Optimal Auctions through Deep Learning

Paul Dtting, Zhe Feng, Harikrishna Narasimham, David C. Parkes, Sai S.Ravindranath

International Conference on Machine Learning(ICML) 2019

Paper link: http://proceedings.mlr.press/v97/duetting19a/duetting19a.pdf

Full version: https://arxiv.org/pdf/1706.03459

Repository: https://github.com/saisrivatsan/deep-opt-auctions

June 7, 2022

incentive compatible (a.k.a. truthfulness or strategy proof)

✓ Dominant strategy incentive compatibility(DSIC)

When bidders get maximum utility by revealing their true information irrespective of the bids from other bidders.

e.g. Second Price Auction

$$b_{-i} = (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$$

(CE) First Price Auction

$$u_i(v_i, (v_i, b_{-i})) \geq u_i(v_i, (b_i, b_{-i}))$$

✓ Bayesian-Nash incentive compatibility(BNIC):

It is a weaker form compared to the former. In this, a bidder reveals its private information when other bidders also reveal their private information.

https://en.wikipedia.org/wiki/First-price_sealed-bid_auction

Background

Designing an incentive compatible auction that maximizes expected revenue is an intricate task!

The single-item was resolved[Myerson's auction 1981]

However, After 30-40 years ...

simple multi-bidder, multi-item settings remains unresolved

[1] Myerson R B. Optimal auction design[J]. Mathematics of operations research, 1981, 6(1): 58-73.

Main Contribution

- ✓ provide the first, general purpose, end-to-end approach for solving the multi-item auction design problem
- ✓ model an auction as a multi-layer neural network, frame optimal auction design as a constrained learning problem(incentive compatibility)
- ✓ **The goal is to learn an auction that maximizes revenue.**
- ✓ Recreate state of the art analytical results of optimal auctions
- ✓ Discover new auctions for settings where optimal solution is unknown

Notation

Bidders: $N = \{1, \ldots, n\}$

$v_{-i} = (v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$

items: $M = \{1, \ldots, m\}$

真实估价

valuation function: $v_i(S) : 2^M \longrightarrow \mathbb{R}_{\geq 0}, S \subseteq M$

profile of valuations: $v = (v_1, \ldots, v_n)$ $v_i \in V_i$ $v_i \sim F_i$

Bidder i Allocation rule: $g_i : V \longrightarrow 2^M$ $v \sim F$

Bidder i Payment rule: $p_i : V \longrightarrow \mathbb{R}_{\geq 0}$

实际出价

profile of bids: $b = (b_1, \ldots, b_n)$

对Bidder i分配到的items进行估值 $b_i \in V_i$

Bidder i utility: $u_i(v_i, b) = v_i(g_i(b)) - p_i(b)$

Bidder i需要支付的金额

The Problem

- A seller with a set of m items
- A set of n bidders with independent private valuations, $v_i: 2^M \to R_{\geq 0}, v_i \sim F_i$
- $F = (F_1, \cdots, F_n)$ is known.
- Design auction (g^w, p^w) that maximizes expected revenue, s.t. strategy-proofness.
  - g^w is parametrized allocation rule
  - p^w is parametrized payment rule

$$max\ \mathbf{E}[\textstyle\sum_{i \in N} p_i^{\omega}(v)]$$

$$\Updownarrow$$

$$min\ -\mathbf{E}[\textstyle\sum_{i \in N} p_i^{\omega}(v)]$$

Approach

Our Approach: RegretNet

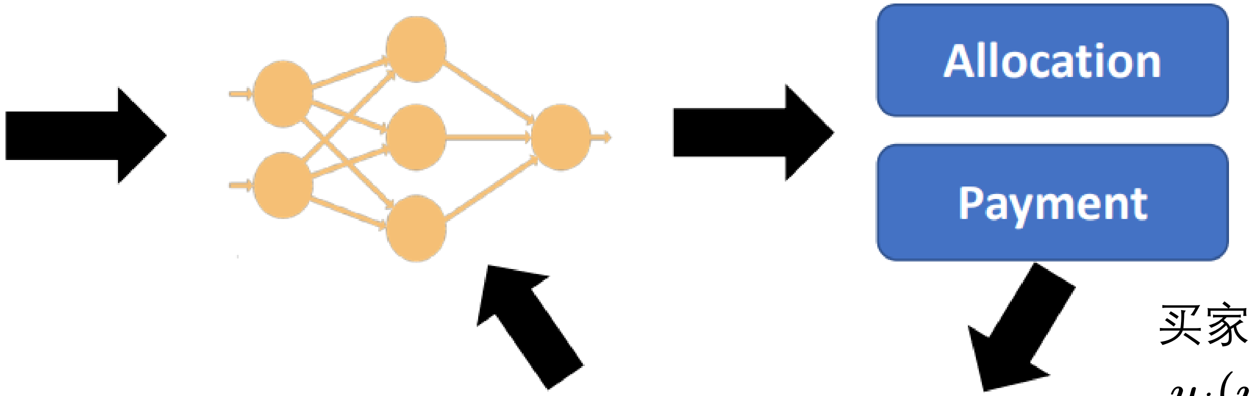
一组拍卖： $(g^{\omega}, p^{w}) \in \mathcal{M}, \omega \in \mathbb{R}^d, d \in \mathbb{N}$

L个样本： $S = (v^{(1)}, v^{(2)}, \ldots, v^{(L)})$

模型训练参数： $\omega \in \mathbb{R}^d$

$$\begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ v^{(L)} \end{bmatrix}$$

