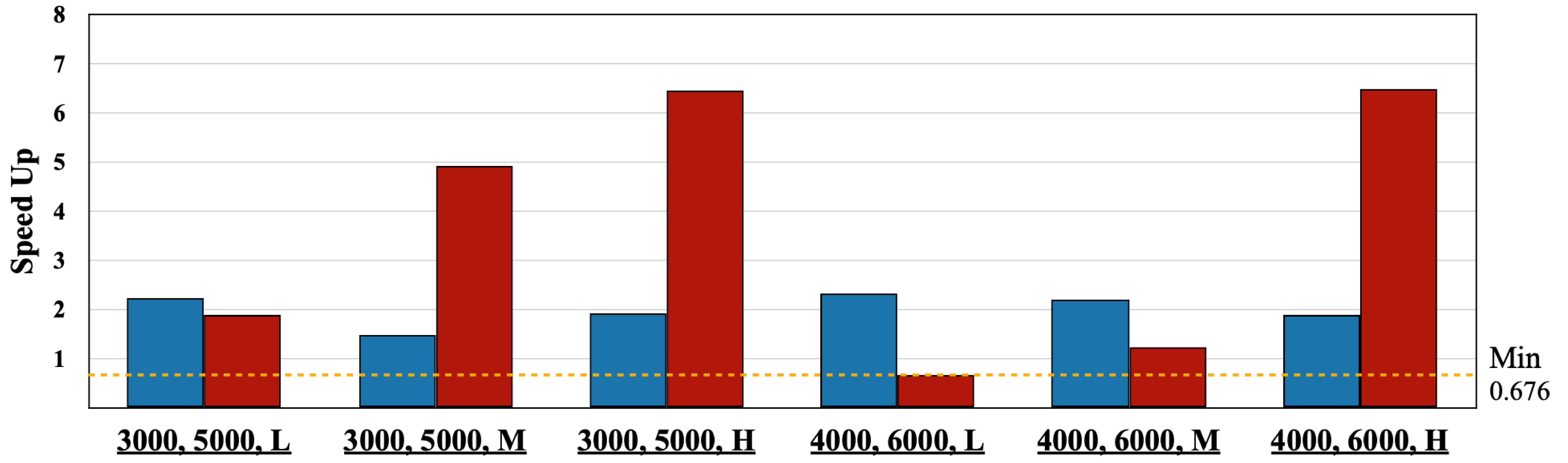
- **Measure the speedup of finding a good solution on in the region formed by two cuts**



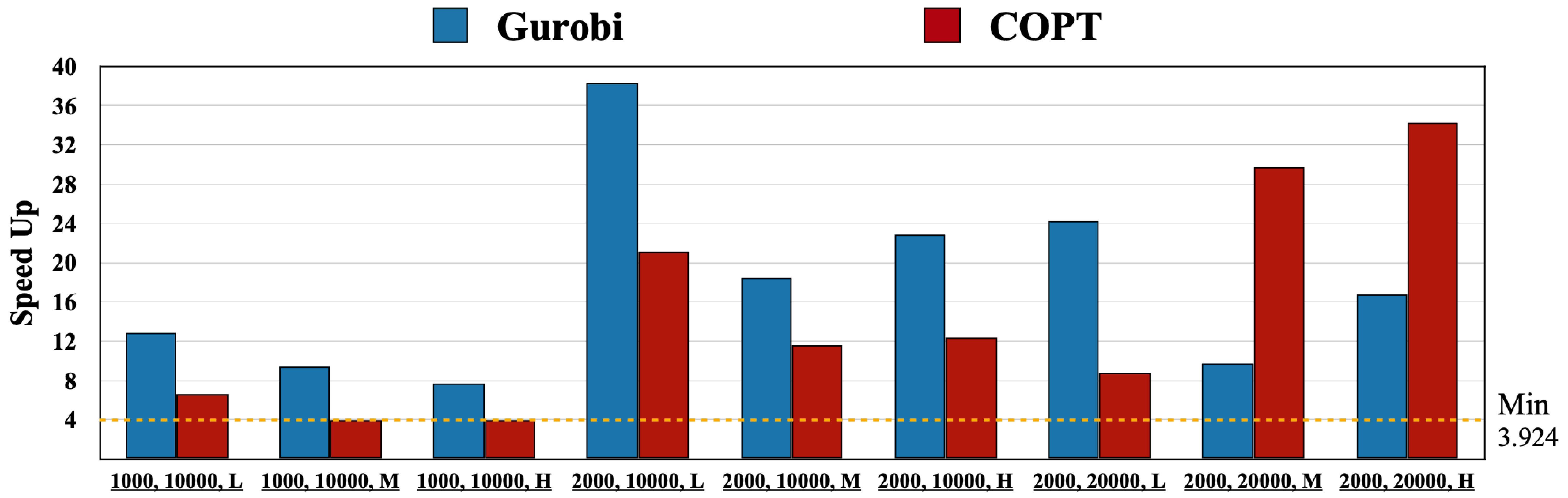
Average speedup on knapsack instances

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

Today’s Sharing

1. AI Model-Training and OR/Math Optimization
2. Online Training Algorithms and Applications
3. Pre-Trained Mixed Integer Linear Programming Solvers and Applications
4. Big-Model Training Needs Efficient Optimization Algorithms?

