Automated Mechanism Design

Final Project Presentation

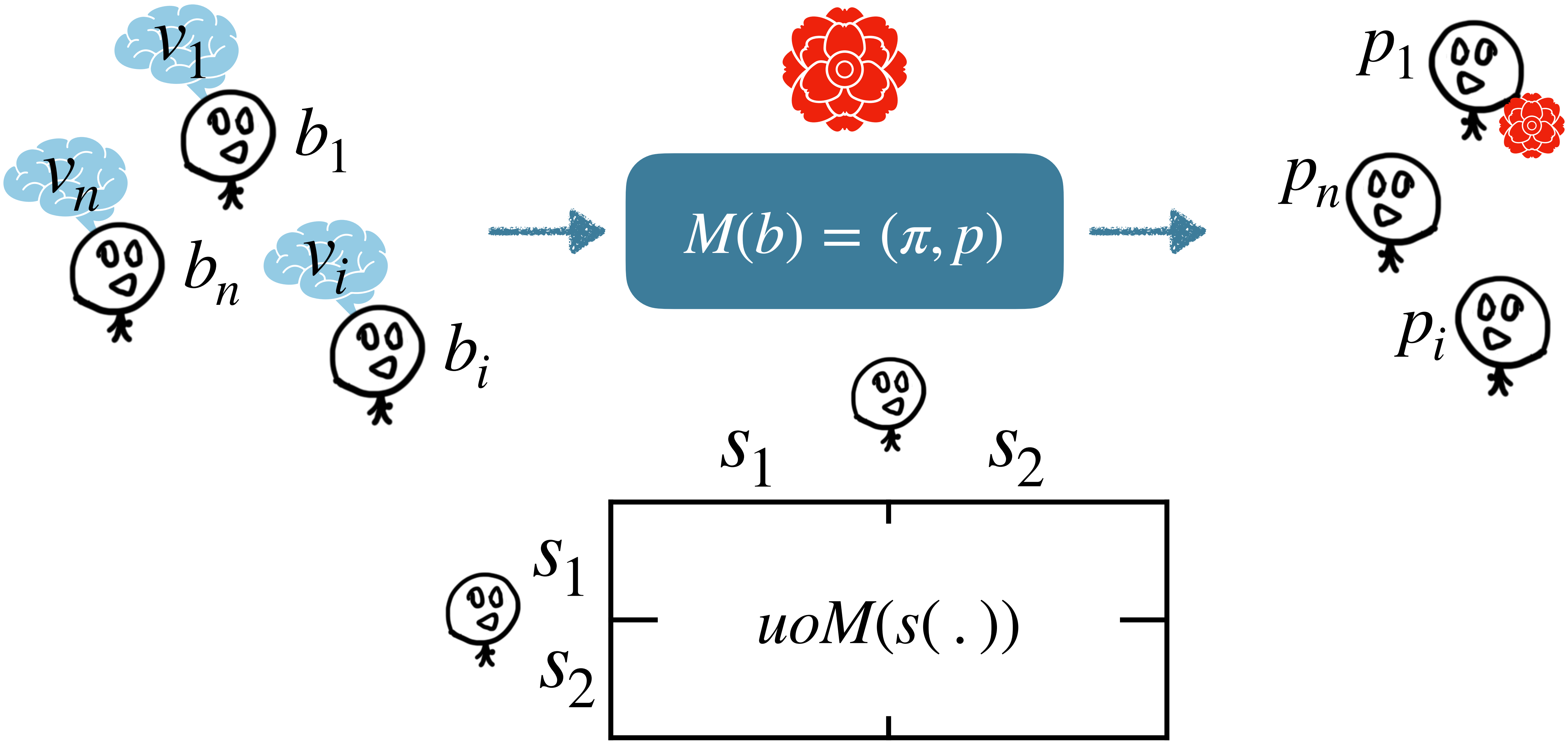
Yasmin Madani, Alireza Masoumian

11 Apr 2024

How can we learn the optimal mechanism with respect to a certain objective? (If there is any!)

- Revelation Principle → Direct and Truthful Mechanisms → Still Big!
- Trial and error ??!!