Sample from F



Training

Tune weights to maximize revenue s.t. ***incentive constraints* (main challenge)**

买家的效用函数

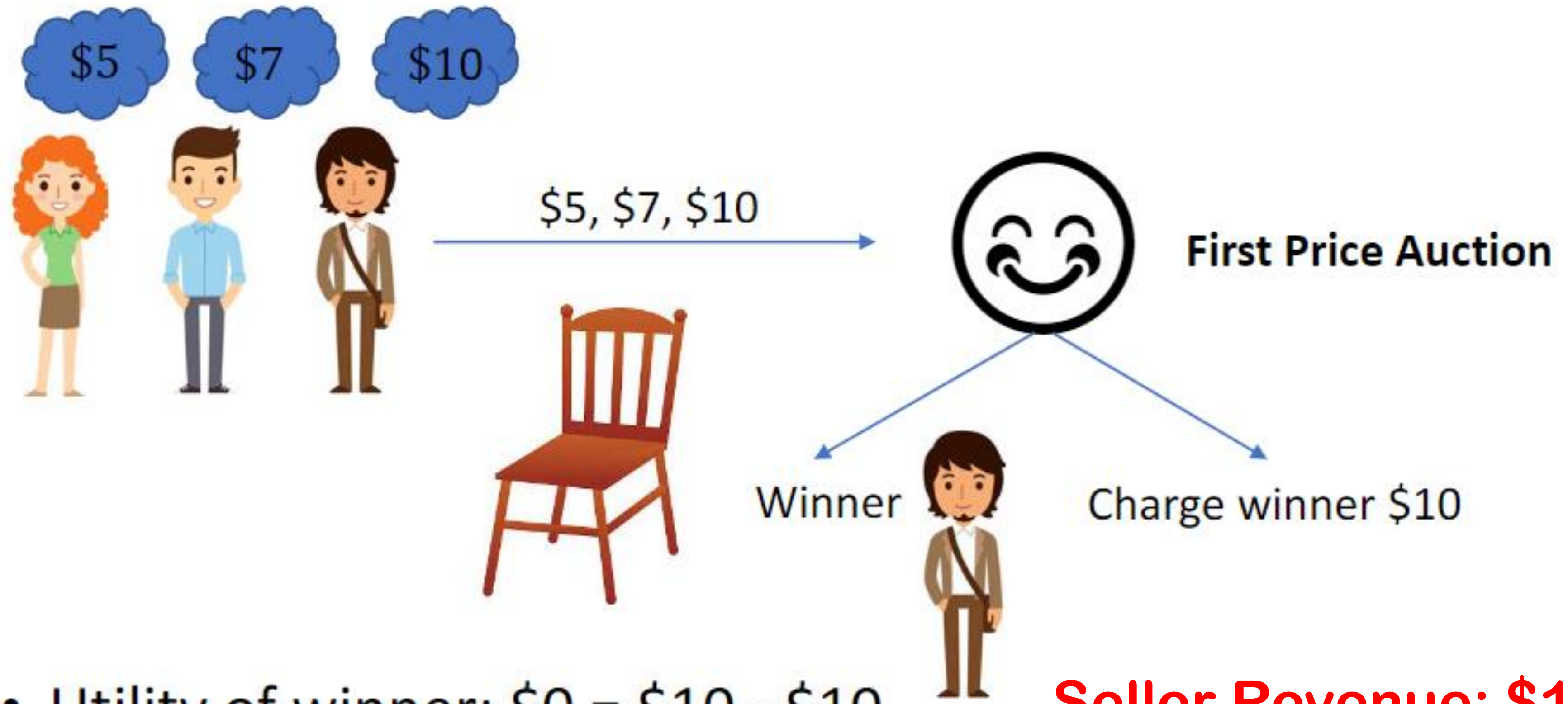
$$u_i(v_i', v_{-i}) = v_i \cdot g_i(v_i', v_{-i}) - p_i(v_i', v_{-i})$$

