Understanding Iterative Combinatorial Auction Designs with Multi-Agent Reinforcement Learning

Greg d'Eon, Neil Newman, Kevin Leyton-Brown
University of British Columbia



EC '24
July 11, 2024

Iterative Combinatorial Auctions

Iterative combinatorial auctions (ICAs):

- **Combinatorial:** bid on bundles of items
- **Iterative:** multiple bidding rounds

Iterative Combinatorial Auctions

Iterative combinatorial auctions (ICAs):

- **Combinatorial:** bid on bundles of items
- **Iterative:** multiple bidding rounds

Spectrum auctions: auctioning the right to use EM spectrum (radio, TV, cellular data...)

- **Complex:** dozens of bidders, hundreds of products, weeks of bidding
- **High stakes:** $1B+ of revenue; strategic bidding is potentially lucrative

Iterative Combinatorial Auctions

Iterative combinatorial auctions (ICAs):

- **Combinatorial:** bid on bundles of items
- **Iterative:** multiple bidding rounds

Spectrum auctions: auctioning the right to use EM spectrum (radio, TV, cellular data...)

- **Complex:** dozens of bidders, hundreds of products, weeks of bidding
- **High stakes:** $1B+ of revenue; strategic bidding is potentially lucrative

How should a bidder **bid**?

How should an auction designer **set the rules**?

Analyzing Iterative Combinatorial Auctions

Many existing methods aren't suitable:

- **Pen-and-paper analysis:** requires restrictive assumptions

  (e.g., Riedel and Wolfstetter, 2006: assumed one product and perfect information)

Analyzing Iterative Combinatorial Auctions

Many existing methods aren't suitable:

- **Pen-and-paper analysis:** requires restrictive assumptions

  (e.g., Riedel and Wolfstetter, 2006: assumed one product and perfect information)

- **Traditional equilibrium solvers:** infeasible

  (enormous extensive-form representations)

Analyzing Iterative Combinatorial Auctions

Many existing methods aren't suitable:

- **Pen-and-paper analysis:** requires restrictive assumptions

  (e.g., Riedel and Wolfstetter, 2006: assumed one product and perfect information)

- **Traditional equilibrium solvers:** infeasible

  (enormous extensive-form representations)

- **Field testing:** too infrequent/high-stakes to learn from data

  (spectrum auctions: every few years, with constantly changing rules)

Multi-Agent Reinforcement Learning

Can off-the-shelf multi-agent reinforcement learning (MARL) algorithms help?

(e.g., algorithms developed for training **poker** agents?)

Multi-Agent Reinforcement Learning

Can off-the-shelf multi-agent reinforcement learning (MARL) algorithms help?

(e.g., algorithms developed for training **poker** agents?)

Unlikely to make “superhuman” autonomous bidders!

Still, valuable for:

- providing **examples** of strong bidding behavior
- building a strategic **playbook**
- evaluating likely costs and benefits of candidate **rule changes**

This Talk

Using MARL algorithms effectively takes care: need to

- Balance real-world fidelity with tractability in the auction **model**
- Navigate common pitfalls of MARL **algorithms**
- **Validate** and **interpret** learned policies

This Talk

Using MARL algorithms effectively takes care: need to

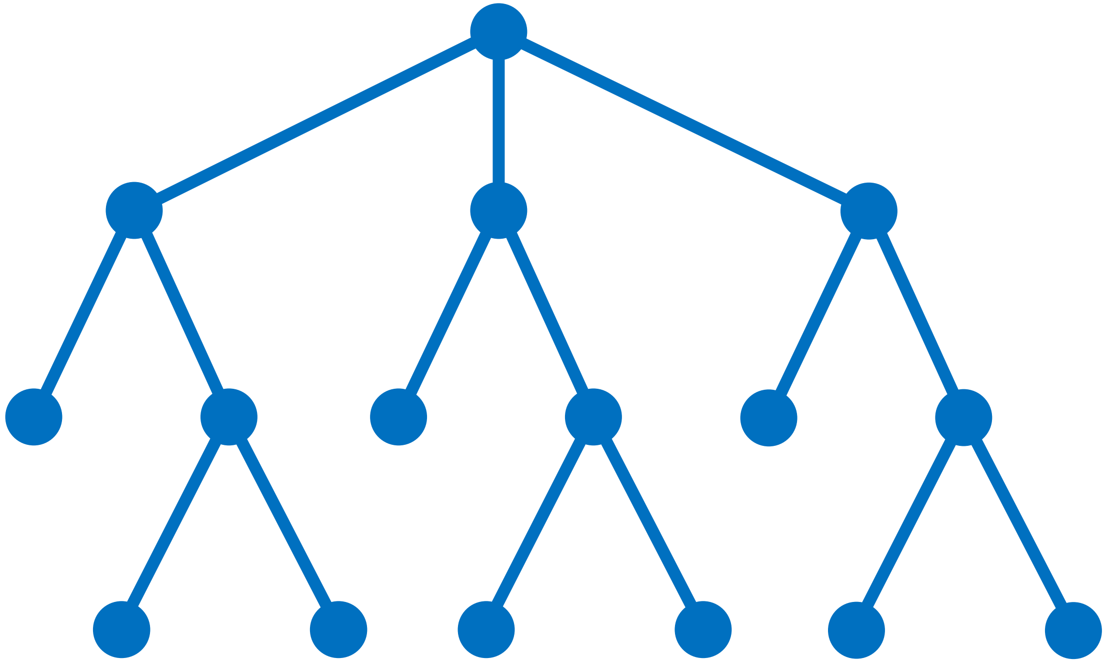
- Balance real-world fidelity with tractability in the auction **model**
- Navigate common pitfalls of MARL **algorithms**
- **Validate** and **interpret** learned policies