Recap: Mechanism Design



- Strategy: $s_i(v_i) = b_i$
- Direct: $\mathscr{V}_i = \mathscr{B}_i$ (e.g. $= \mathbb{R}$)
- Truthful: $s_i^*(v) = v \quad \forall v \in \mathscr{V}, \forall i \in [n]$

s_i^* is a dominant strategy for all the players (DSIC)

s^* is a NE (BIC)

Recap: Mechanism Design

- M should be an implementation of a <u>social choice function C</u>
- C maps a utility function profile to an outcome

$$\boxed{M(s^*(v)) = C(u(\,.\,, v))}$$

- Examples:

$$M(b) = (\pi, p)$$

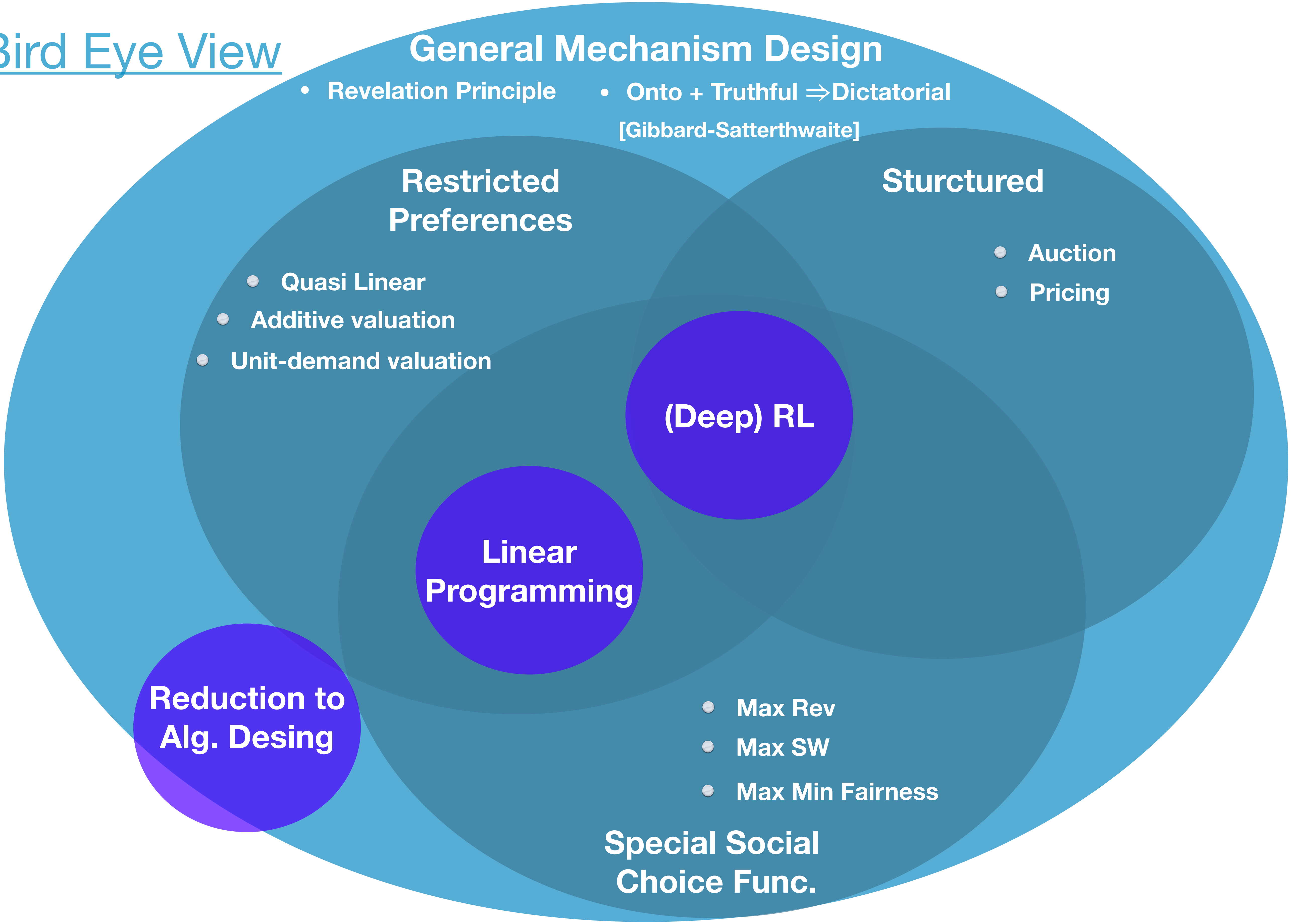
Max Social Welfare: $\max \sum_i v_i(\pi_i)$