true value   bid   Other' bids

For example: First Price Auction

Auction 101

$V_i = \{8, 9, 10\}, v_i = 10$

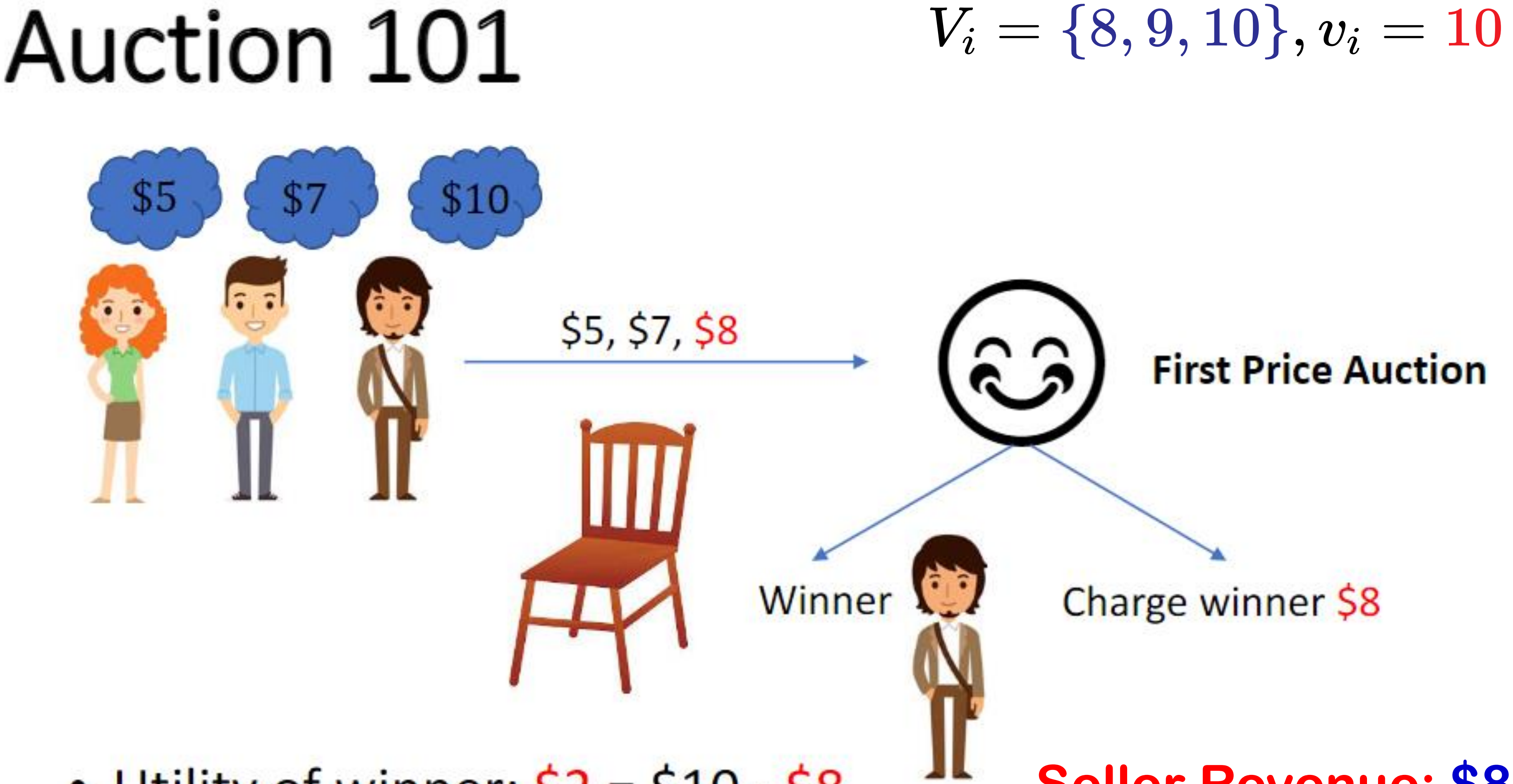


- Utility of winner: $0 = $10 - $10

Seller Revenue: $10

For example: First Price Auction

$V_i = \{8, 9, 10\}, v_i = 10$



- Utility of winner: $2 = $10 - $8

Seller Revenue: $8

Formulation as a learning problem

- expected utility for bidder i with bid v_i'

$$U(g,p,v_i) = \mathbb{E}_{v_{-i}}[v_i \cdot g_i(v_i', v_{-i}) - p_i(v_i', v_{-i})]$$

- What is Regret (expected ex post regret)?

$$rgt_i(w) = \mathbf{E}\left[\max_{v_i' \in V_i} u_i^w(v_i; (v_i', v_{-i})) - u_i^w(v_i; (v_i, v_{-i}))\right] \geq 0$$

the ex post regret for a bidder is the maximum increase in her utility

- re-formulate the learning problem as minimizing **the expected loss**

$$\min_{w \in \mathbb{R}^d} \mathbf{E}_{v \sim F}\left[-\sum_{i \in N} p_i^w(v)\right] \quad \text{s.t.} \quad rgt_i(w) = 0, \ \forall i \in N.$$

Formulation as a learning problem

- a sample of bidder valuation profiles

$$S = (v^{(1)}, v^{(2)}, \dots, v^{(L)})$$

- estimate the empirical ex post regret for bidder i (事后遗憾)

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L} \max_{v_i' \in V_i} u_i^w(v_i^{(\ell)}; (v_i', v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)})$$

- minimize **the empirical loss** subject to the empirical regret being zero for all bidders

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & -\tfrac{1}{L}\textstyle\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)}) \\ \text{s.t.} \quad & \widehat{rgt}_i(w) = 0, \ \forall i \in N. \end{aligned}$$

Generalization Error bound

- Generalization bound

We bound the gap between the expected regret and the empirical regret in terms of the number of sampled valuations profiles

- solving for (2) with a large sample yields an auction with near-optimal expected revenue and close-to-zero expected regret

$$\begin{cases} \mathbf{E}_{v\sim F}\left[-\sum_{i\in N} p_i^w(v)\right] \leq -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w\left(v^{(\ell)}\right) + 2n\Delta_L + Cn\sqrt{\frac{\log(1/\delta)}{L}} \\ \frac{1}{n}\sum_{i=1}^{n} rgt_i(w) \leq \frac{1}{n}\sum_{i=1}^{n}\widehat{rgt}_i(w) + 2\Delta_L + C'\sqrt{\frac{\log(1/\delta)}{L}} \end{cases}$$

$$\Delta_L = \inf_{\epsilon>0}\left\{\frac{\epsilon}{n} + 2\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{M},\epsilon/2))}{L}}\right\}$$

