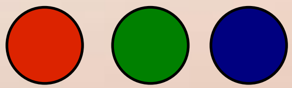
Beyond Equilibria:
Mechanisms for Repeated Combinatorial Auctions

Brendan Lucier
University of Toronto

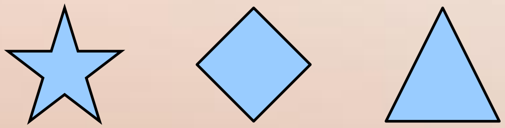
ICS 2010

Combinatorial Auctions

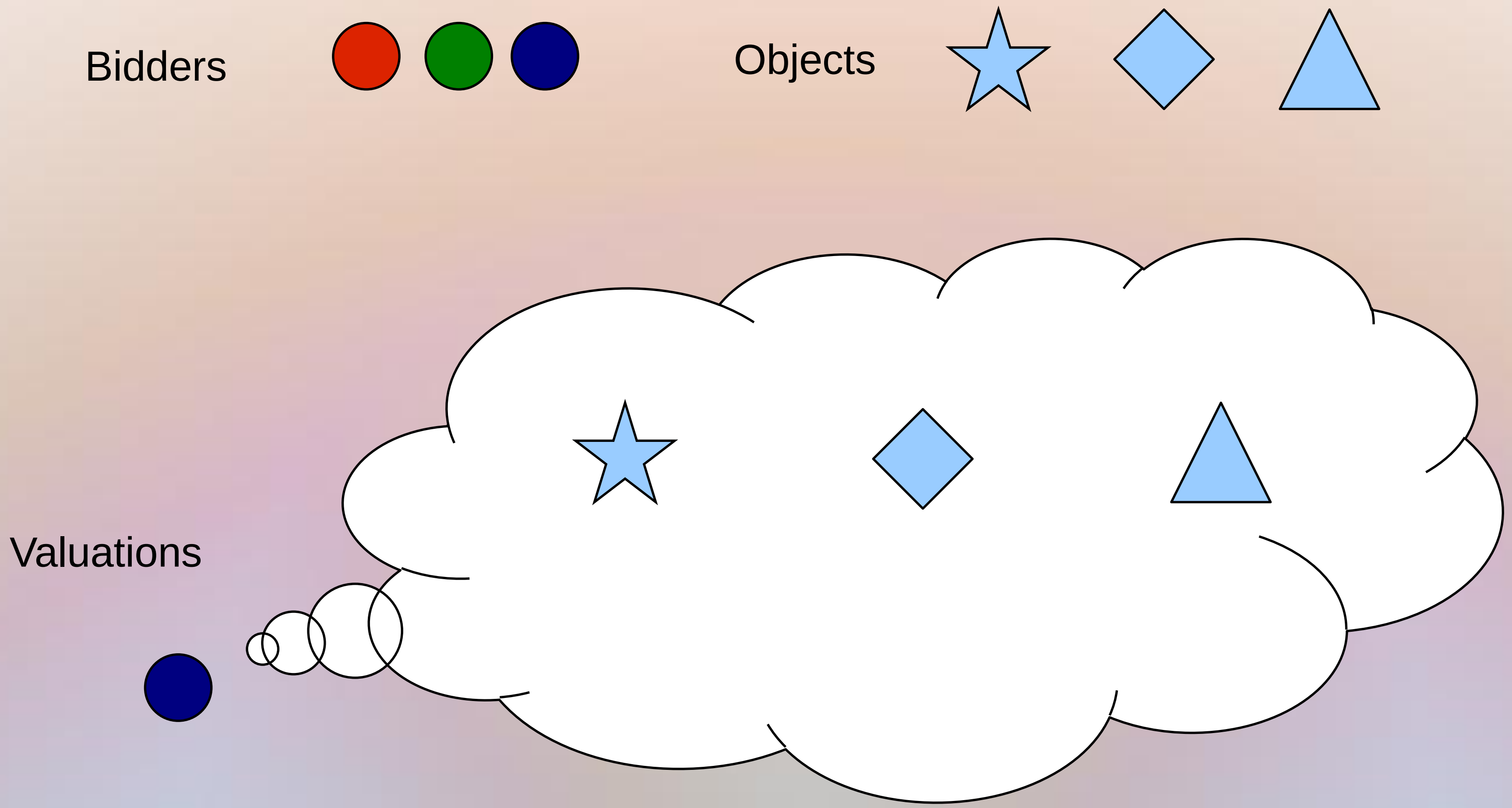
Bidders



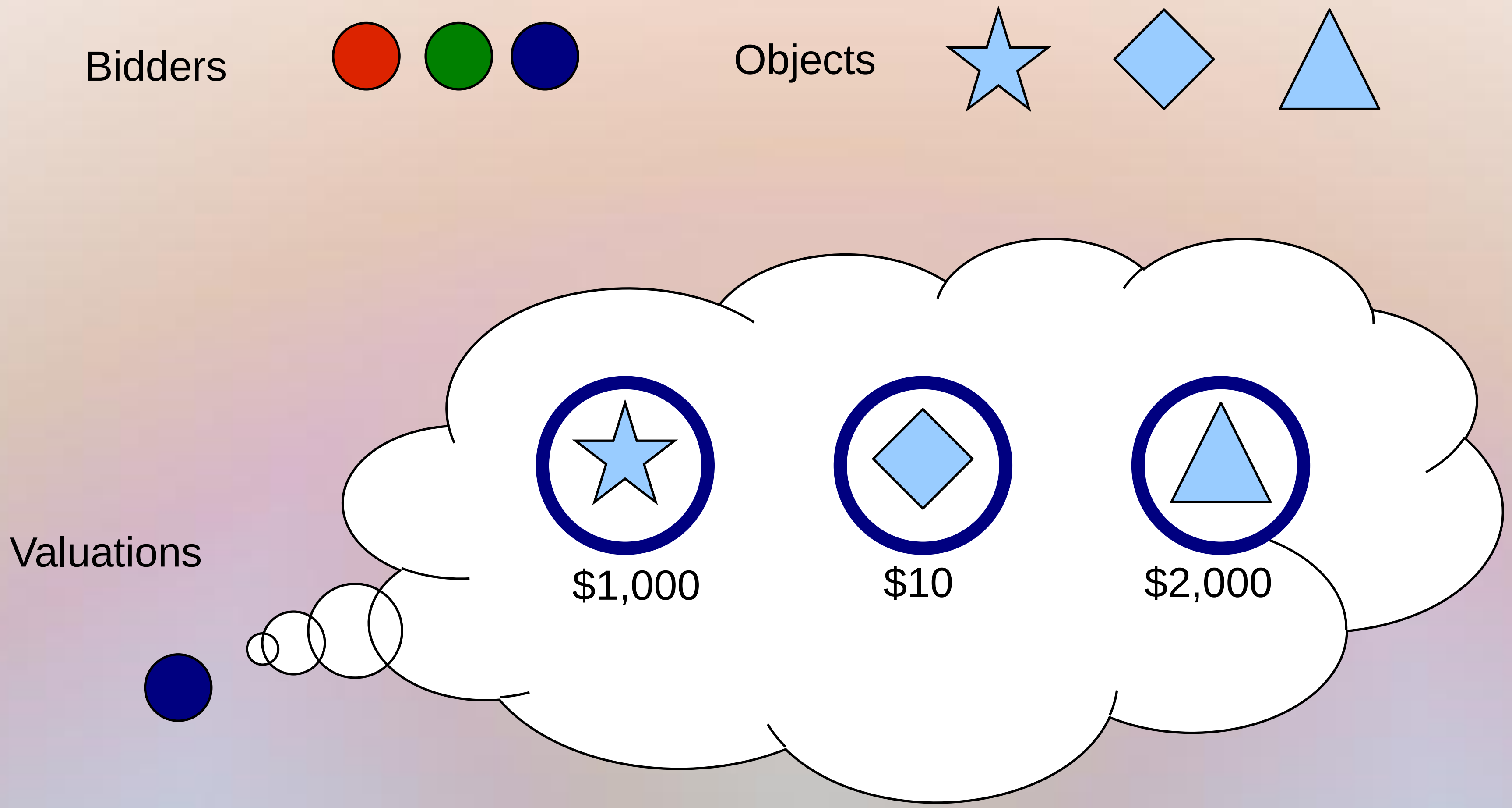