Accelerated Second-Order Methods for Unconstrained Optimization 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 (FOM) typically need **$O(n^2\epsilon^{-2})$** arithmetic operations
- Second-order methods (SOM) typically need **$O(n^3\epsilon^{-1.5})$** arithmetic operations
- New? Yes, HSODM and DRSOM: a single-loop method with **$O(n^2\epsilon^{-1.75})$** operations

  (Zhang et al. https://arxiv.org/abs/2211.08212)

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

- The Newton-based trust-region method minimizes the Taylor quadratic model

$$\min_{d \in \mathbb{R}^n} m_k(d) := g_k^T d + \frac{1}{2} d^T H_k d \quad \text{s.t.} \|d\| \le \Delta_k . \quad \Rightarrow \quad \psi_k(\xi_0, t; \delta) := \frac{1}{2} \begin{bmatrix} \xi_0 \\ t \end{bmatrix}^T \begin{bmatrix} H_k & g_k \\ g_k^T & -\delta \end{bmatrix} \begin{bmatrix} \xi_0 \\ t \end{bmatrix} = \frac{t^2}{2} \begin{bmatrix} \xi_0/t \\ 1 \end{bmatrix}^T \begin{bmatrix} H_k & g_k \\ g_k^T & -\delta \end{bmatrix} \begin{bmatrix} \xi_0/t \\ 1 \end{bmatrix}$$

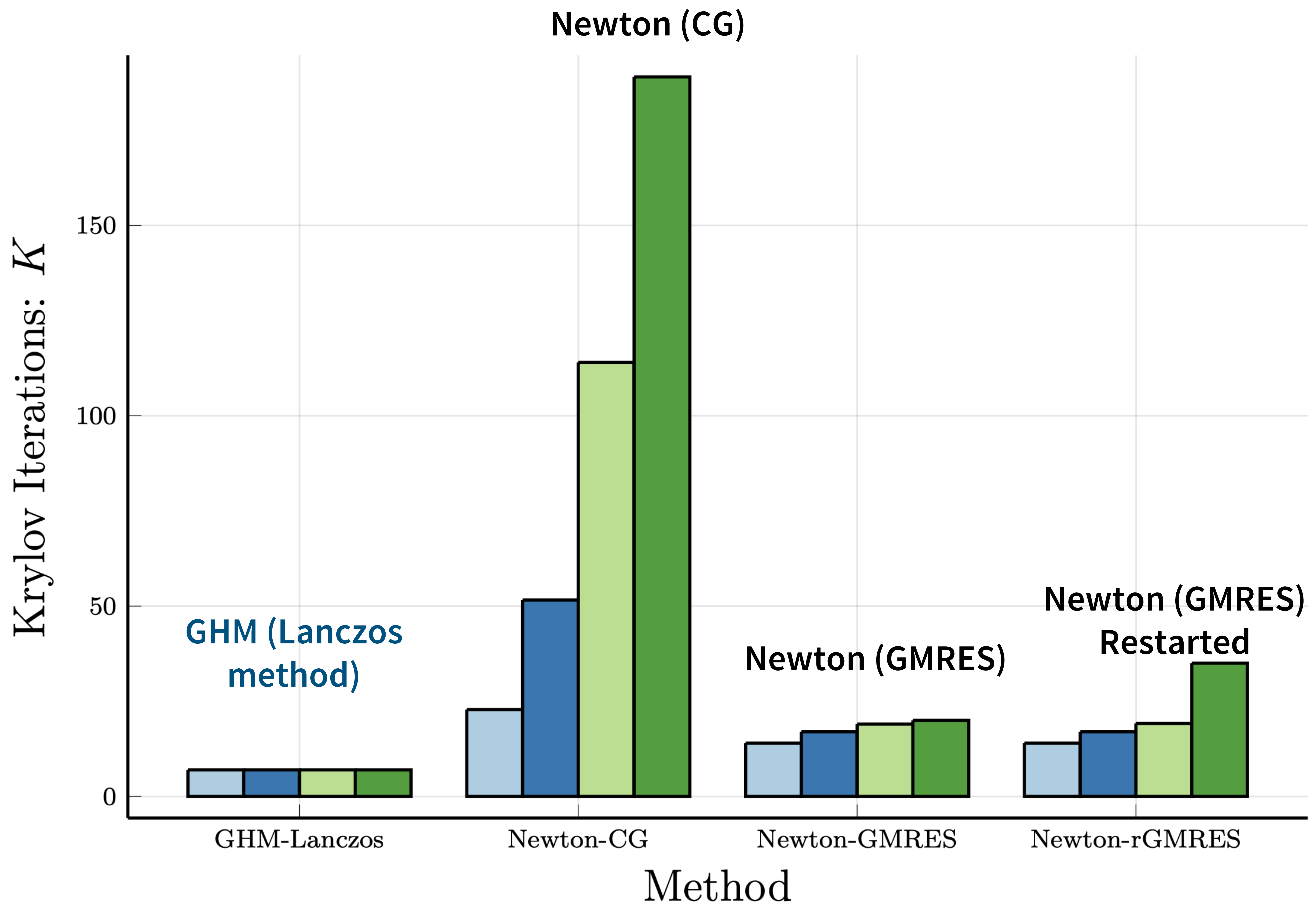
where $\Delta_k = \epsilon^{1/2}/M$ is the trust radius

- $-g_k$ 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: with a suitable $\boldsymbol{\delta}_k$ and use the most-left eigen vector as the direction to go – a single loop algorithm to solve the original problem and replace the Newton step.