$$L \to \infty, \Delta_L \to 0$$

Bidders with additive valuations $v_i(S) = \sum_j v_i(j)$

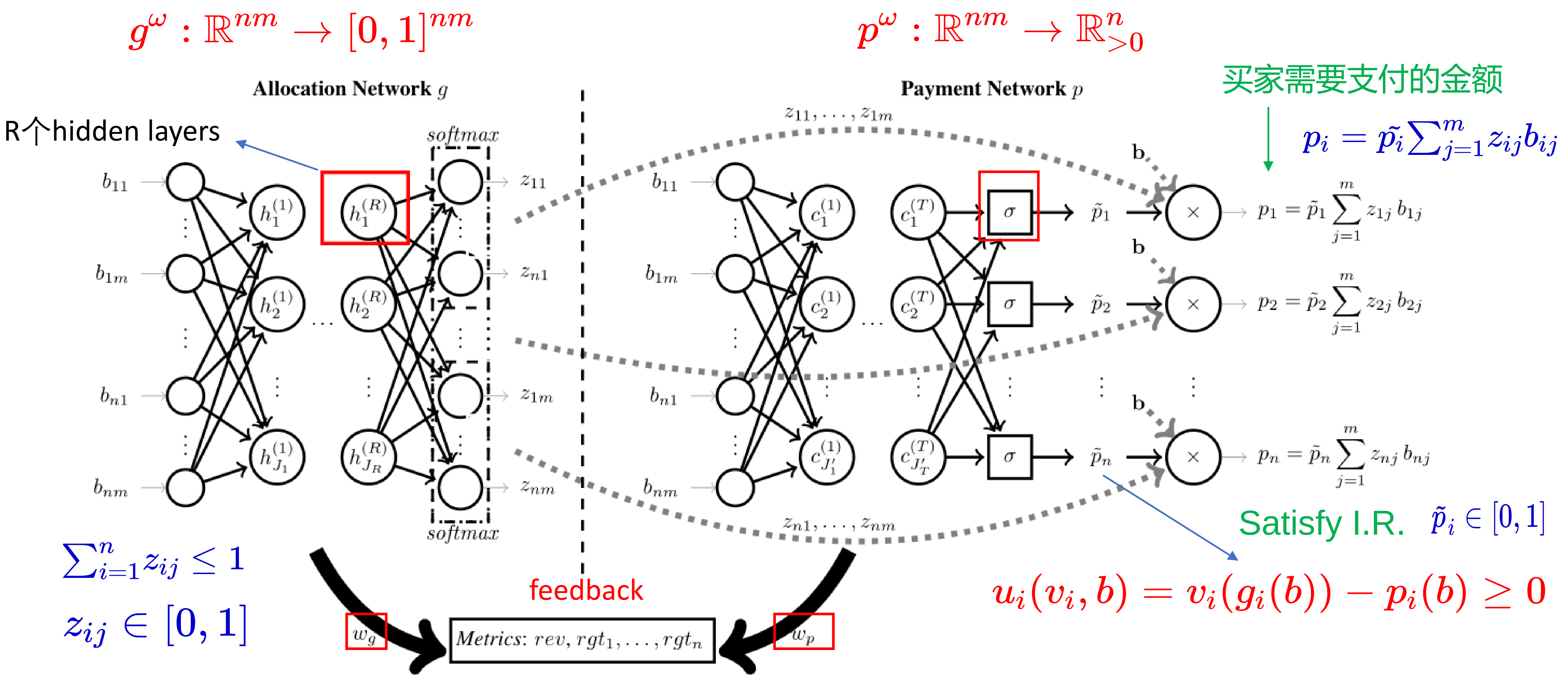
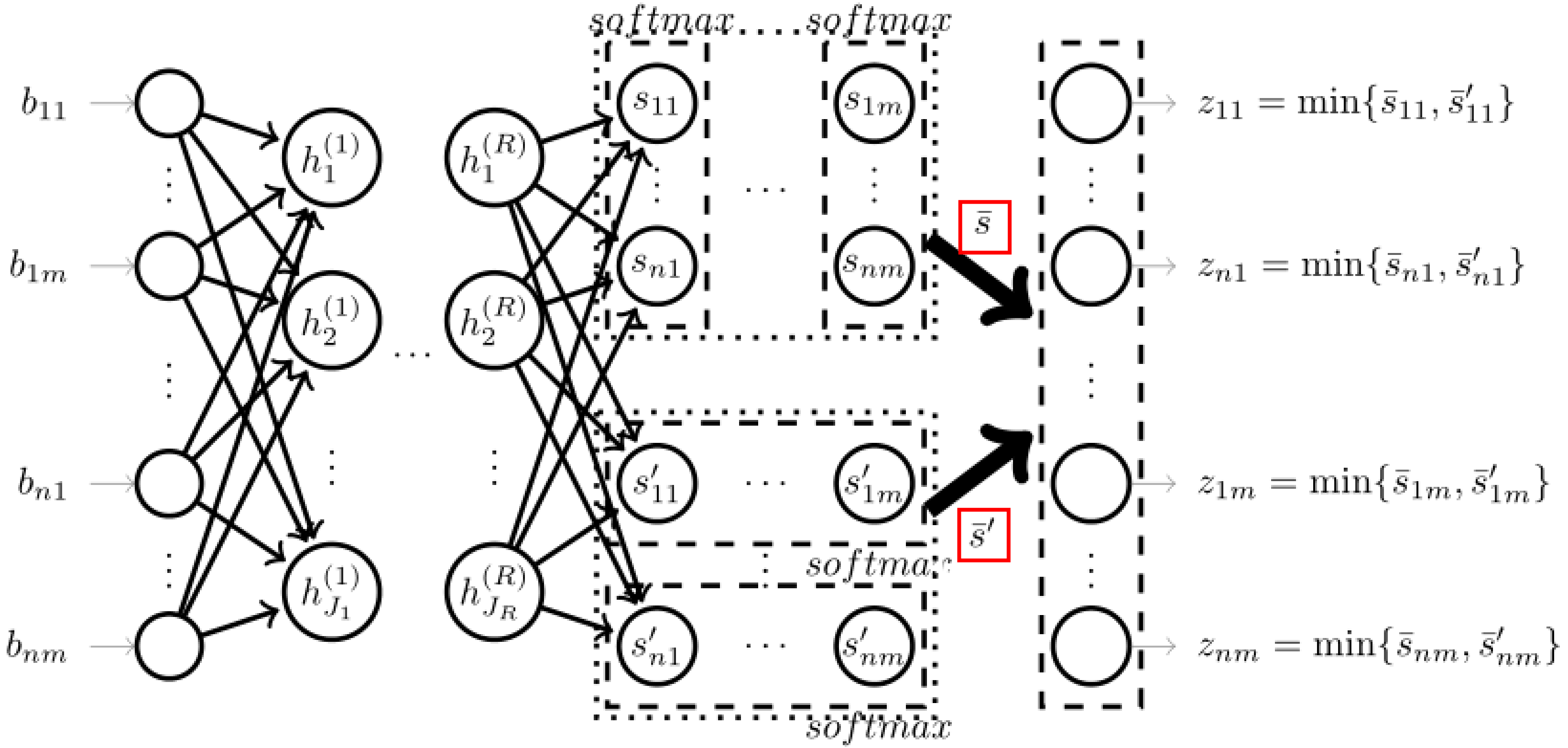