Max Revenue: $\max \sum_i p_i$

Max Min Fairness: $\max \min_i v_i(\pi_i)$

Different objectives (C)

Bird Eye View



Mechanism Design to Algorithm Design

Mechanism Design $\xrightarrow{\text{Computationaly efficient reduction}}$ Algorithm Design

- Virtual Welfare, and Virtual Revenue

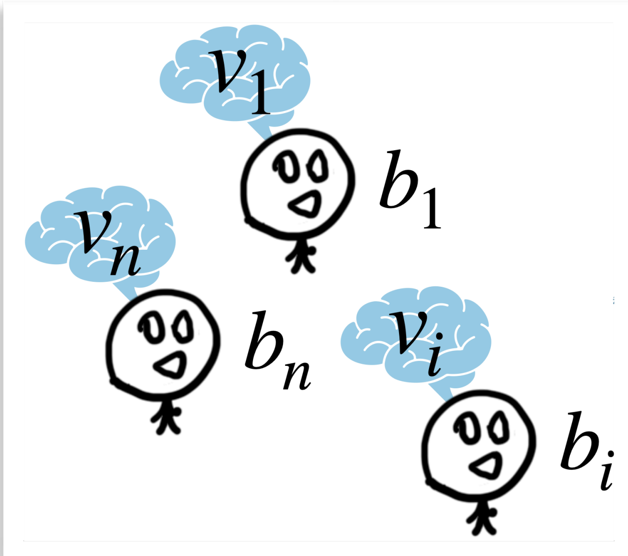
Mechanism Desgin → **Solve Any-Difference Problem (SADP)**

C = An allocation only objective → $C + VW$

C = General objective → $C + VR + \mathrm{VW}$

C = Revenue objective → VW

Common Theme



$$M(b) = (\pi, p)$$

Max Social Welfare: $\max \sum_i v_i(\pi_i)$

Max Revenue: $\max \sum_i p_i$

Max Revenue: $\max \min_i v_i(\pi_i)$

State (S)