When done right, can be a **powerful tool!**

- **Case study:** for one potential clock auction rule change, non-trivial **behavior changes** lead to substantially different **auction outcomes**

Modeling an Auction

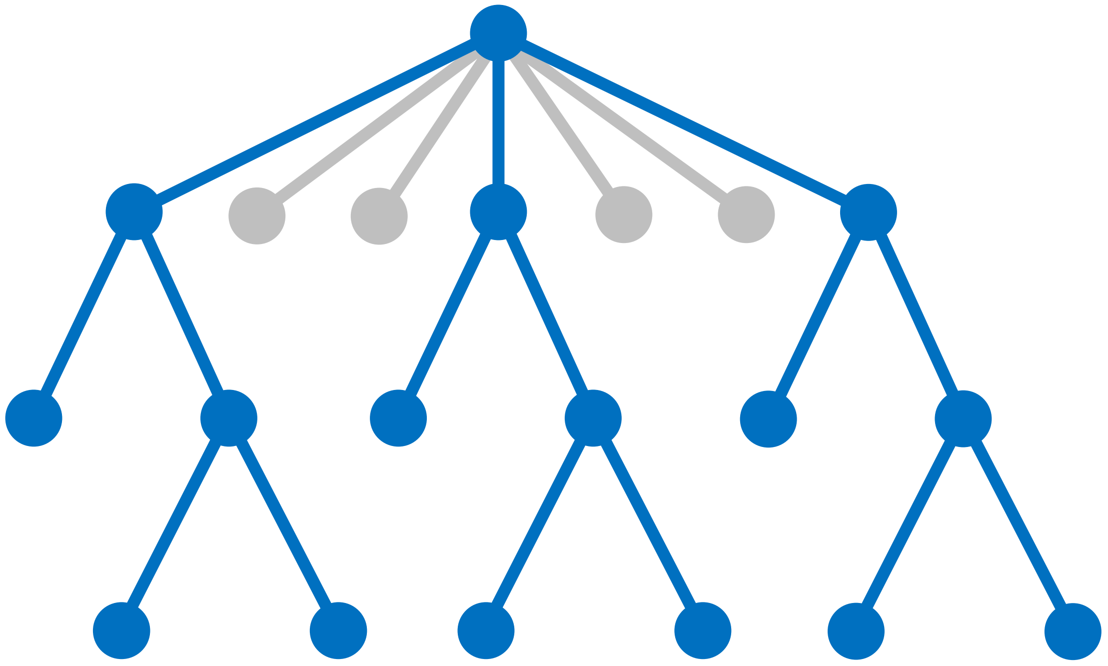
Control number of infostates
without losing key strategic elements:



Modeling an Auction

Control number of infostates without losing key strategic elements:

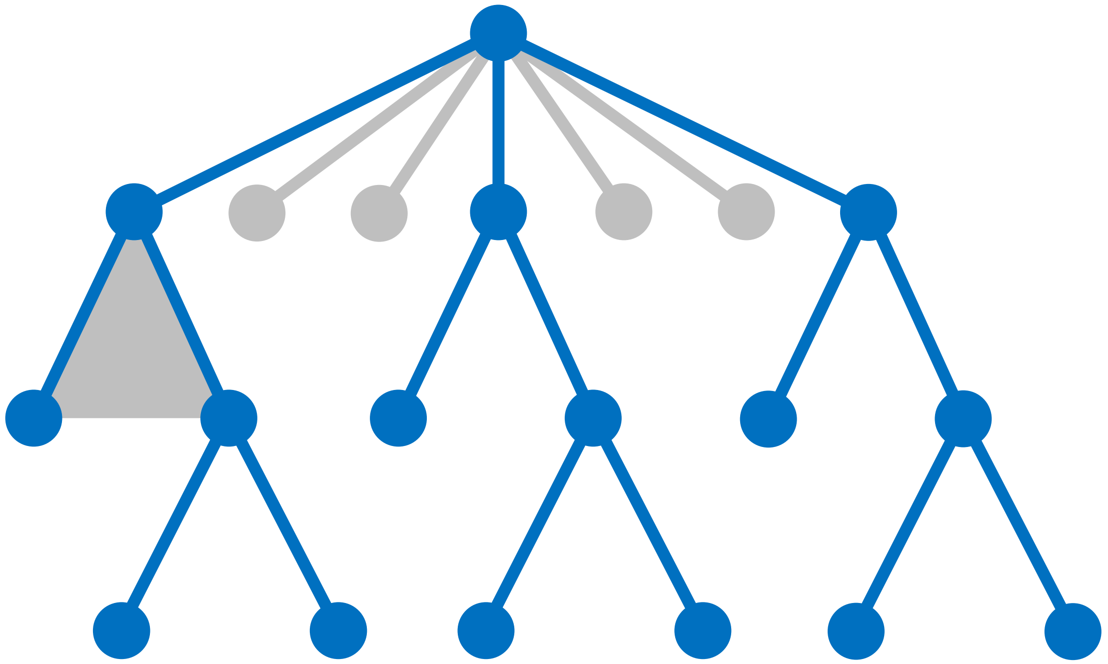
- Restrict number of **actions**



Modeling an Auction

Control number of infostates
without losing key strategic elements:

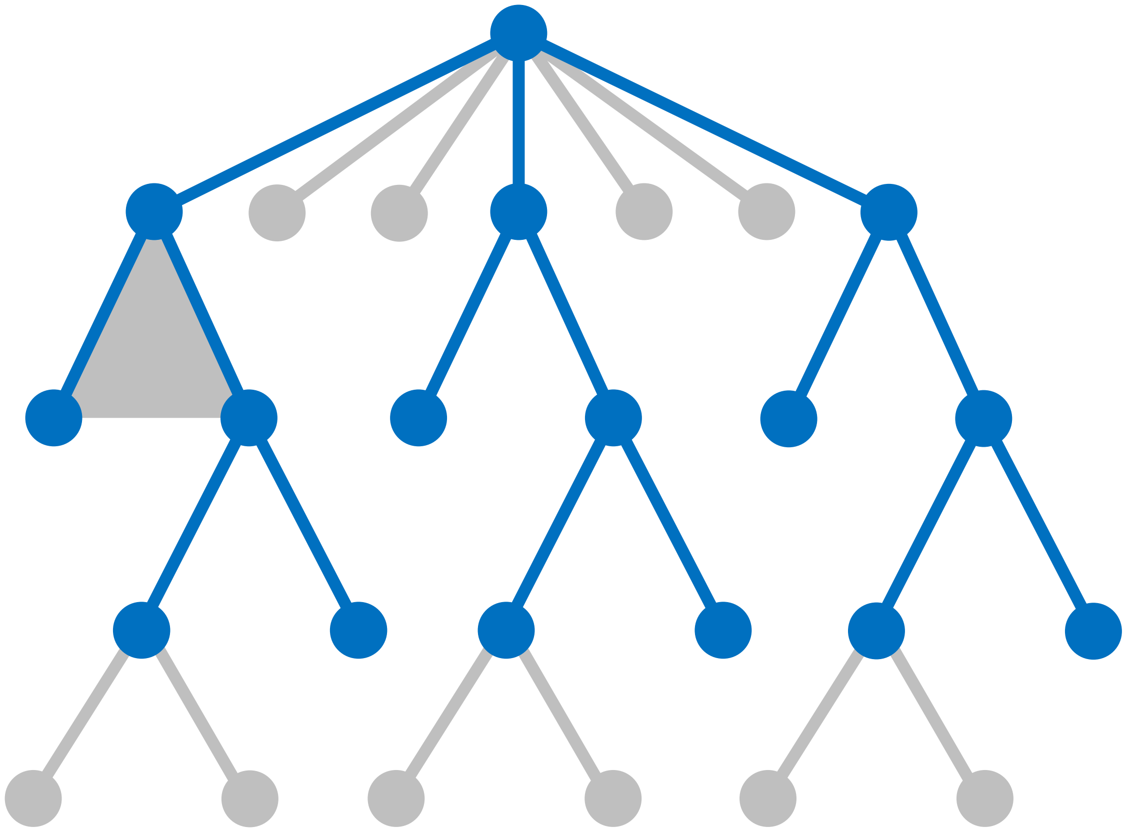
- Restrict number of **actions**
- **Discretize** continuous action spaces