Figure 1: The allocation and payment networks for a setting with n additive bidders and m items. The inputs are bids from bidders for each item. The rev and each rgt_i are defined as a function of the parameters of the allocation and payment networks $\boxed{w = (w_g, w_p)}$.

Bidders with unit-demand valuations $v_i(S) = max_{j \in S} v_i(j)$



$\boxed{g^\omega : \mathbb{R}^{nm} \to [0,1]^{nm}}$

$$\begin{cases} \sum_{i=1}^{n} z_{ij} \leq 1 \\ \sum_{j=1}^{m} z_{ij} \leq 1 \end{cases}$$

Figure 2: The allocation network for settings with (a) n *unit-demand* bidders and m items

Lemma 1. *$\varphi^{DS}(s, s')$ is doubly stochastic $\forall\, s, s' \in \mathbb{R}^{nm}$. For any doubly stochastic allocation $z \in [0,1]^{nm}$, $\exists\, s, s' \in \mathbb{R}^{nm}$, for which $z = \varphi^{DS}(s, s')$.*

$$z_{ij} = \varphi_{ij}^{DS}(s, s') = \min\left\{ \boxed{\frac{e^{s_{ij}}}{\sum_{k=1}^{n+1} e^{s_{kj}}}}, \boxed{\frac{e^{s'_{ij}}}{\sum_{k=1}^{m+1} e^{s'_{jk}}}} \right\}$$

$\bar{s}_{ij}$ $\quad$ $\bar{s}'_{ij}$

Bidders with additive or unit-demand valuations

- Bidders with additive valuations $\forall i \in N, \forall j \in M$

$\sum_{i=1}^{n} z_{ij} \leq 1$ → 每个item最多只能分配给一个人

分配策略1：从资源分配效率和卖家获取的期望收入出发，用户可以分配到多个item

- Bidders with unit-demand valuations

$$\begin{cases} \sum_{i=1}^{n} z_{ij} \leq 1 \\ \sum_{j=1}^{m} z_{ij} \leq 1 \end{cases}$$

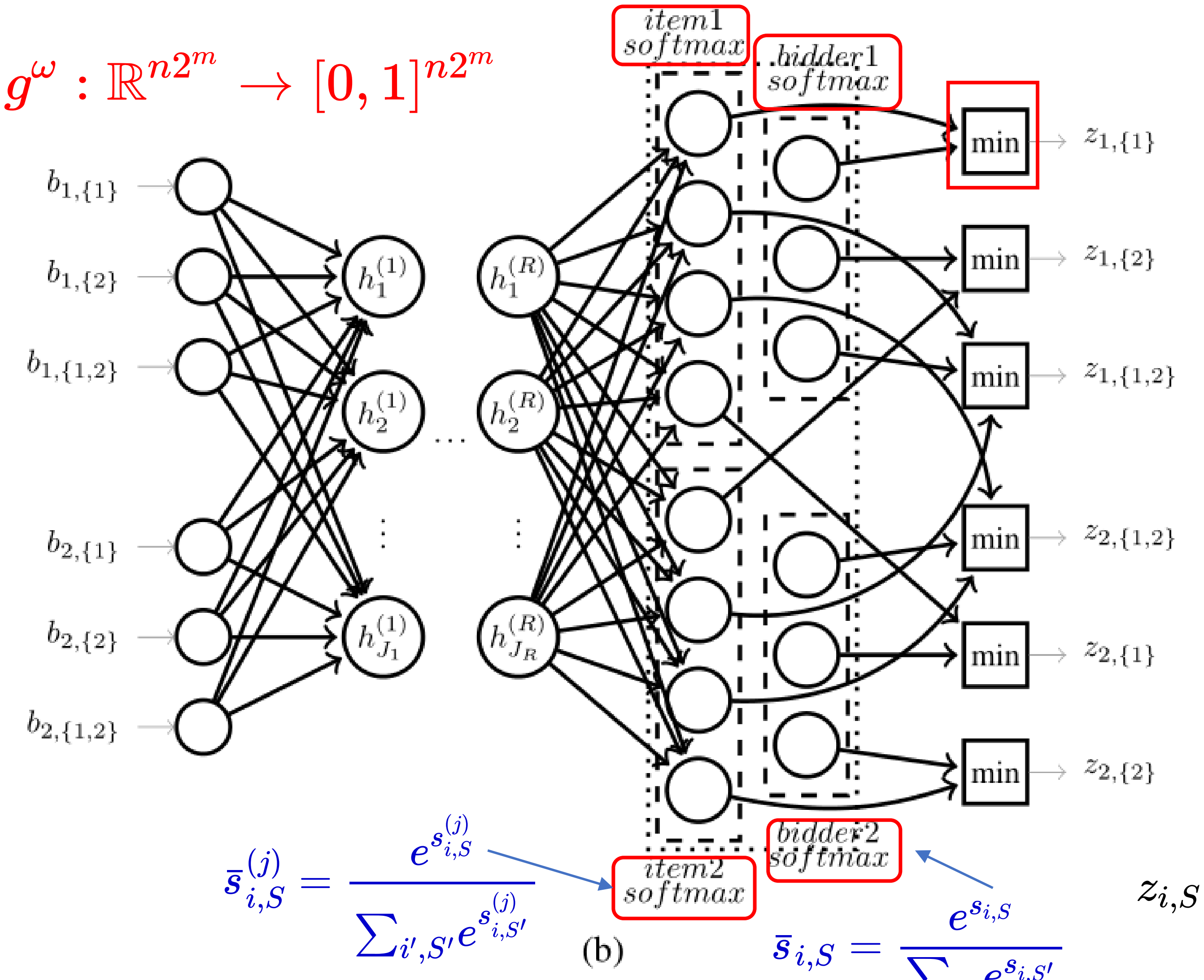
$\sum_{i=1}^{n} z_{ij} \leq 1$ → 每个item的分配概率之和不超过1
每个item最多分配给一个bidder