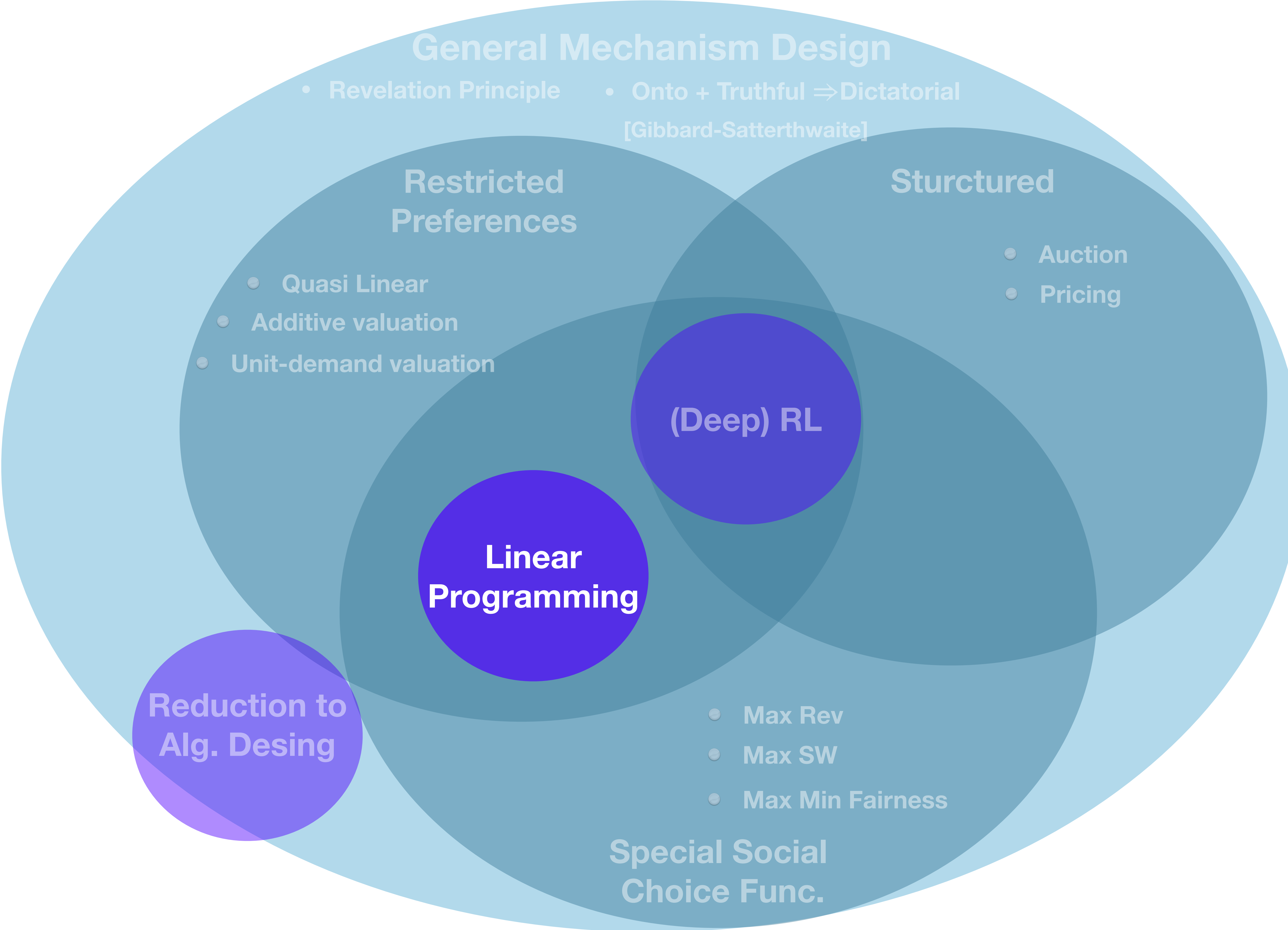
- Partial Observable
- Dynamic vs Static

Policy (π)

- What we learn!

Reward

- Depended on social choice funciton C


General Mechanism Design
Revelation Principle
Onto + Truthful ⇒Dictatorial
[Gibbard-Satterthwaite]
Restricted Preferences
Quasi Linear
Additive valuation
Unit-demand valuation
Sturctured
Auction
Pricing
(Deep) RL
Linear Programming
Reduction to Alg. Desing
Max Rev
Max SW
Max Min Fairness
Special Social Choice Func.

LP for Dynamic Mechanism Design

Can we efficiently compute optimal mechanism in unstructured dynamic environment?

- What is a dynamic env. ? $\longrightarrow$ $v_t(s, a), \quad P_t(s, a, s')$
- Incentive compatible : misreporting s' does not improve utilities
- Individually Rational $\rightarrow$ Overall IR, Dynamic IR
- Utilities:

$$u_P^M(h, s) = \sum_a \pi(h, s, a) \cdot \left(v_{|h|+1}^P(s, a) + \sum_{s'} P_{|h|+1}(s, a, s') \cdot u_P^M(h + (s, a), s') \right) + p(h, s),$$

$$u_A^M(h, s) = \sum_a \pi(h, s, a) \cdot \left(\underbrace{v_{|h|+1}^A(s, a)}_{\text{Instant reward}} + \underbrace{\sum_{s'} P_{|h|+1}(s, a, s') \cdot u_A^M(h + (s, a), s')}_{\text{Look ahead accumulated}} \right) \underbrace{- p(h, s)}_{\text{Payment}},$$

LP for Dynamic Mechanism Design

$$\textbf{objective:} \quad \max \sum_{h \in \mathcal{H}, s \in \mathcal{S}: (h,s) \text{ is feasible}} \left(\sum_{a \in \mathcal{A}} v^P_{|h|+1}(s,a) \cdot x(h,s,a) + y(h,s) \right)$$