Modeling an Auction

Control number of infostates
without losing key strategic elements:

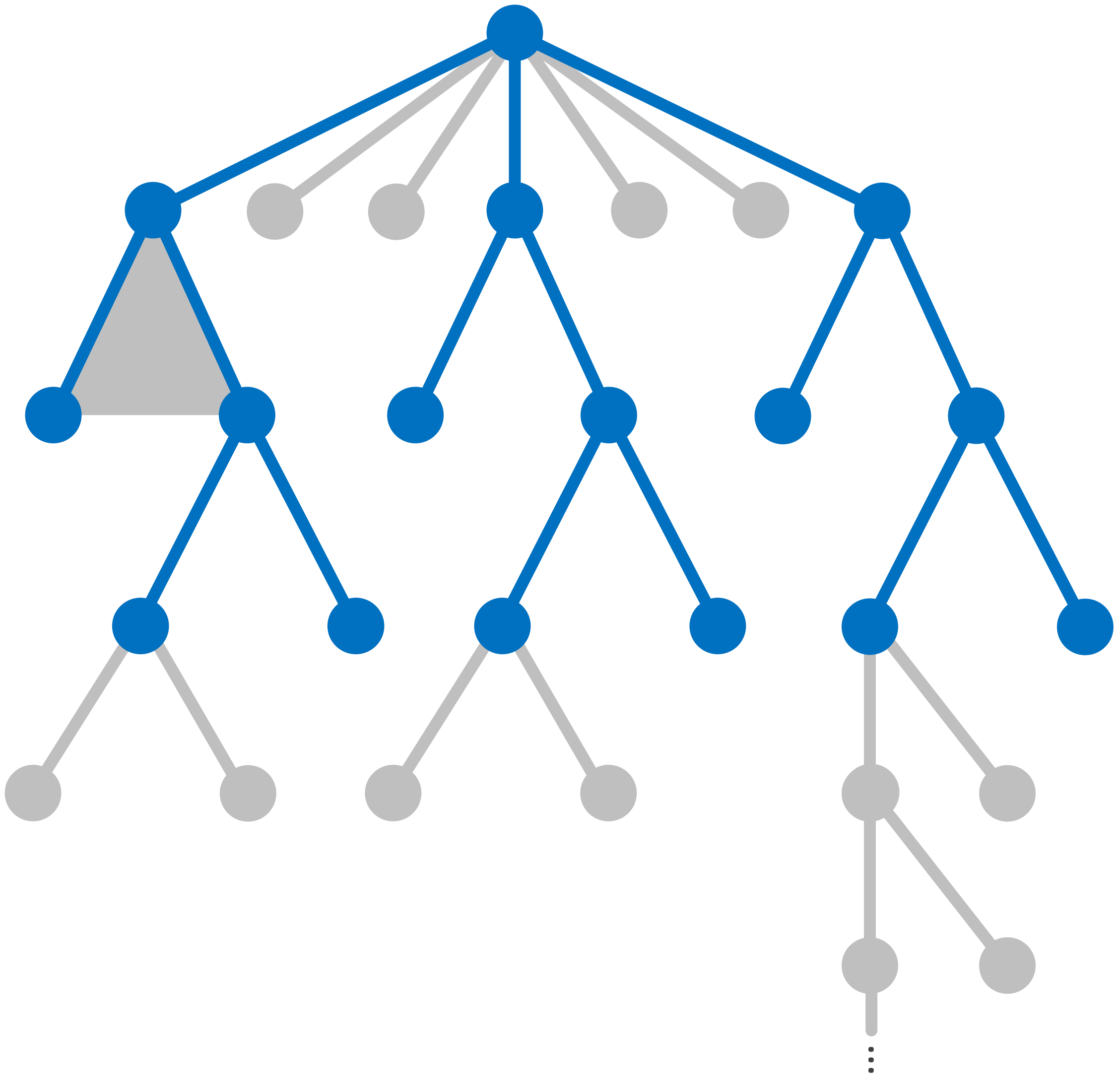
- Restrict number of **actions**
- **Discretize** continuous action spaces
- Limit auction **length**



Modeling an Auction

Control number of infostates without losing key strategic elements:

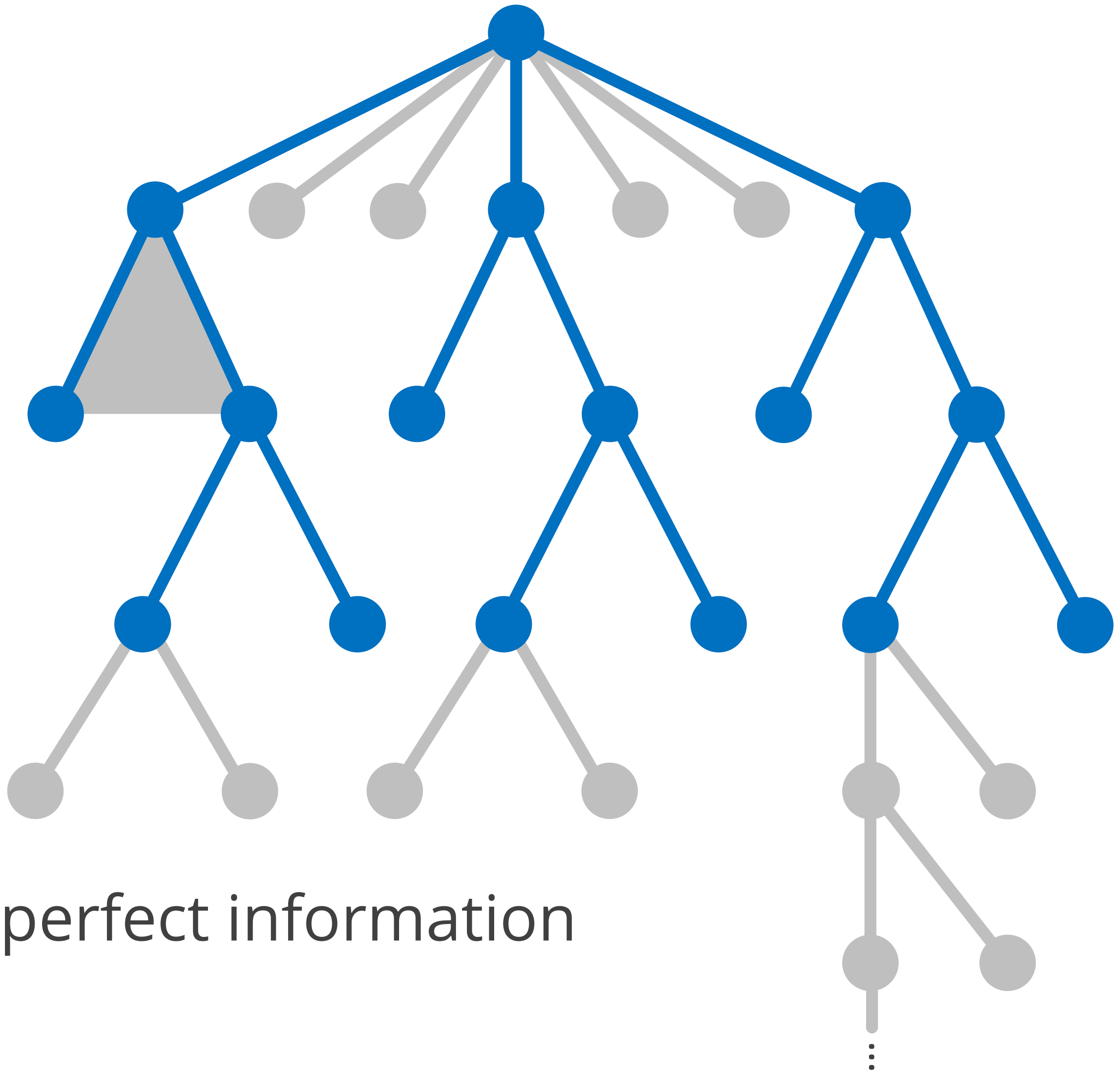
- Restrict number of **actions**
- **Discretize** continuous action spaces
- Limit auction **length**
- Avoid **infinite-length histories**



Modeling an Auction

Control number of infostates
without losing key strategic elements:

- Restrict number of **actions**
- **Discretize** continuous action spaces
- Limit auction **length**
- Avoid **infinite-length histories**



Some features are **ideal** for MARL:

- Asymmetric bidders, case-based rules, imperfect information

Finding Equilibria

Two key aspects of MARL algorithms:

- Policies: represent with a **lookup table** or **function approximation**

  (lookup tables more stable; function approximation necessary for scale)

- Exploration: **single path** or **counterfactual actions** in each iteration

  (exploring one path scales further, but can struggle to train effectively)

Finding Equilibria

Two key aspects of MARL algorithms:

- Policies: represent with a **lookup table** or **function approximation**
  (lookup tables more stable; function approximation necessary for scale)
- Exploration: **single path** or **counterfactual actions** in each iteration
  (exploring one path scales further, but can struggle to train effectively)

Other considerations:

- Break **indifferences** between identical rewards
- Consider restricting policies to **pure strategies**
- Find **multiple equilibria** by running with multiple seeds

Validating & Interpreting Policies

Test for convergence by computing **NashConv:** sum of each player's **regret**

(possible gain in utility by best-responding, holding opponents fixed)

- Smaller games: compute exactly with depth-first search
- Larger games: lower-bound with single-agent RL

Validating & Interpreting Policies

Test for convergence by computing **NashConv:** sum of each player's **regret**

(possible gain in utility by best-responding, holding opponents fixed)

- Smaller games: compute exactly with depth-first search
- Larger games: lower-bound with single-agent RL

Auction statistics alone can give helpful insight (revenue, welfare, length, …)

- With multiple equilibria, report **ranges**, not averages

Case Study: Clock Auctions

Auctioneer has:

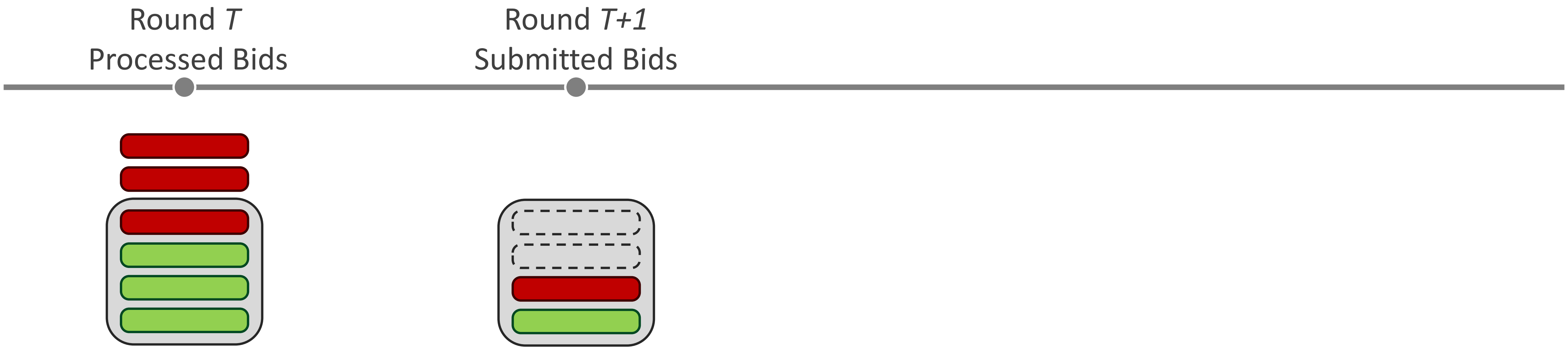
- A set of **regions**
- A number of (identical) **items** to sell in each region

Basic clock auction: set **initial prices** for each region; in each **round**,

- Every bidder makes a **bid** (vector of quantities for each region)
- If demand ≤ supply in every region, **end auction**
- Else, **reveal total demands** to bidders
  and **raise prices** on over-demanded regions