$\sum_{j=1}^{m} z_{ij} \leq 1$ → 每个bidder最多分配到一个item

分配策略2：考虑到公平分配的原则，从而需要保证一个bidder最多分配到一个item。

Bidders with combinatorial valuations

$g^{\omega}:\mathbb{R}^{n2^m}\rightarrow[0,1]^{n2^m}$



$\bar{s}^{(j)}_{i,S}=\frac{e^{s^{(j)}_{i,S}}}{\sum_{i',S'}e^{s^{(j)}_{i,S'}}}$

$\bar{s}_{i,S}=\frac{e^{s_{i,S}}}{\sum_{S'}e^{s_{i,S'}}}$

Fig (b): 2 combinatorial bidders and 2 items

- In this case, each bidder i reports a bid $b_{i,S}$ for every bundle of items S ⊆ M
- combinatorial feasible(CF)

所有bidders的bundle S中存在item j的概率不超过1

$$\sum_{i\in N}\sum_{S\subseteq M:j\in S}z_{i,S}\leq 1,\forall j\in M$$

$$\sum_{S\subseteq M}z_{i,S}\leq 1,\forall i\in N$$

一个bidder所有bundle分配的概率之和不超过1

- Allocation rule and Payment rule

$$z_{i,S}=\varphi^{CF}_{i,S}\left(s,s^{(1)},\dots,s^{(m)}\right)=\min\left\{\bar{s}_{i,S},\bar{s}^{(j)}_{i,S}:j\in S\right\}$$

$$p_i=\tilde{p}_i\sum_{S\subseteq M}z_{i,S}b_{ij}$$

Covering number bounds

Theorem 2. *For RegretNet with R hidden layers, K nodes per hidden layer, d_a parameters in the allocation network, d_p parameters in the payment network, and the vector of all model parameters $\|w\|_1 \leq W$, the following are the bounds on the term Δ_L for different bidder valuation types:*

(a) additive valuations:

$$\Delta_L \leq O\left(\sqrt{R(d_a + d_p)\log(LW \max\{K, mn\})/L}\right),$$

(b) unit-demand valuations:

$$\Delta_L \leq O\left(\sqrt{R(d_a + d_p)\log(LW \max\{K, mn\})/L}\right),$$

(c) combinatorial valuations:

$$\Delta_L \leq O\left(\sqrt{R(d_a + d_p)\log(LW \max\{K, n\,2^m\})/L}\right).$$

- Gradient clipping

$$clip(g_i, L) = g_i \cdot min(1, \frac{L}{\|g_i\|_1})$$

$$L \rightarrow \infty, \triangle L \rightarrow 0$$

$$\Delta_L = O(\sqrt{\frac{log(L)}{L}})$$

RegretNet Training

Algorithm 1 RegretNet Training

Input: Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size B
Parameters: $\forall t, \rho_t > 0, \gamma > 0, \eta > 0, R \in \mathbb{N}, K \in \mathbb{N}$
Initialize: $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n$
for $t = 0$ **to** T **do**
  Receive minibatch $\mathcal{S}_t = \{u^{(1)}, \ldots, u^{(B)}\}$
  Initialize misreports $v_i'^{(\ell)} \in V_i, \forall \ell \in [B],\ i \in N$
  for $r = 0$ **to** R **do**
    $\forall \ell \in [B],\ i \in N$:
      $v_i'^{(\ell)} \leftarrow v_i'^{(\ell)} + \gamma \nabla_{v_i'} u_i^w\big(v_i^{(\ell)}; (v_i'^{(\ell)}, v_{-i}^{(\ell)})\big)$
  end for
  Compute regret gradient: $\forall \ell \in [B], i \in N$:
    $g_{\ell,i}^t =$
    $\nabla_w \left[u_i^w\big(v_i^{(\ell)}; (v_i'^{(\ell)}, v_{-i}^{(\ell)})\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big)\right]\Big|_{w=w^t}$
  Compute Lagrangian gradient using (5) and update w^t:
    $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
  Update Lagrange multipliers once in Q iterations:
    if t is a multiple of Q
      $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t\, \widetilde{rgt}_i(w^{t+1}), \quad \forall i \in N$
    else
      $\lambda^{t+1} \leftarrow \lambda^t$
end for