Objects



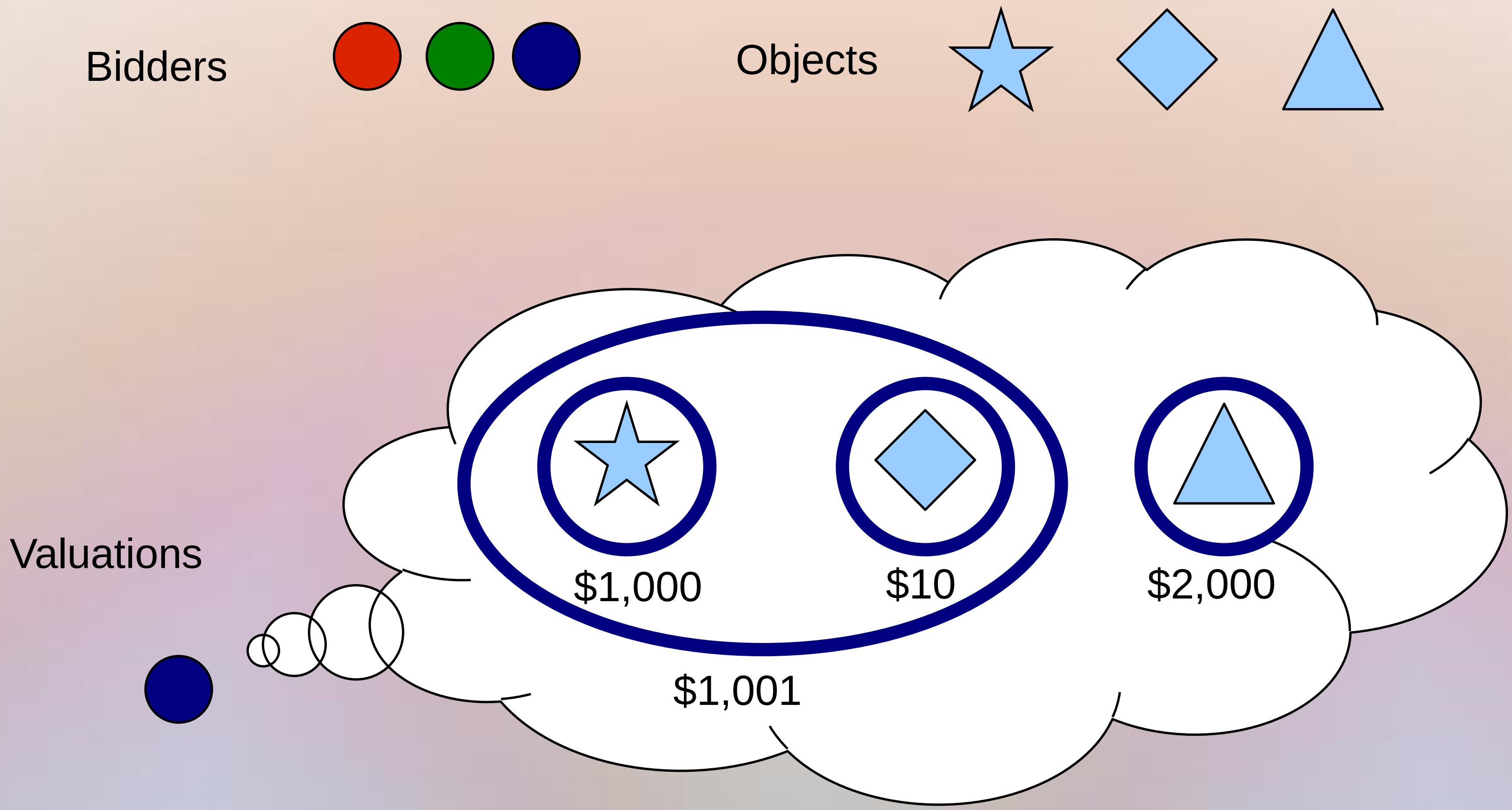
Combinatorial Auctions



Combinatorial Auctions

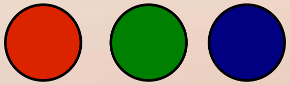


Combinatorial Auctions

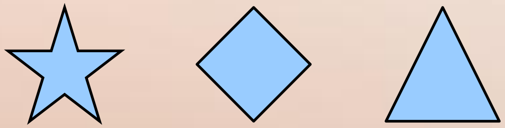


Combinatorial Auctions

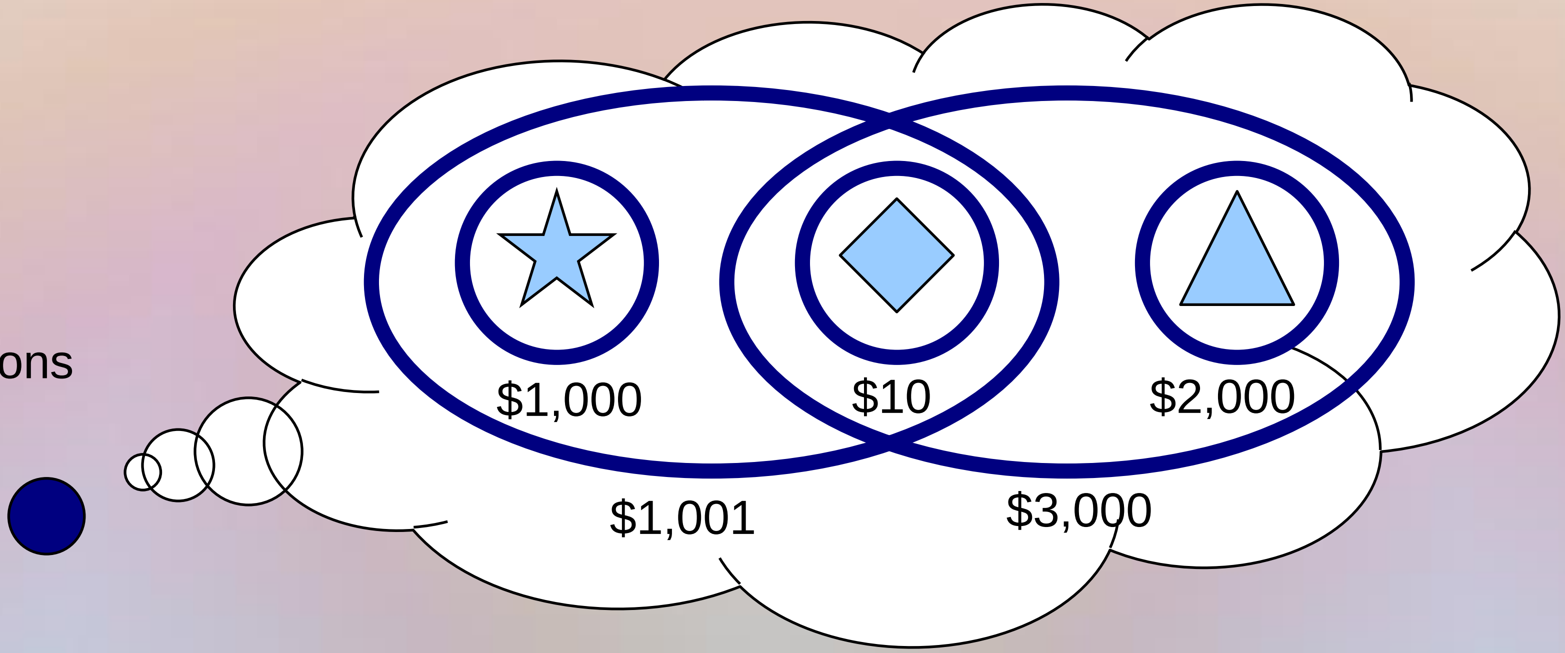
Bidders