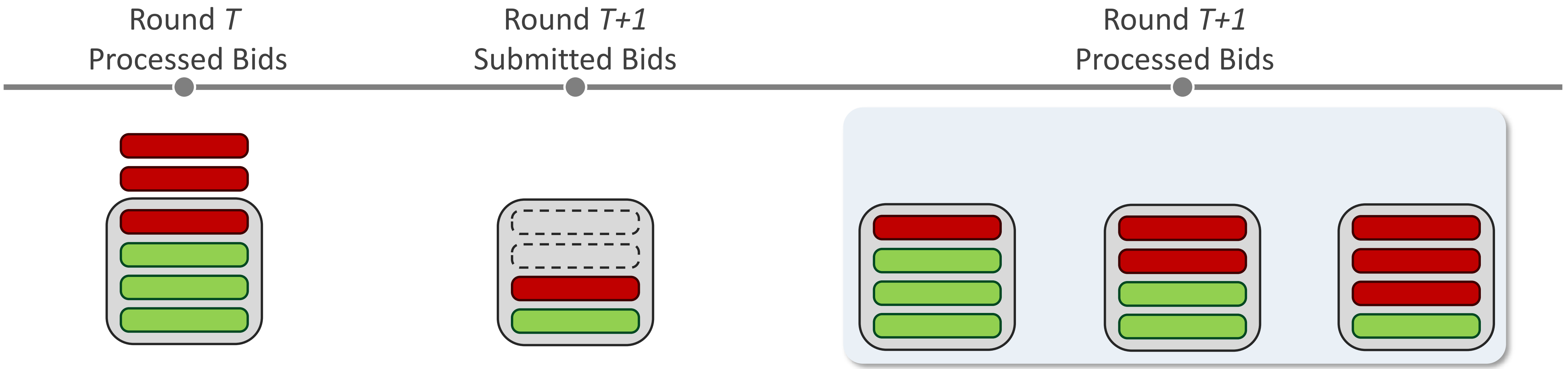
Case Study: Bid Processing

Undersell rule: don’t allow demand < supply



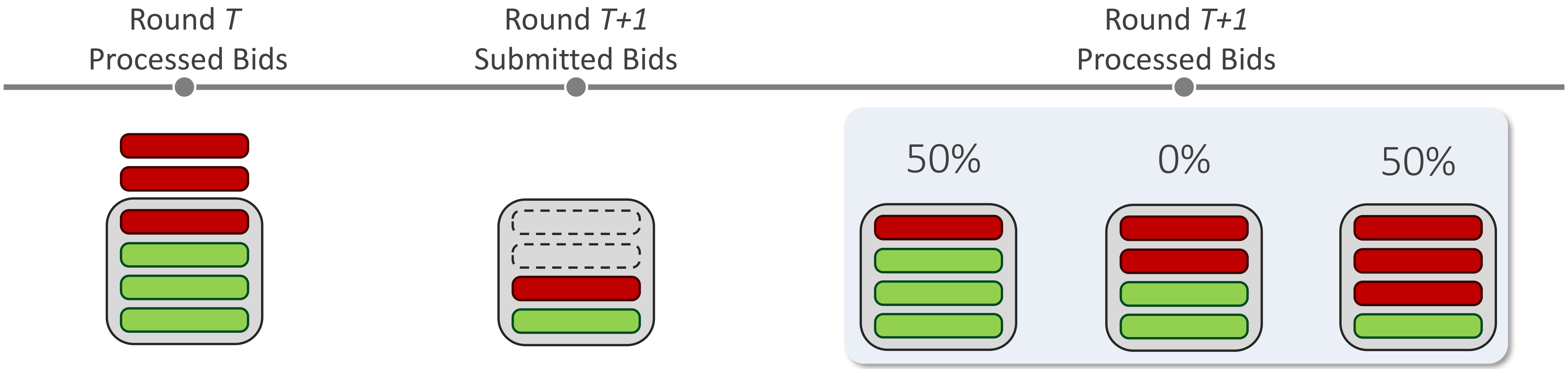
Case Study: Bid Processing

Undersell rule: don’t allow demand < supply



Case Study: Bid Processing

Undersell rule: don't allow demand < supply

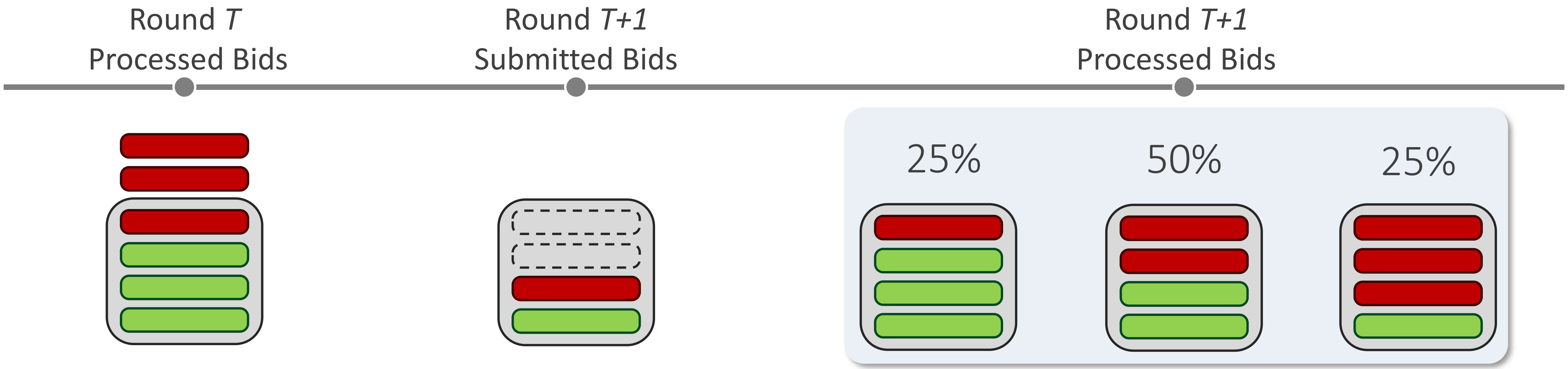


Two natural tiebreaking solutions:

- **Drop-by-bidder:** process each **bid** in a random order

Case Study: Bid Processing

Undersell rule: don't allow demand < supply



Two natural tiebreaking solutions:

- **Drop-by-bidder:** process each **bid** in a random order
- **Drop-by-license:** process each **unit of demand** in a random order

Case Study: Experiments

Auction: 2 bidders; 2 regions with {4, 1} licenses

- Value functions drawn from MRVM model [Weiss et al., 2017]
- 5 games with 500-700 information states

Case Study: Experiments

Auction: 2 bidders; 2 regions with {4, 1} licenses

- Value functions drawn from MRVM model [Weiss et al., 2017]
- 5 games with 500-700 information states

1. Monte-Carlo Counterfactual Regret Minimization (MCCFR)
    - Tabular policy; explores counterfactual actions
    - **Easy to use:** required little tuning

Case Study: Experiments

Auction: 2 bidders; 2 regions with {4, 1} licenses

- Value functions drawn from MRVM model [Weiss et al., 2017]
- 5 games with 500-700 information states

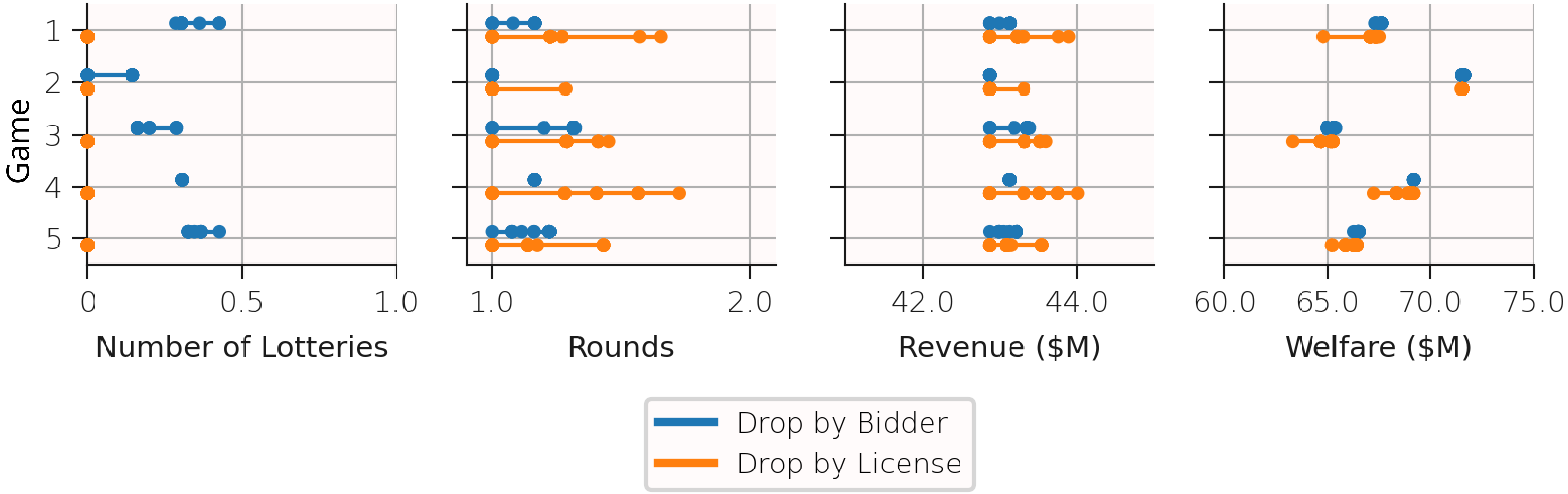
1. Monte-Carlo Counterfactual Regret Minimization (MCCFR)
   - Tabular policy; explores counterfactual actions
   - **Easy to use:** required little tuning

2. Proximal Policy Optimization (PPO)
   - Function approximation; single path
   - **Needs tuning:** few hyperparameter settings worked well

Case Study: Results

Drop-by-license: bidders **completely avoid** tiebreaks

- Leads to **longer auctions** with **higher revenue** and **lower welfare**



Conclusion

Multi-agent RL: a potentially powerful tool for economic analysis

- **Model**, **algorithm**, and **validation** require care
- When done right, can give empirical solutions to problems **out of reach for traditional methods**

Thank you!

✉️ gregdeon@cs.ubc.ca

🌎 gregdeon.com

💻 github.com/newmanne/open_spiel

ArXiv