$O(n^2\epsilon^{-1.75})$ vs $O(n^3)$ operations

Advantage of the Homogeneous Direction over Newton's



- **Consider Hilbert matrix**
- **Computing a Newton step by iterative methods versus Homogeneous Direction from $\boldsymbol{H} + \delta \boldsymbol{I}$**
- $\delta \downarrow$ **Condition number** $\kappa_H \uparrow$

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

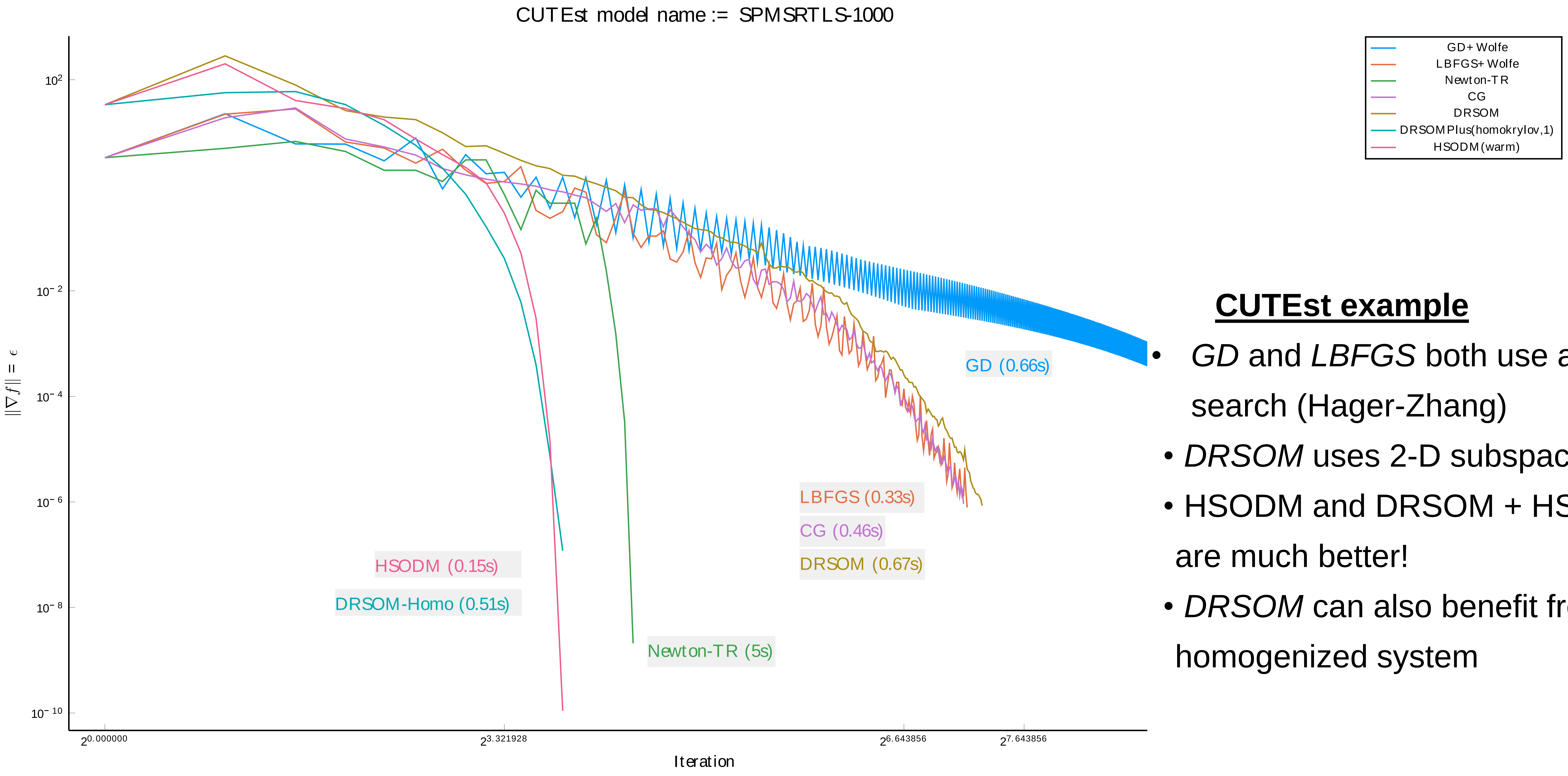
  where $g_k = \nabla f(x_k), H_k = \nabla^2 f(x^k), d_k = x_k - x_{k-1}$
- Then we solve a 2-d quadratic minimization problem to decide the two step-sizes:

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

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

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

Preliminary Results: HSODM, DRSOM and DRSOM+HSODM



CUTEst example

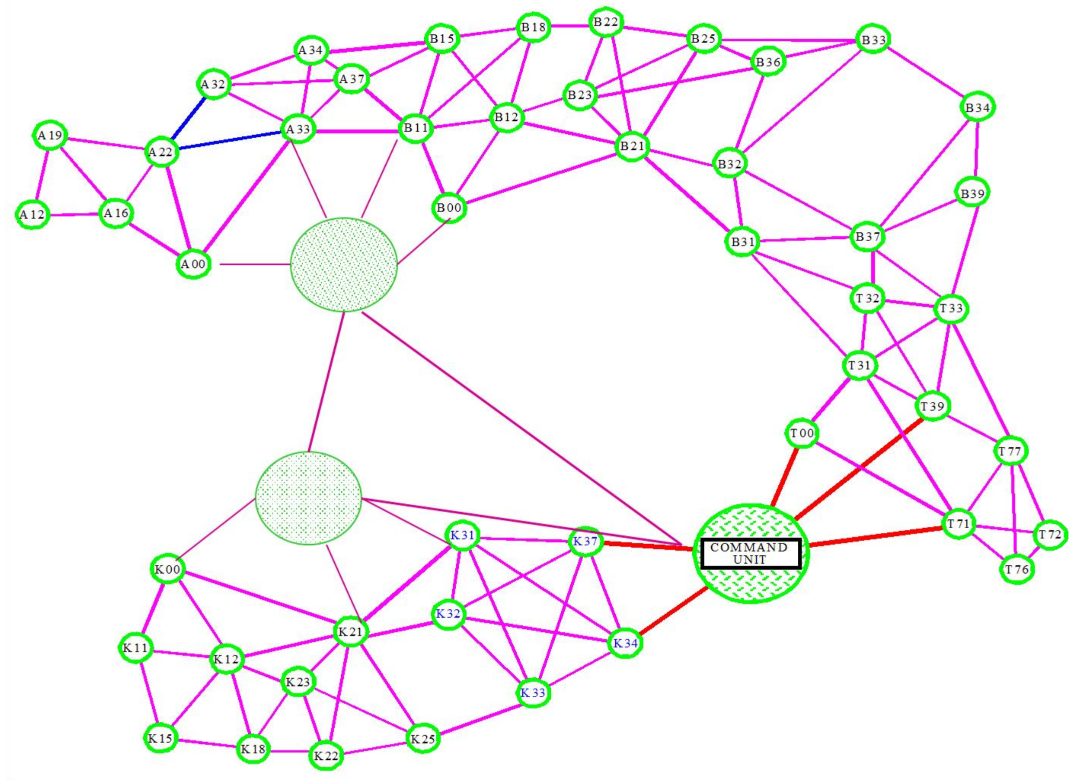
- *GD* and *LBFGS* both use a Line-search (Hager-Zhang)
- *DRSOM* uses 2-D subspace
- HSODM and DRSOM + HSODM are much better!
- *DRSOM* can also benefit from the homogenized system

Sensor Network Localization I

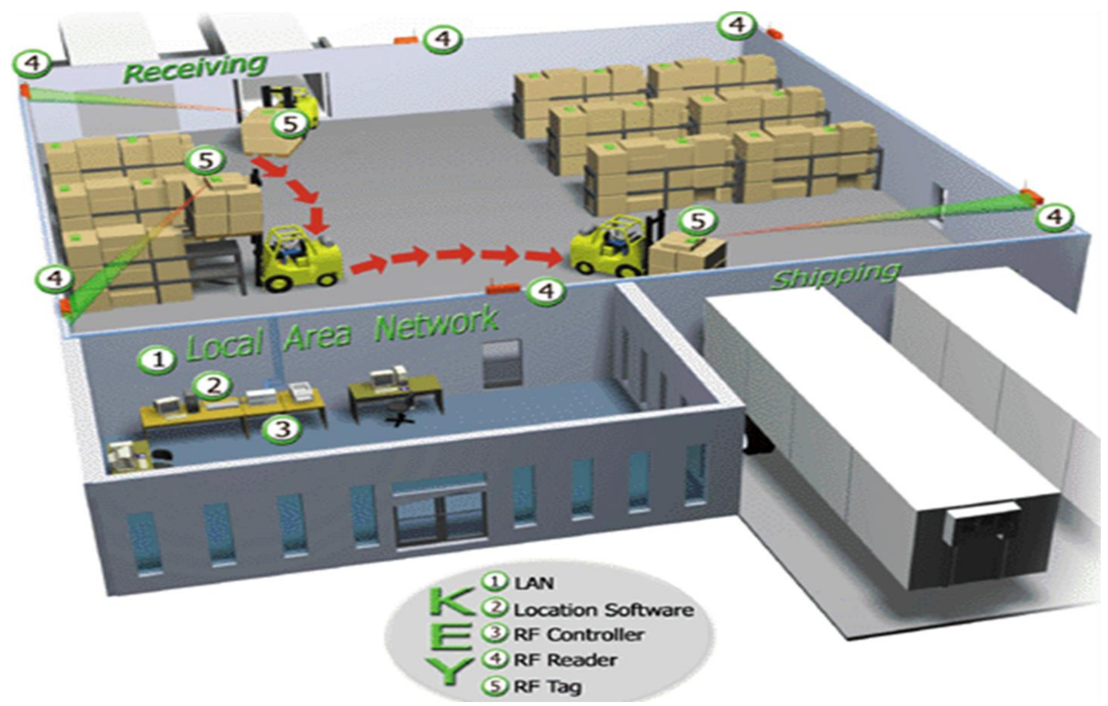
- Consider Sensor Network Location (SNL)