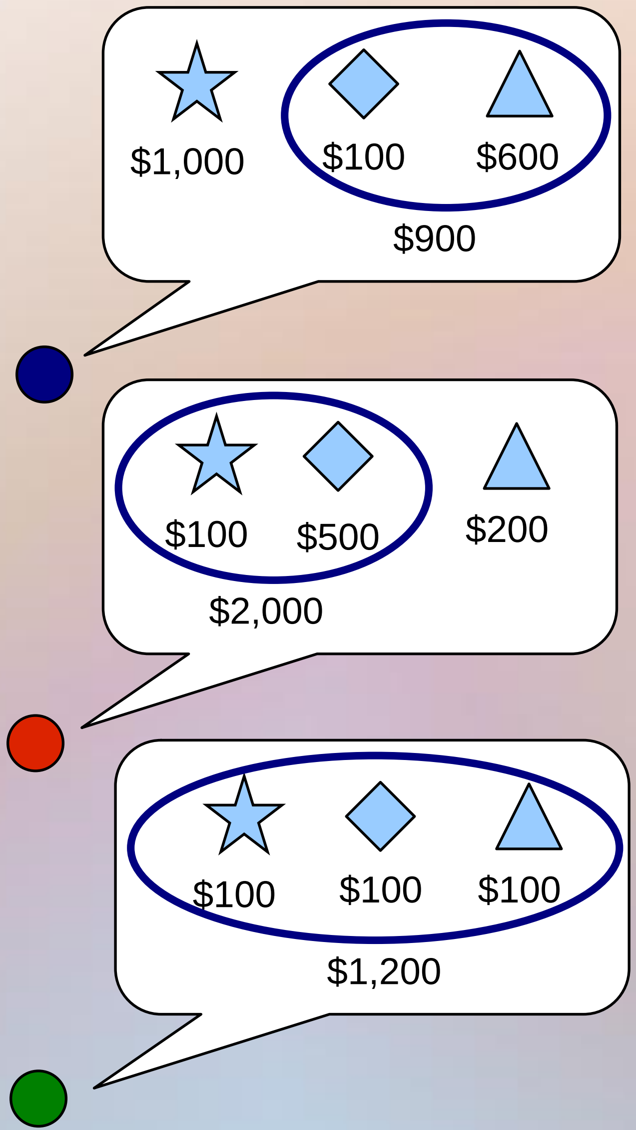
Objects



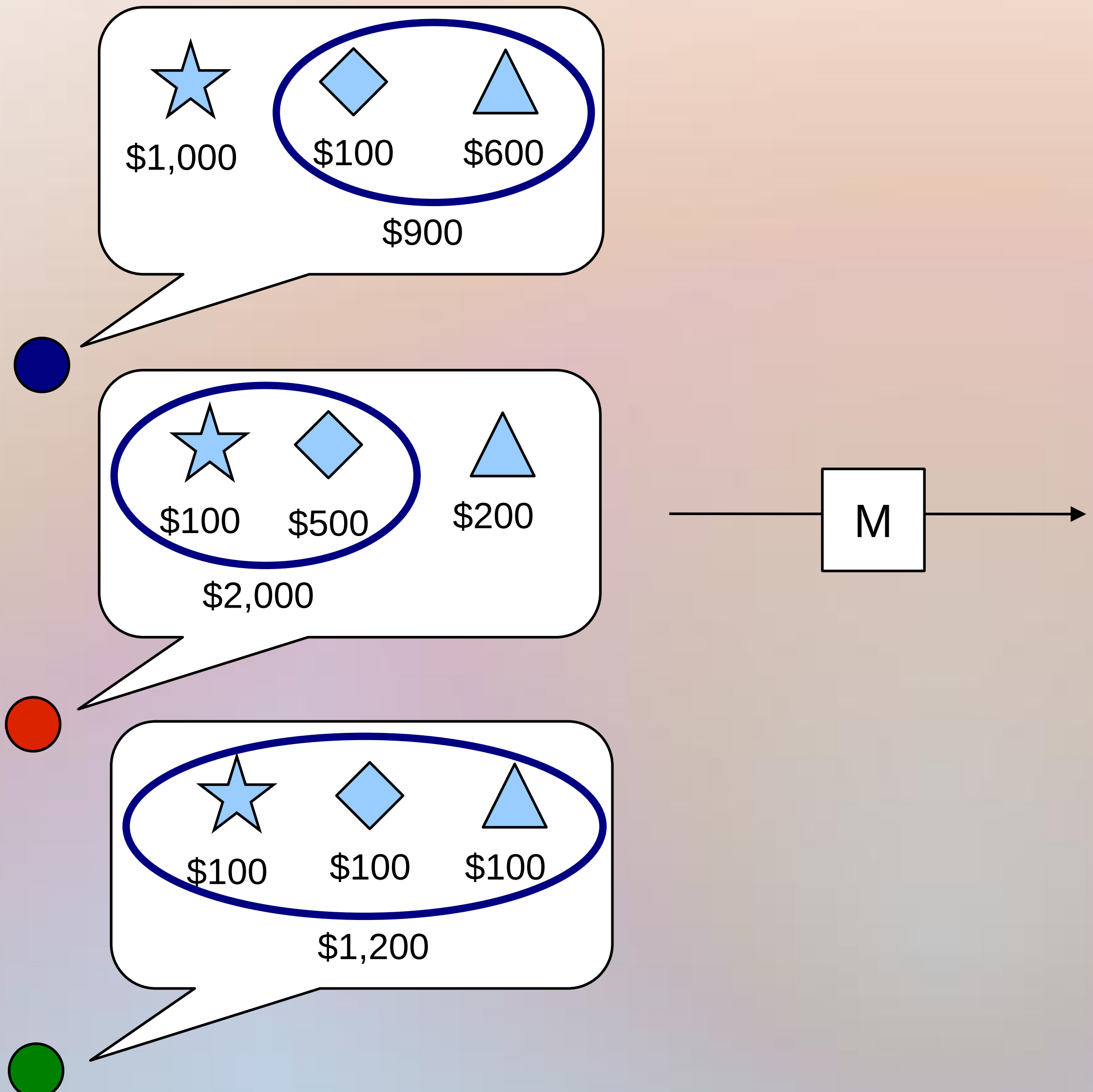
Valuations



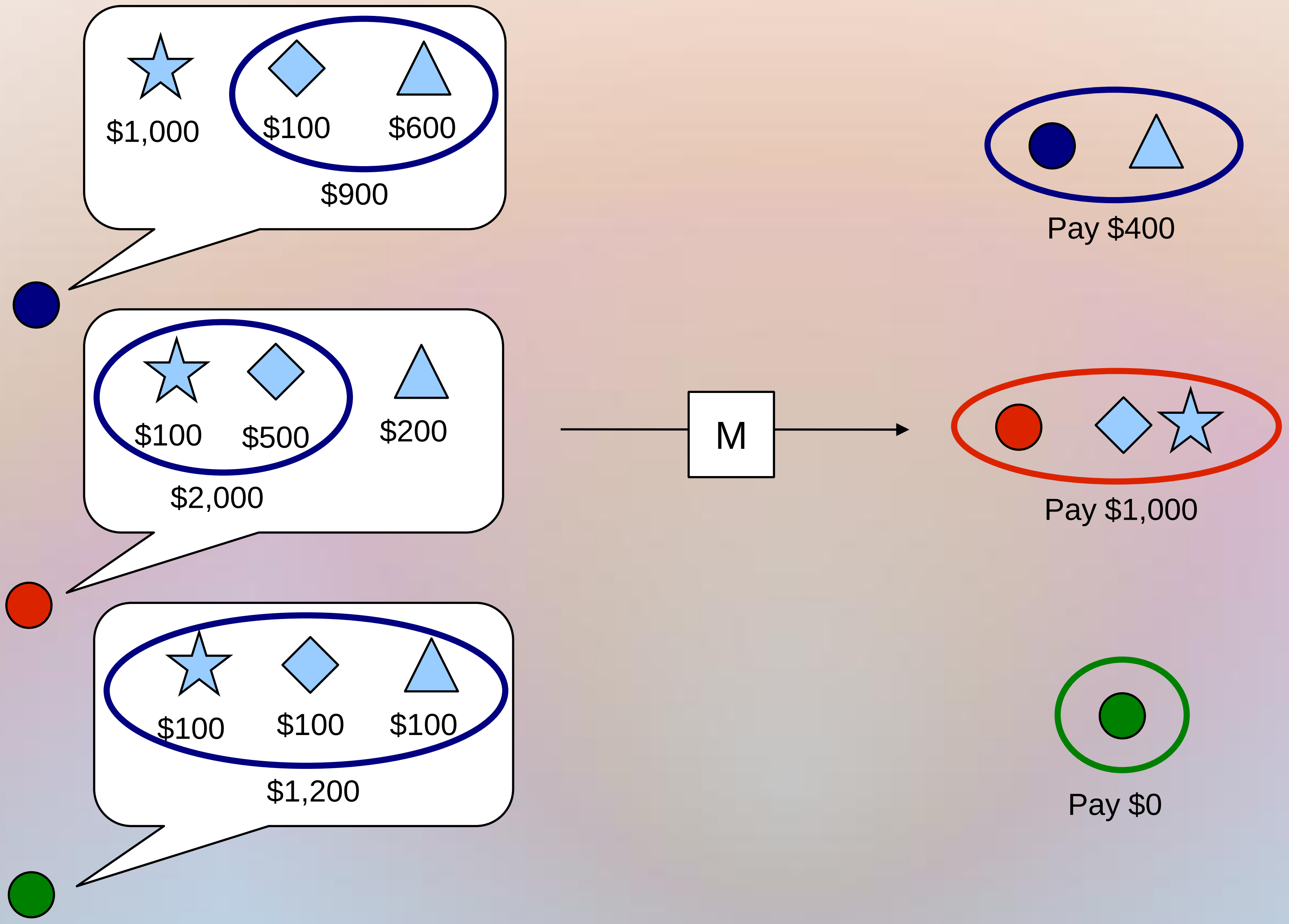
Combinatorial Auctions



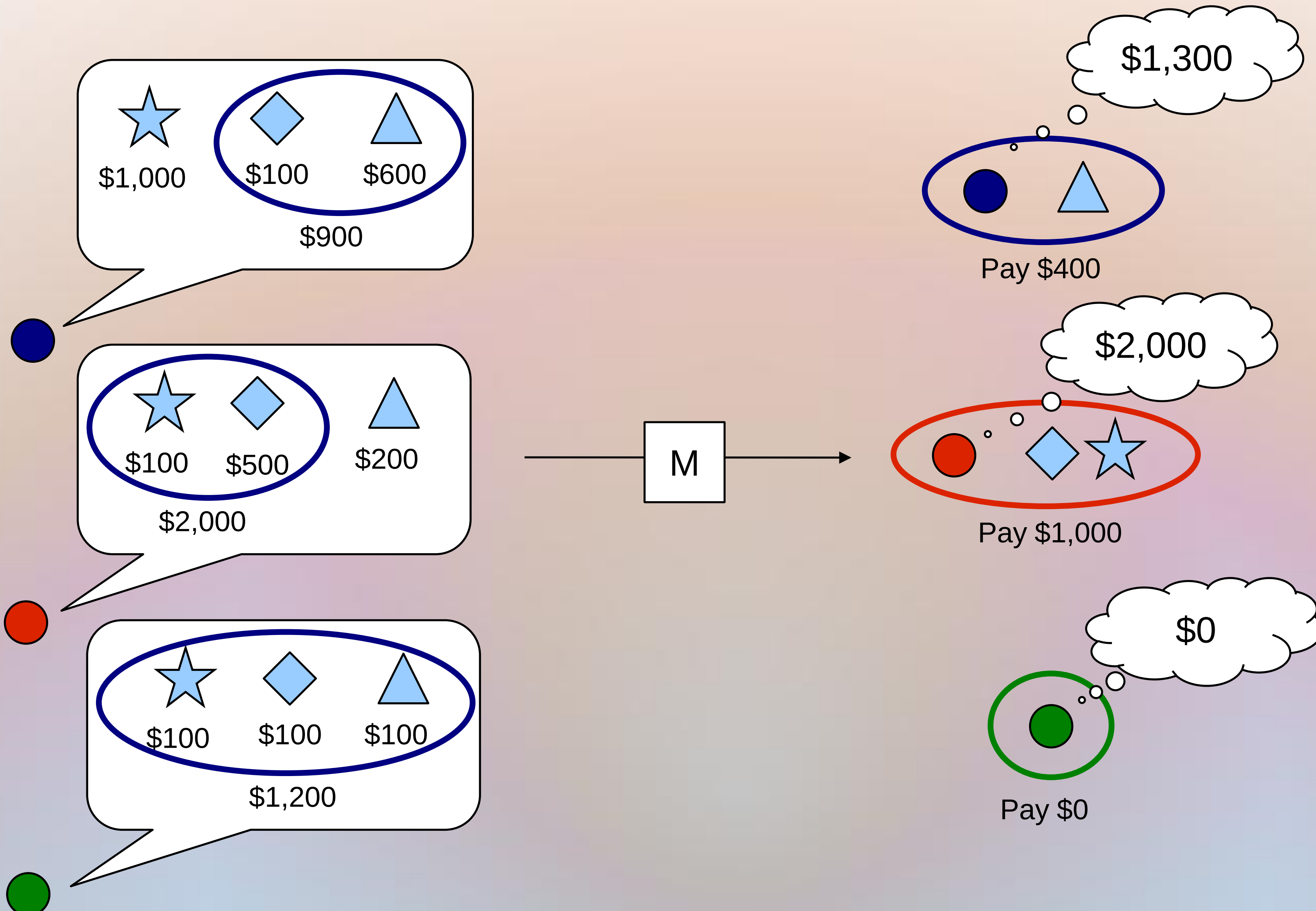
Combinatorial Auctions



Combinatorial Auctions



Combinatorial Auctions



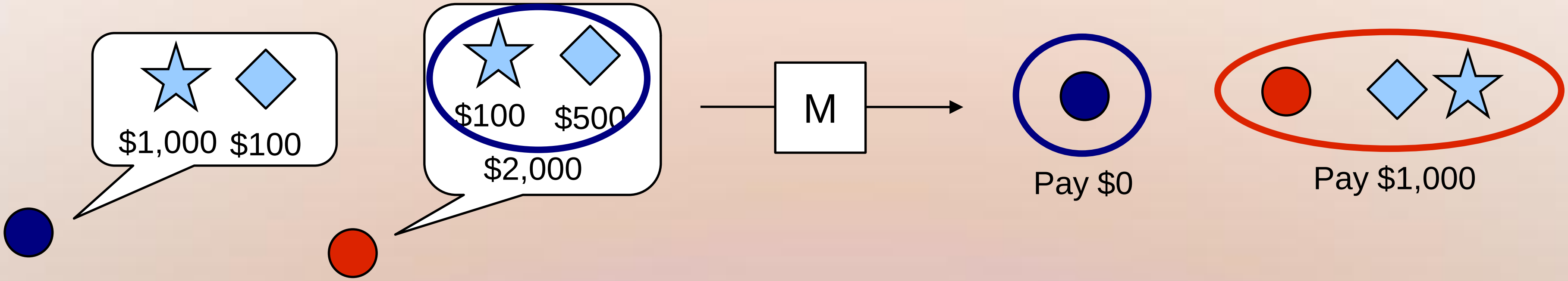
Combinatorial Auctions

- Ignoring game-theoretic concerns, a simple greedy algorithm gives an $O(\sqrt{m})$ approximation to optimal social welfare [Lehmann-O'Callaghan-Shoham 99].
- Can we design a mechanism that obtains an $O(\sqrt{m})$ approximation when input is provided by rational agents?