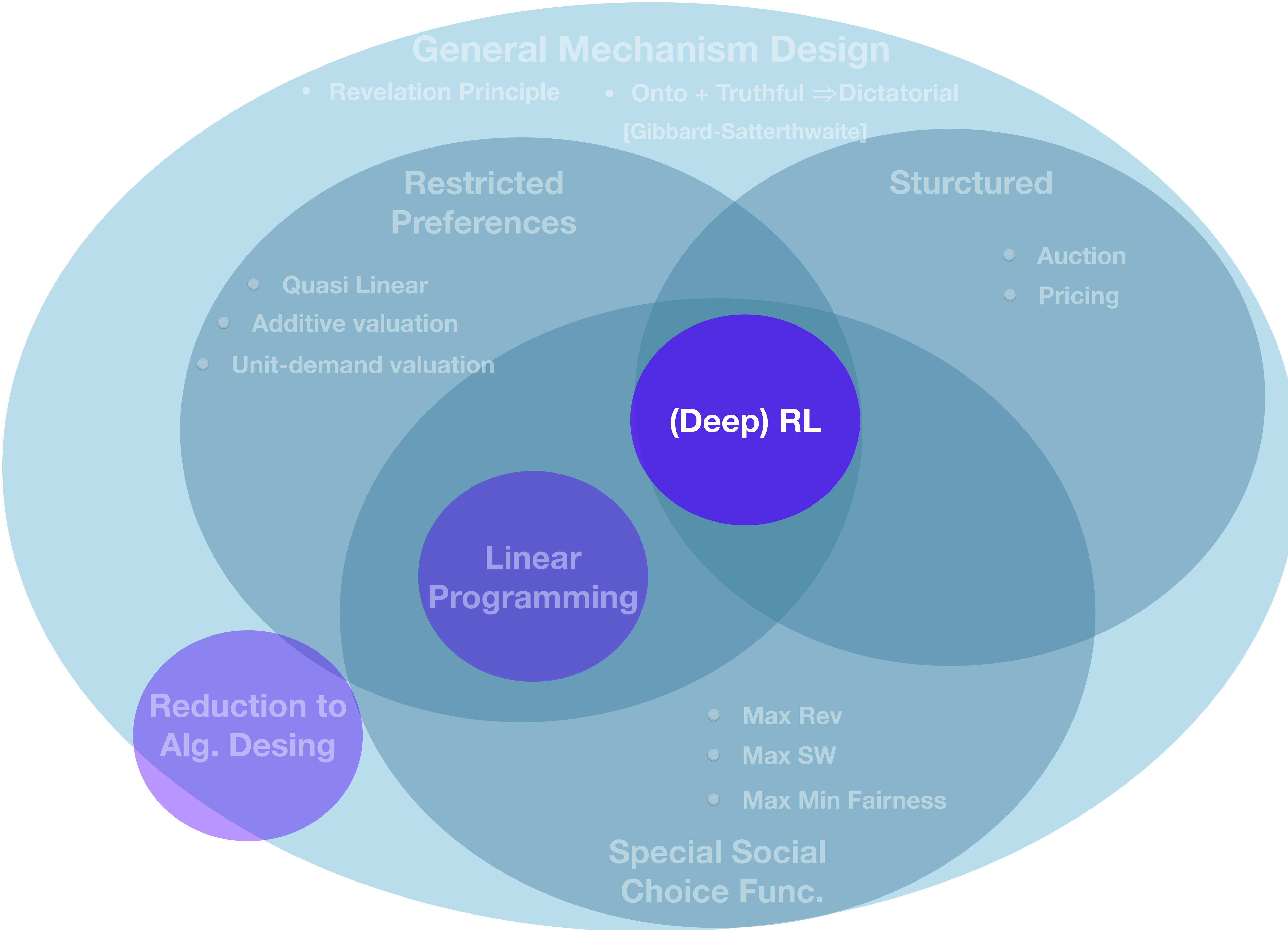
Probability $\rightarrow \pi$ Payment $\rightarrow p$