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

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



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

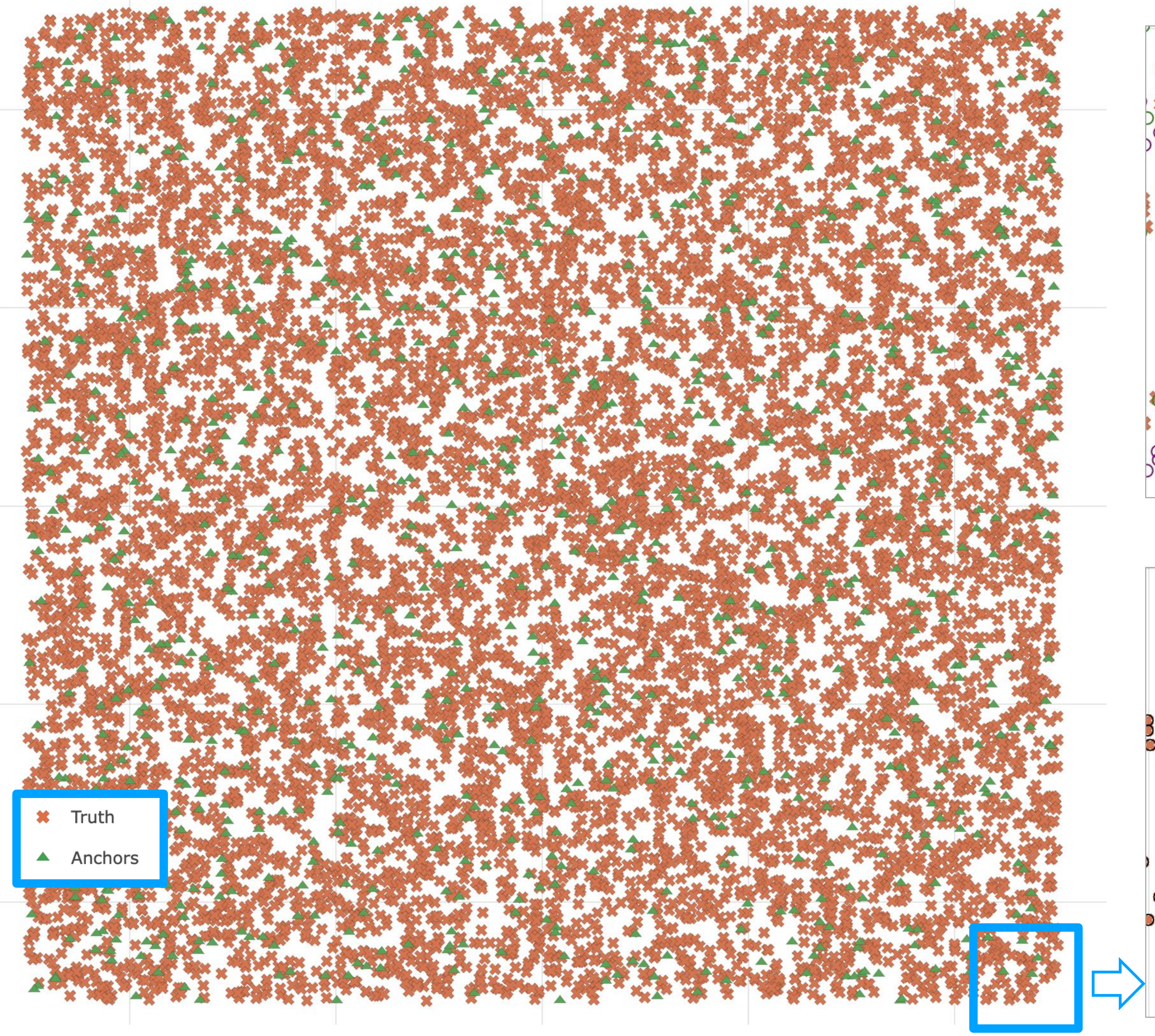


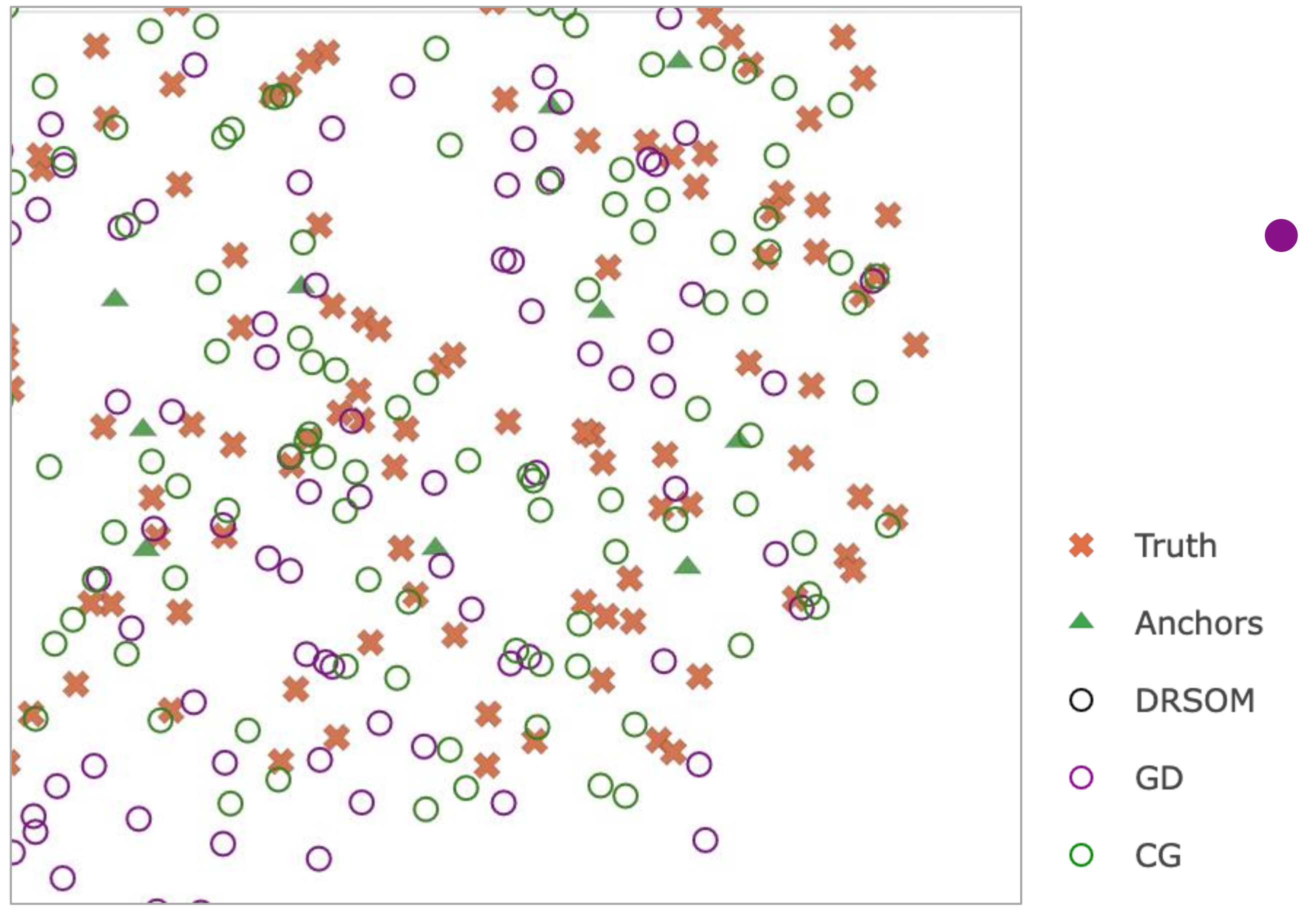
- We can solve SNL by the nonconvex nonlinear least square (NLS) problem