- the constrained training problem

$$\min_{w \in \mathbb{R}^d} \quad -\frac{1}{L} \sum_{\ell=1}^{L} \sum_{i=1}^{n} p_i^w(v^{(\ell)})$$
$$\text{s.t.} \quad \widehat{rgt}_i(w) = 0, \quad \forall i \in N. \qquad (2)$$

- the augmented Lagrangian method(solve(2))

$$\mathcal{C}_\rho(w; \lambda) = -\frac{1}{L} \sum_{\ell=1}^{L} \sum_{i \in N} p_i^{\boxed{w}}\left(v^{(\ell)}\right) + \sum_{i \in N} \boxed{\lambda_i} \widehat{rgt}_i(w) + \frac{\rho}{2} \left(\sum_{i \in N} \widehat{\mathrm{rgt}}_i(w)\right)^2$$

- update model parameters and Lagrange multipliers

$$\begin{cases} \omega^{new} = \underset{w}{\operatorname{argmin}} \left[\mathcal{L}(g^w, p^w) + \sum_i \lambda_i^{t-1} \cdot rgt_i^w + \frac{\rho}{2} \sum_i (rgt_i^w)^2\right] \\ \lambda_i^{new} = \lambda_i^{old} + \rho\, \widehat{rgt}_i(\omega^{new}), \forall i \in N \end{cases}$$

RegretNet Training

Algorithm 1 RegretNet Training

Input: Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size B
Parameters: $\forall t, \rho_t > 0, \gamma > 0, \eta > 0, R \in \mathbb{N}, K \in \mathbb{N}$
Initialize: $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n$
for $t = 0$ **to** T **do**
  Receive minibatch $\mathcal{S}_t = \{u^{(1)}, \ldots, u^{(B)}\}$
  Initialize misreports $v_i'^{(\ell)} \in V_i, \forall \ell \in [B],\ i \in N$
  for $r = 0$ **to** R **do**
    $\forall \ell \in [B],\ i \in N$:
    $\boxed{v_i'^{(\ell)}} \leftarrow v_i'^{(\ell)} + \gamma \nabla_{v_i'} u_i^w(v_i^{(\ell)}; (v_i'^{(\ell)}, v_{-i}^{(\ell)}))$
  end for
  Compute regret gradient: $\forall \ell \in [B], i \in N$:
    $g_{\ell,i}^t =$
    $\nabla_w \left[u_i^w(v_i^{(\ell)}; (\boxed{v_i'^{(\ell)}}, v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)}) \right] \Big|_{w=w^t}$
  Compute Lagrangian gradient using (5) and update w^t:
    $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
  Update Lagrange multipliers once in Q iterations:
    if t is a multiple of Q
      $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t\, \widetilde{rgt}_i(w^{t+1}), \quad \forall i \in N$
    else
      $\lambda^{t+1} \leftarrow \lambda^t$
end for

- estimate the empirical ex post regret for bidder i

Mini-batch的大小，L->B

$$\boxed{\widetilde{rgt}_i(w)} = \frac{1}{B} \sum_{\ell=1}^{B} \boxed{\max_{v_i' \in V_i} u_i^w\left(v_i^{(\ell)}; \left(v_i', v_{-i}^{(\ell)}\right)\right) - u_i^w\left(v_i^{(\ell)}; v^{(\ell)}\right)}$$

- The gradient of C_p w.r.t. w for fixed λ_t is given by

$$\nabla_w \mathcal{C}_\rho(w; \lambda^t) = -\frac{1}{B} \sum_{\ell=1}^{B} \sum_{i \in N} \nabla_w p_i^w\left(v^{(\ell)}\right)$$

$$+ \sum_{i \in N} \sum_{\ell=1}^{B} \lambda_i^t g_{\ell,i} + \rho \sum_{i \in N} \sum_{\ell=1}^{B} \widetilde{rgt}_i(\omega) g_{\ell,i}$$

$$g_{\ell,i} = \nabla_w \left[\max_{v_i' \in V_i} u_i^w\left(v_i^{(\ell)}; \left(v_i', v_{-i}^{(\ell)}\right)\right) - u_i^w\left(v_i^{(\ell)}; v^{(\ell)}\right) \right]$$

- perform R gradient updates to compute the optimal misreports

$$v_i'^{(\ell)} = v_i'^{(\ell)} + \gamma \nabla_{v_i'} u_i^\omega(v_i^{(\ell)}; (v_i'^{\ell}, v_{-i}^{(\ell)}))$$