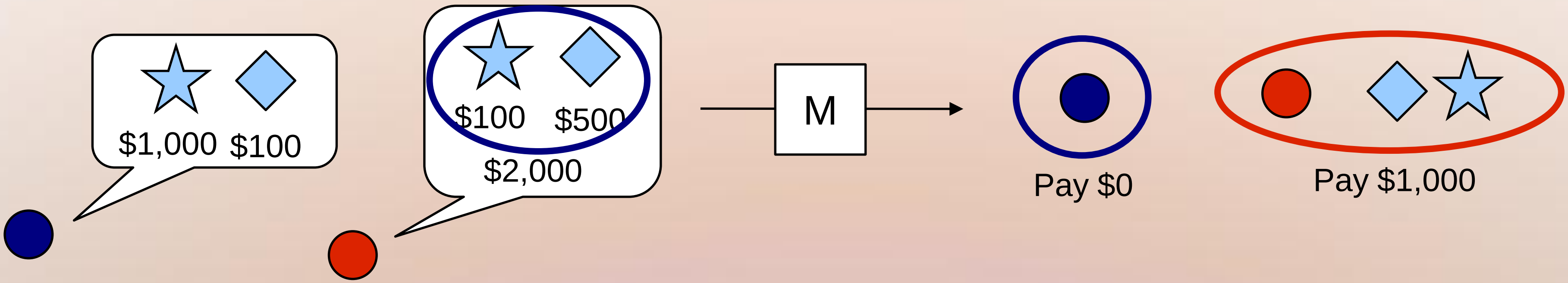
Combinatorial Auctions

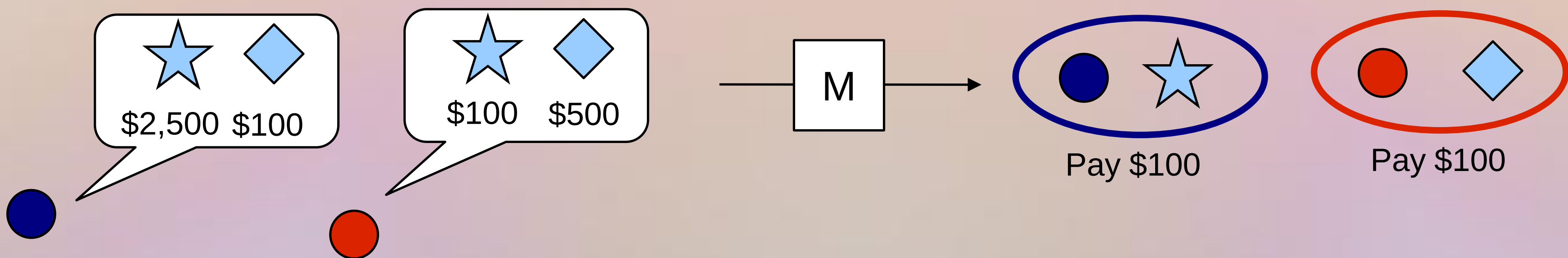
- Ignoring game-theoretic concerns, a simple greedy algorithm gives an $O(\sqrt{m})$ approximation to optimal social welfare [Lehmann-O'Callaghan-Shoham 99].
- Can we design a mechanism that obtains an $O(\sqrt{m})$ approximation when input is provided by rational agents?
- Best-known *truthful* approximation algorithm obtains approximation ratio $O(m/\sqrt{\log m})$ [Holzman Kfir-Dahav Monderer Tennenholz 04].

Repeated Combinatorial Auctions

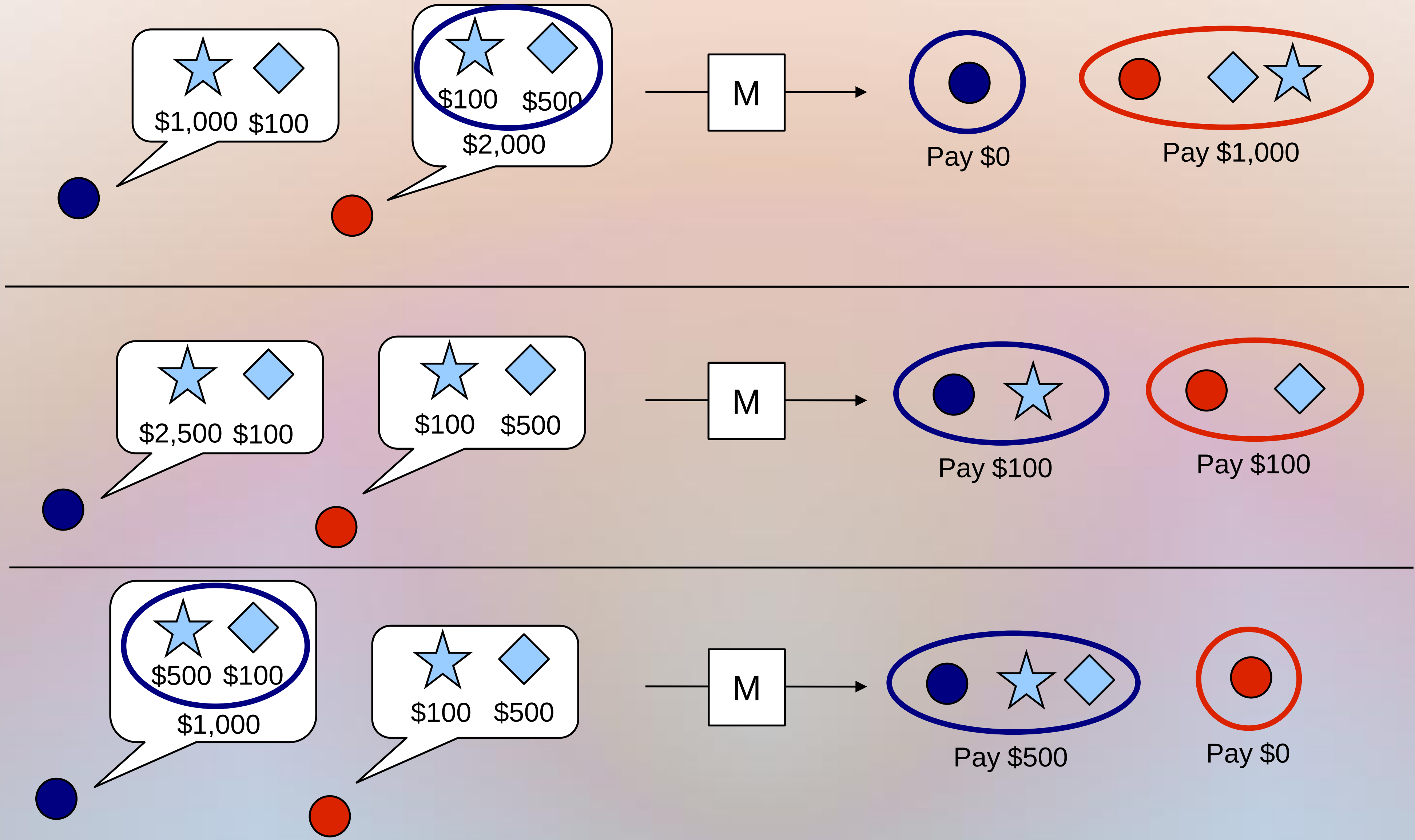


Repeated Combinatorial Auctions





Repeated Combinatorial Auctions



Repeated Combinatorial Auctions

- *History* of agent declarations $D = (d^1, \ldots, d^T)$
- A mechanism M maps each d^t to an allocation $(S_1^t, \ldots, S_n^t)$ and payments $(p_1^t, \ldots, p_n^t)$.
- $t_i(S)$ - agent i's true value for object set S
- Designer's objective: maximize *average social welfare:*

$$SW_{avg}(D) = \frac{1}{T} \sum_t \sum_i t_i(S_i^t)$$

- Agent's objective: maximize *average utility*:

$$u_i(D) = \frac{1}{T} \sum_t \left(t_i(S_i^t) - p_i^t \right)$$

Repeated Combinatorial Auctions

- *History* of agent declarations $D=(d^1,\ldots,d^T)$
- A mechanism M maps each d^t to an allocation $(S_1^t,\ldots,S_n^t)$ and payments $(p_1^t,\ldots,p_n^t)$
- $t_i(S)$ - agent i's true value for object set S
- Designer's objective: maximize *average social welfare:*

$$SW_{avg}(D)=\frac{1}{T}\sum_t\sum_i t_i(S_i^t)$$

- Agent's objective: maximize *average utility*:

$$u_i(D)=\frac{1}{T}\sum_t\left(t_i(S_i^t)-p_i^t\right)$$

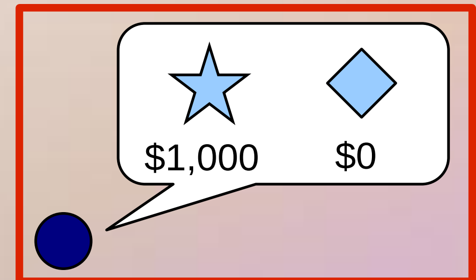
- What bidding strategies do we expect from rational agents?

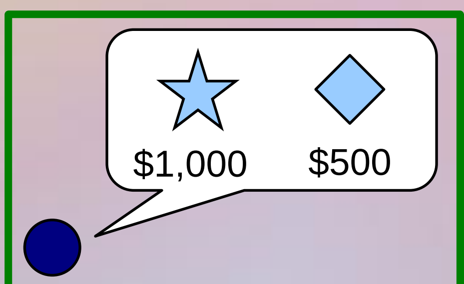
Regret Minimization

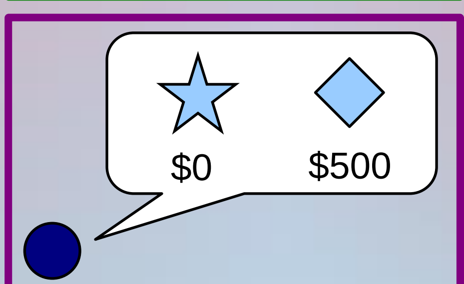
- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i \langle D \rangle$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \rightarrow \infty$.

Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.





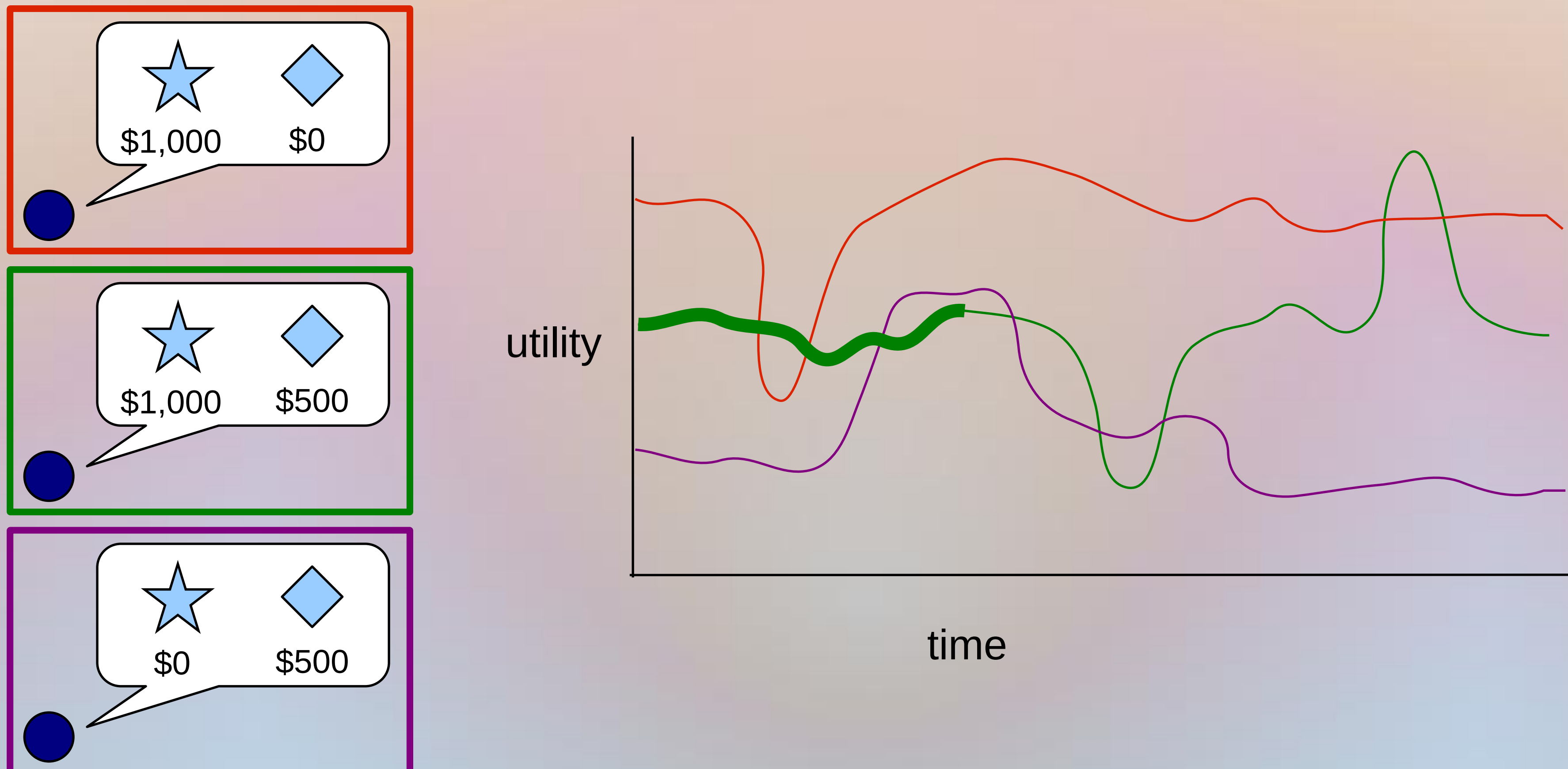


Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.

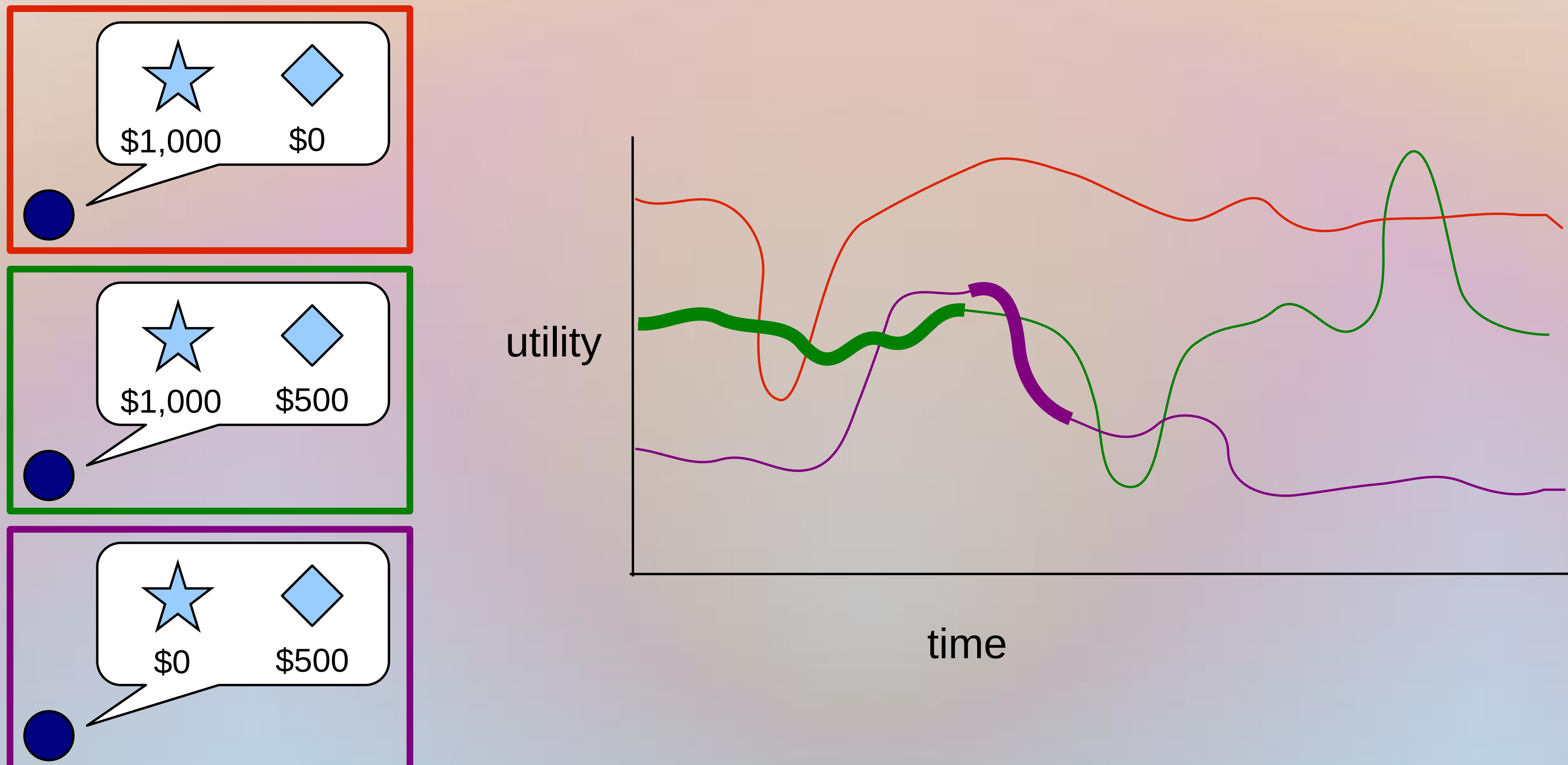
Regret Minimization

- Given history $D = \langle d^1, \dots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.