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

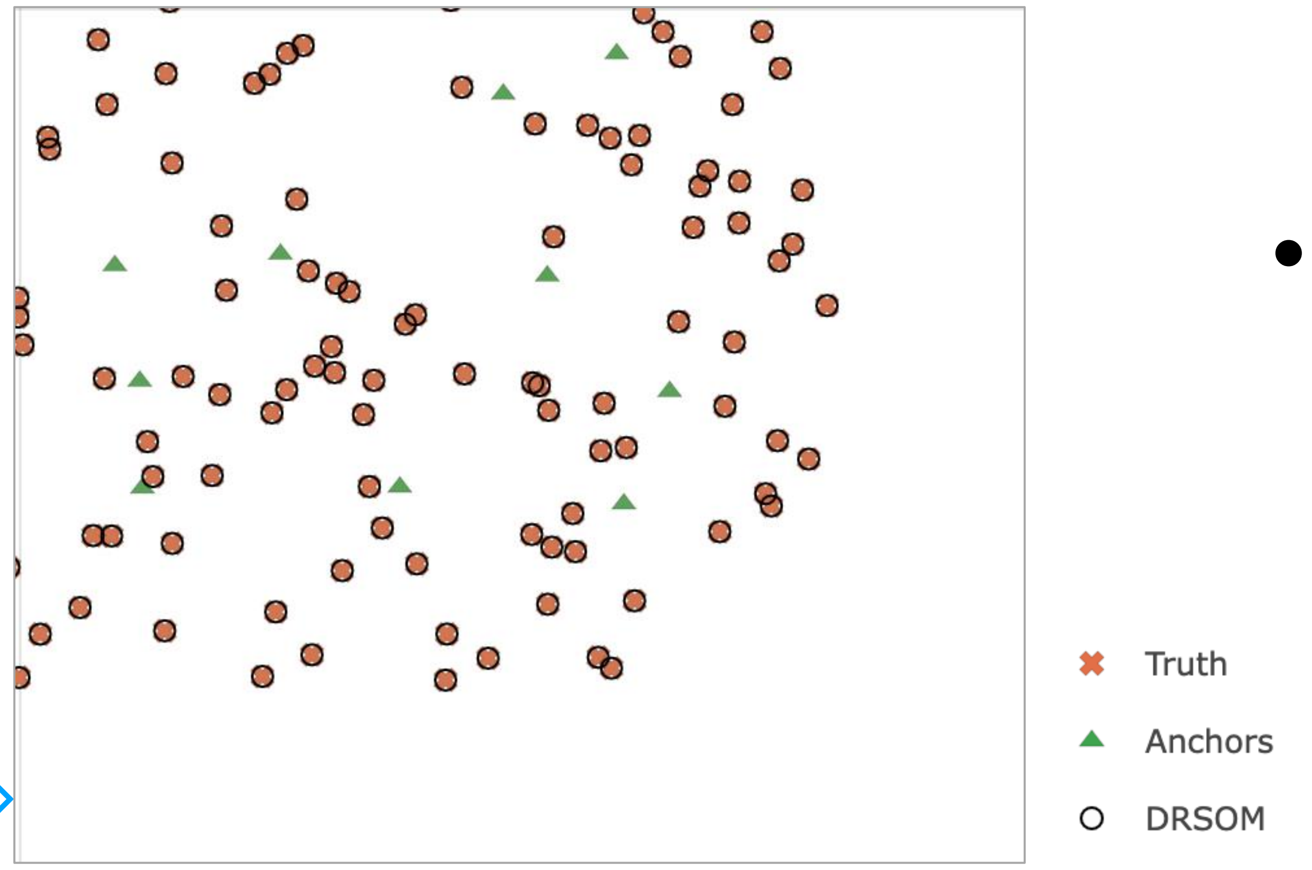
Sensor Network Location II

- 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**

Zero-Order Optimization: SOLNP+

- First proposed by Y in 1989.
- Originally implemented (SOLNP) in **Matlab**, 1989.
- **R** implementation (**Rsolnp**) by Alexios Ghalanos and Stefan Theussl, 2011.
- **New and C implementation (SOLNP+) with improvements, 2022; and addition of Randomized BCG and DRSOM for unconstrained optimization by Tan et al., 2023**
- **Github link: https://github.com/COPT-Public/SOLNP_plus**
- Use **forward difference** to evaluate the gradient.