- All the constrains for IC and IR are linear

Theorem:

$O(poly(|S|^T, |A|^T, L)) \Rightarrow$ An IR, IC mechanism implementing *max* Revenue

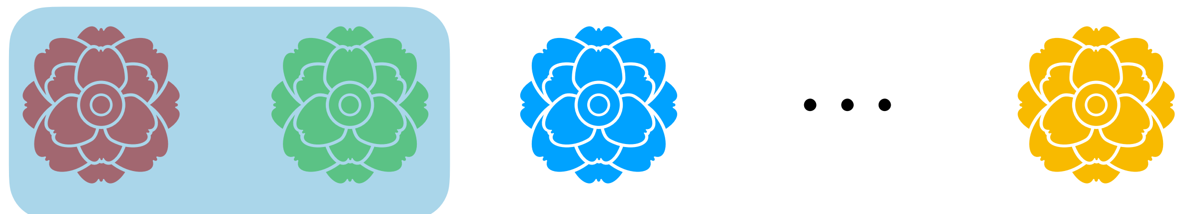
- Deterministic Mechanism → Non-linear Constraints
- Arbitrary large T → NP-Hard to approximate up to a multiplicative constant


General Mechanism Design
Revelation Principle
Onto + Truthful ⇒Dictatorial
[Gibbard-Satterthwaite]
Restricted Preferences
Quasi Linear
Additive valuation
Unit-demand valuation
Sturctured
Auction
Pricing
(Deep) RL
Linear Programming
Reduction to Alg. Desing
Max Rev
Max SW
Max Min Fairness
Special Social Choice Func.

POMDP for Sequential Pricing

- Sequential Pricing Mechanism (SPM)

$$p_1^t \quad p_2^t \quad p_3^t \quad \cdots \quad p_m^t$$



$$x^t$$

- Agent i^t takes bundle based on his valuation funciton v_i
- After time $T = n \rightarrow$ the outcome of the episod is determined $(\boldsymbol{x}, p)$
- Terminal Reward of the episode := $C(\boldsymbol{x}, p)$

Episodic POMDP

- Action $a^t := (p^t, i^t)$
- Observation $o^t := x^t$
- State $s^t := v$
- Reward $r^\tau := C(\boldsymbol{x}, p)$

POMDP for Sequential Pricing

- POMDP $\Rightarrow$ Optimal policy might not be memoryless

How complex can be the policy wrt the history?

Proposition:

There exists a value function (an environment) where the Welfare optimal policy does not lie on the linear policy set!

- By linear we mean using the allocation matrix and remaining agents. i.e $\theta \in \mathbb{R}^{n+m}$
- Good to use Neural Nets and deep RL algorithms.

Optimal Auction Desing Through Deep Learning

Optimal Auction design → Constrained learning problem

- Using Multi-layer NN to encode rules of auctions.
- Exploit the Objective on the training!!: We want to converge to an IC mechanism
- Common prior $F = (F_1, \ldots, F_n)$ Distributions over valuations $v_i \sim F_i$
- What are the inputs? Bids or valuations?!!

Take Away

- There are some imposibility results for the general MD problem.
- For some specific settings we can model the MD problem as a learning problem.
- There is a meaningful connection between the components of MD and that of RL

Reference

Understanding Incentives: Mechanism Design becomes Algorithm Design

Yang Cai*
EECS, MIT
ycai@csail.mit.edu

Constantinos Daskalakis†
EECS, MIT
costis@mit.edu

S. Matthew Weinberg‡
EECS, MIT
smw79@mit.edu

May 20, 2013

Automated Dynamic Mechanism Design

Hanrui Zhang
Duke University

Vincent Conitzer
Duke University

Reinforcement Learning of Sequential Price Mechanisms*

Gianluca Brero[a], Alon Eden[a], Matthias Gerstgrasser[a], David C. Parkes[a], and Duncan Rheingans-Yoo[a]

[a]Harvard University

Optimal Auctions through Deep Learning: Advances in Differentiable Economics*

Paul Dütting[a], Zhe Feng[a], Harikrishna Narasimhan[a], David C. Parkes[b], and Sai Srivatsa Ravindranath[b]

Thank you