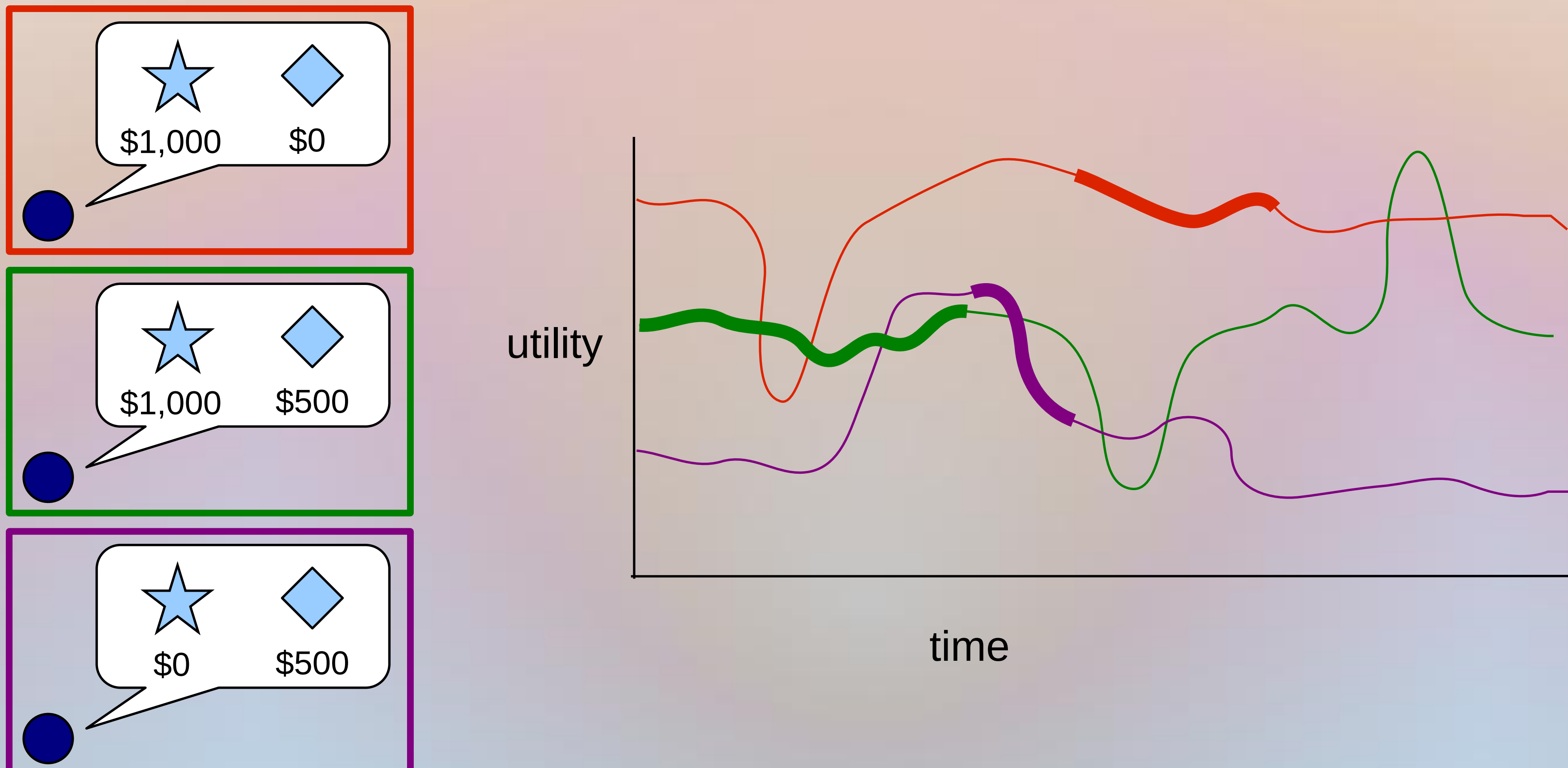
Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.



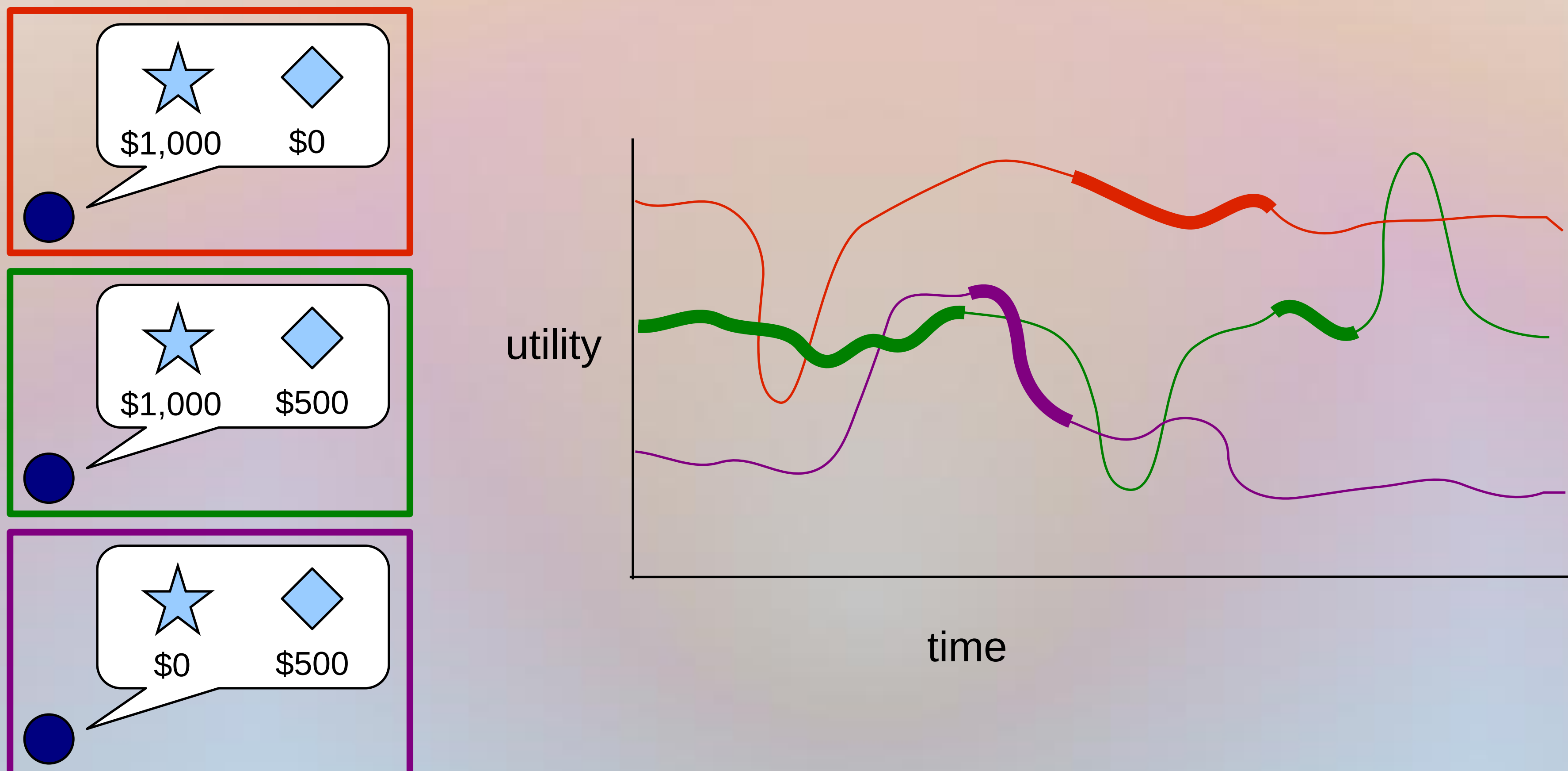
Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.