$$[\nabla_\delta f(x)]_i = \frac{f(x+\delta e_i) - f(x)}{\delta}, \;\; e_i = [0, \; \cdots, 1, \; \cdots 0].$$

RMP: Multi-Point Random Perturbation

- Multipoint ZO Gradient Estimates

$$\hat{\nabla} f(x) := \frac{\phi(n)}{\delta b} \sum_{i=1}^{b} \left[\left(f(x + \delta u_i) - f(x) \right) u_i \right]$$

$$\mathbb{E}[\hat{\nabla} f(x)] = \nabla f_\delta(x)$$

$$f_\delta(x) = \mathbb{E}_u[f(x + \delta u)]$$

where u_i is i.i.d. **random direction**.

- Advantage: **Fewer** function queries to evaluate the gradient.

SOLNP+: Adopt Two Strategies

- With gradient estimates, SOLNP+ implements ZO version of
  - ZO-RMP (Ghadimiet al., 2013; Duchi et al., 2014), or
  - ZO-BCD, (recent research see Sun/Y 2020, Cai et al., 2021), that is, use e_i (vector with zero components except that dimension i is 1), i is randomly chosen
- DRSOM (Zhang et al., 2022) with interpolation

S. Ghadimi and G. Lan, “Stochastic first-and zeroth-order methods for nonconvex stochastic programming,” *SIAM J. Optimiz.*, vol. 23, no. 4, pp. 2341–2368, 2013. doi: 10.1137/120880811

J.C.Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, “Optimal rates for zero-order convex optimization: The power of two function evaluations,”IEEE Trans.Inf Theory, vol.61,no.5,pp.2788-2806,2015.doi: 10.1109/TIT.2015.2409256.

R. Sun and Y. Y, “Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version.” Mathematical Programming, Volume 185, 487-520, 2021.

Cai, HanQin, et al. "A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization." *International Conference on Machine Learning*. PMLR, 2021.

Zhang, Chuwen, et al. "DRSOM: A Dimension Reduced Second-Order Method and Preliminary Analyses." *arXiv preprint arXiv:2208.00208* (2022)

Experiments in Large Problems: Rosenbrock I

- Rosenbrock function is a well-known nonconvex functions in the form of

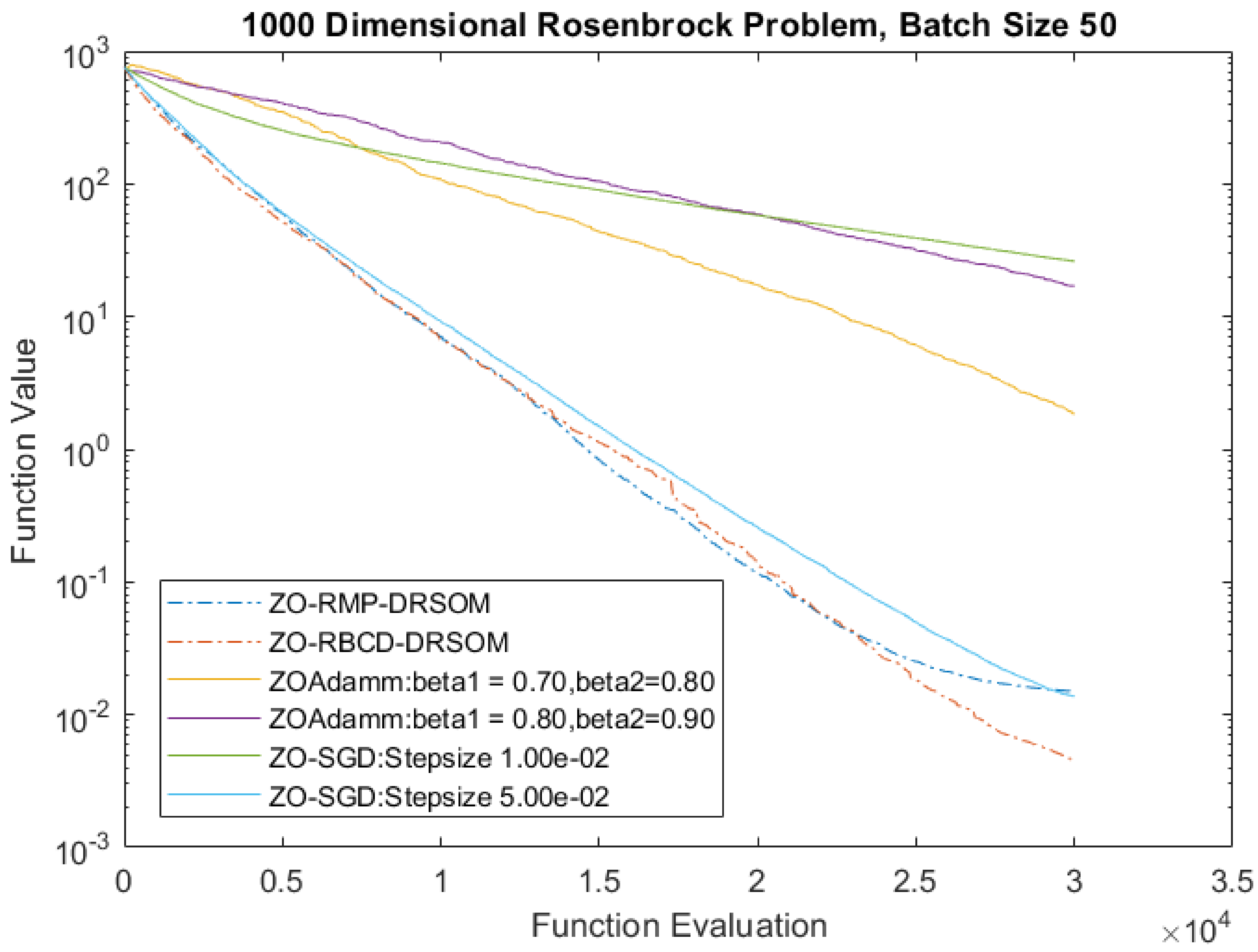
$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$$

- ZO-Adamm, ZO-SGD and ZO-DRSOM are tested in a 1200 dimensional Rosenbrock problem.
  - Batch size 50
  - Each experiment is repeated for 10 times.