梯度上升法

Experimental Results

$$rgt = \frac{1}{n}\sum_{i=1}^{n}\widehat{rgt}_i(f,p)$$

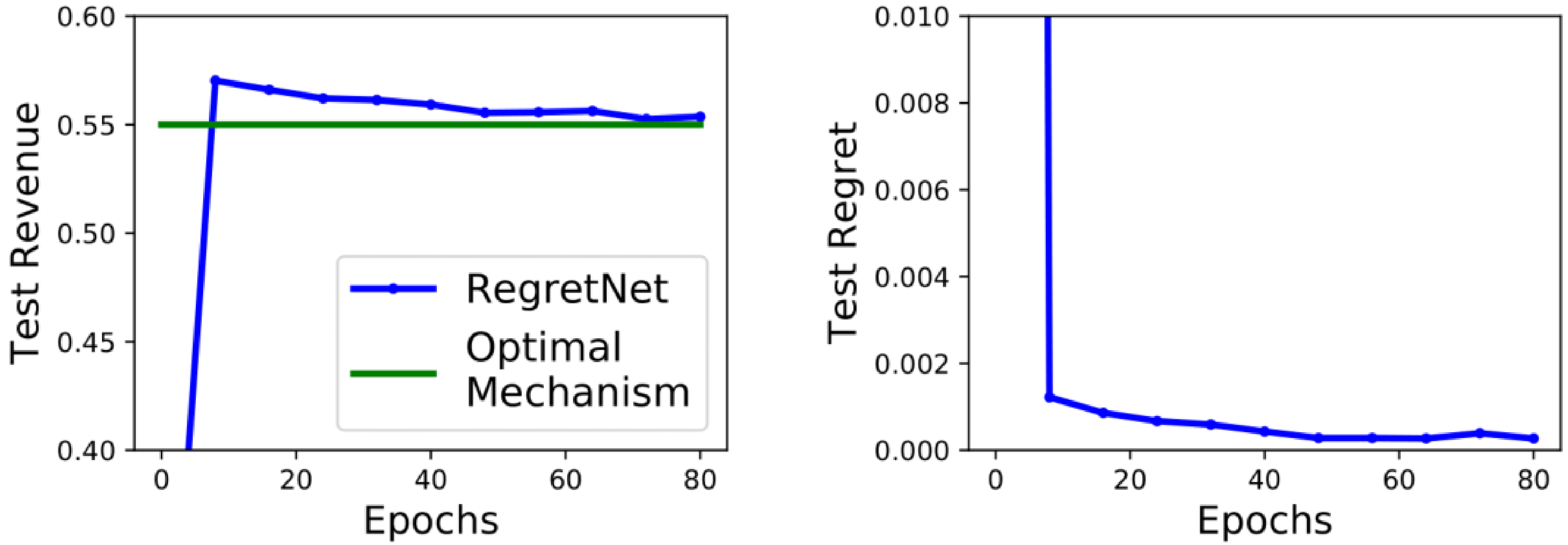
其他两种最先进机制

Dist.	OPT	rev	rgt
(I)	0.550	0.554	< 0.001
(II)	2.137	2.137	< 0.001

single bidder, 2 items (a) $U[0,1]$

Dist.	rev	rgt	VVCA	AMA_{bsym}
(III)	**0.878**	< 0.001	0.860	0.862
(IV)	**2.871**	< 0.001	2.741	2.765
(V)	**4.270**	< 0.001	4.209	3.748

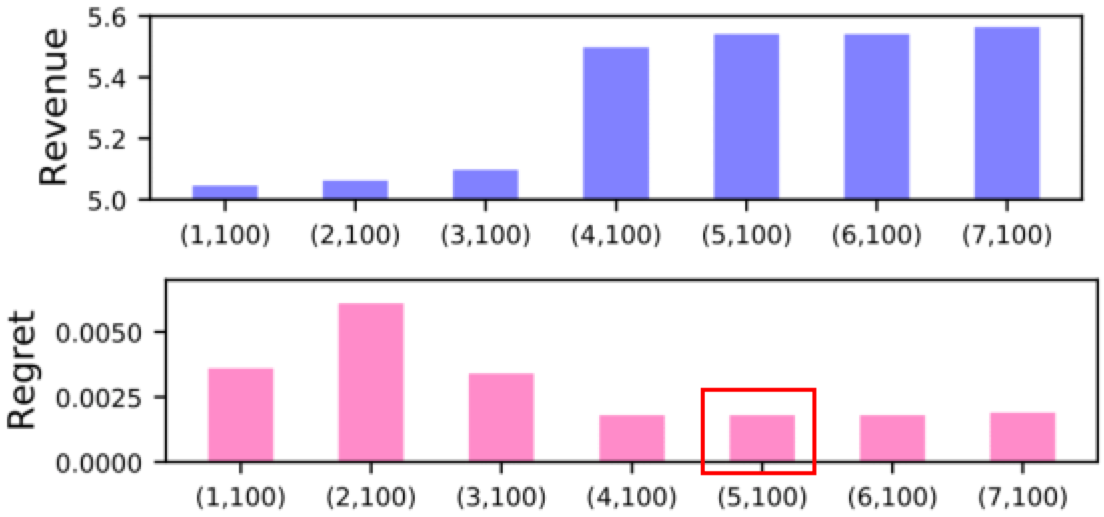
2 bidders, 2 items (b) $U[2,3]$



(c)

Figure 3: (a)-(b): Test revenue and regret for (a) single bidder, 2 items and (b) 2 bidder, 2 items settings. (c): Plot of test revenue and regret as a function of training epochs for setting (I).

Experimental Results (Scaling up)



(a)

- **Settings**

(VI) 3 additive bidders and 10 items, where bidders draw their value for each item from $U[0, 1]$.

(VII) 5 additive bidders and 10 items, where bidders draw their value for each item from $U[0, 1]$.

(R, K) denotes an architecture with R hidden layers and K nodes per layer

Distribution	rev	rgt	Item-wise Myerson	Bundled Myerson
VI: 3×10	**5.541**	< 0.002	5.310	5.009
VII: 5×10	**6.778**	< 0.005	6.716	5.453

(b)

Figure 5: (a) Revenue and regret on validation set for auctions learned for setting (VI) using different architectures. (b) Test revenue and regret for setting (VI) - (VII).