Chen, Xiangyi, et al. "Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization." *Advances in neural information processing systems* 32 (2019).

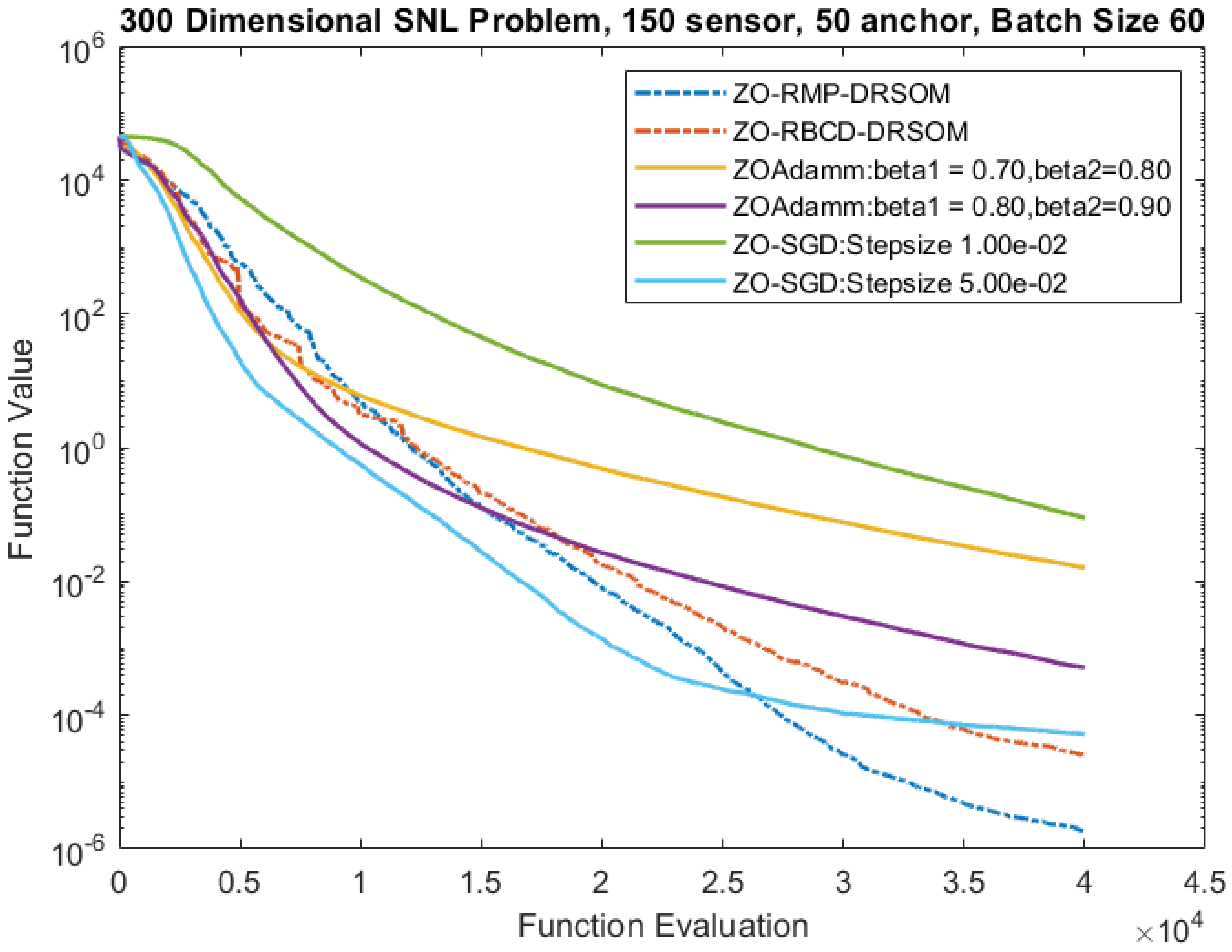
Experiments in Large Problems: Rosenbrock II

- ZO-RMP-DRSOM, ZO-RBCD-DRSOM and ZO-SGD decrease most smoothly. However, inappropriate parameters lead to worse performance of ZO-SGD and ZO-ADAMM.



Experiments in Large-Sacle Problems: SNL I

- ZO-Adam, ZO-SGD, ZO-RBCD-DRSOM and ZO-RMP-DRSOM are tested in a 150-sensor SNL problem.
  - Batch size 60
  - Each experiment is repeated for 10 times.



Overall Takeaways

Know the pros and cons of OR and AI models and use them intelligently

Most OR optimization models/algorithms can be readily adopted for online training

Pre-training greatly improves Mixed Integer LP solvers that benefit real economy

Better to integrate ZOM, FOM and SOM for Nonlinear and/or Black-Box Optimization!

- THANK YOU