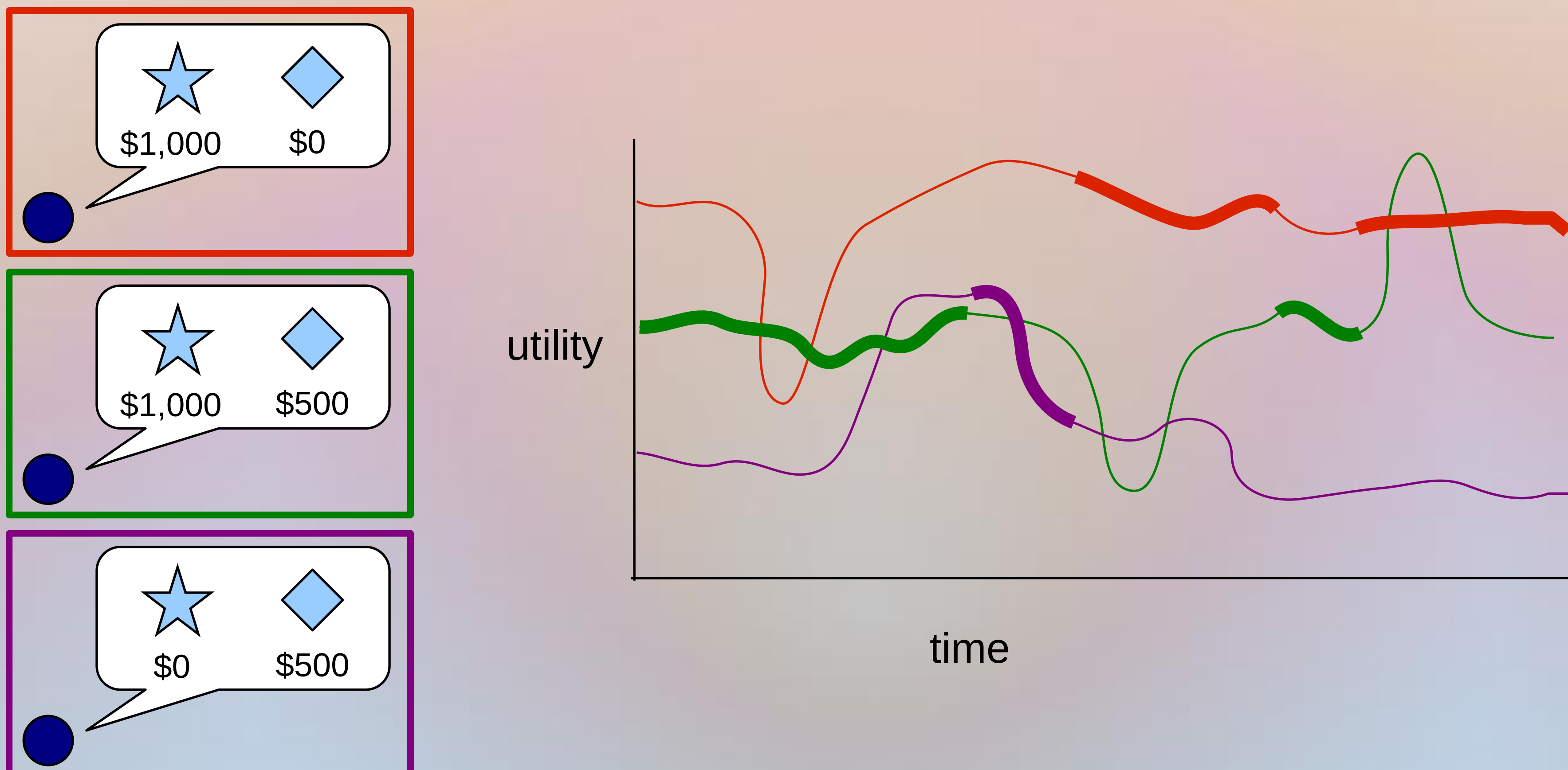
Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.



Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.



Regret Minimization

- Given history $D = \langle d^1, \ldots, d^T \rangle$, the *external regret* for agent i is the difference between $u_i(D)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.
- Simple* algorithms can be used to minimize external regret [Kalai-Vempala 05].

Regret Minimization

- Given history $D=\left(d^1,\ldots,d^T\right)$, the *external regret* for agent i is the difference between $u_i\left(D\right)$ and the best possible average utility obtainable (in hindsight) using the same declaration each round.
- Agent i *minimizes external regret* if his regret tends to 0 as $T\rightarrow\infty$.
- Simple* algorithms can be used to minimize external regret [Kalai-Vempala 05].
- The *price of total anarchy* of mechanism M is

$$max_D \frac{SW_{opt}}{SW_{avg}\left(D\right)}$$

with the max taken over histories in which agents minimize regret [Blum-Hajiaghayi-Ligett-Roth 08].

- Can we implement a mechanism for the CA problem with price of total anarchy $O\left(\sqrt{m}\right)$?

Mechanism M(A)

Let A denote the greedy $O(\sqrt{m})$ approximation algorithm.

Mechanism M(A)

Let A denote the greedy $O(\sqrt{m})$ approximation algorithm.

Mechanism M(A):

```
Input: declaration profile d

   Return allocation A(d)

   Payments: critical prices
```

Mechanism M(A)

Let A denote the greedy $O(\sqrt{m})$ approximation algorithm.

Mechanism M(A):

```
Input: declaration profile d

Return allocation A(d)

Payments: critical prices
```

- Issue: very rich strategy space.
  - Regret minimization does not imply a high social welfare.

Mechanism M(A)

Let A denote the greedy O惜下m下 approximation algorithm.

Mechanism M(A):

```
Input: declaration profile d

   Return allocation A(d)

   Payments: critical prices
```

- Issue: very rich strategy space.
  - Regret minimization does not imply a high social welfare.
  - Minimizing regret is not a feasible task for the agents.

Mechanism M(A)

Let A denote the greedy $O(\sqrt{m})$ approximation algorithm.

Mechanism M(A):

```
Input: declaration profile d
   SIMPLIFY(d)
   Return allocation A(d)
   Payments: critical prices
```

- Issue: very rich strategy space.
  - Regret minimization does not imply a high social welfare.
  - Minimizing regret is not a feasible task for the agents.

Mechanism M(A)

Let A denote the greedy $O(\sqrt{m})$-approximation algorithm.

Mechanism M(A):

```
Input: declaration profile d
    SIMPLIFY(d)
    Return allocation A(d)
    Payments: critical prices
```

- Issue: very rich strategy space.
  - Regret minimization does not imply a high social welfare.
  - Minimizing regret is not a feasible task for the agents.

`SIMPLIFY:`

Mechanism M(A)

```
Input: declaration profile d
   SIMPLIFY(d)
   Return allocation A(d)
   Payments: critical prices
```

- Without loss of generality, each agent bids only on a single *desired set* each round.

Mechanism M(A)

```
Input: declaration profile d
   SIMPLIFY(d)
   Return allocation A(d)
   Payments: critical prices
```

- Without loss of generality, each agent bids only on a single *desired set* each round.
- If agent i bids on set S, then his utility is maximized by declaring his true value, $t_i(S)$, for S.

Mechanism M(A)

```
Input: declaration profile d
   SIMPLIFY(d)
   Return allocation A(d)
   Payments: critical prices
```

- Without loss of generality, each agent bids only on a single *desired set* each round.
- If agent i bids on set S, then his utility is maximized by declaring his true value, $t_i(S)$, for S.
- We conclude that there are only as many *undominated strategies* as there are sets desired by the agent. Standard regret-minimizing algorithms are feasible.

Mechanism M(A)

```
Input: declaration profile d
   SIMPLIFY(d)
   Return allocation A(d)
   Payments: critical prices
```

- Without loss of generality, each agent bids only on a single *desired set* each round.
- If agent i bids on set S, then his utility is maximized by declaring his true value, $t_i(S)$, for S.
- We conclude that there are only as many *undominated strategies* as there are sets desired by the agent. Standard regret-minimizing algorithms are feasible.

Theorem: Mechanism M(A) has price of total anarchy $O(\sqrt{m})$

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- **monotone**: if agent i makes a single-minded declaration for set S and wins it, then he would also win if he increased his declared value for S.

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- **monotone**: if agent i makes a single-minded declaration for set S and wins it, then he would also win if he increased his declared value for S.
- **loser-independent**: The outcome for agent i depends only on d_i and on the outcome that would have occurred if agent i did not participate.

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- **monotone**: if agent i makes a single-minded declaration for set S and wins it, then he would also win if he increased his declared value for S.
- **loser-independent**: The outcome for agent i depends only on d_i and on the outcome that would have occurred if agent i did not participate.

- Includes many greedy-like algorithms.

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- Applies to many greedy algorithms for variants of the CA problem.

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- Applies to many greedy algorithms for variants of the CA problem.

Problem	Feasibility condition	Approximation Ratio
CA Problem	Disjoint sets	$O(\sqrt{m})$

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- Applies to many greedy algorithms for variants of the CA problem.

Problem	**Feasibility condition**	**Approximation Ratio**
CA Problem	Disjoint sets	$O(\sqrt{m})$
s-CA Problem	Disjoint, size at most s	$s+1$

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- Applies to many greedy algorithms for variants of the CA problem.

Problem	Feasibility condition	Approximation Ratio
CA Problem	Disjoint sets	$O(\sqrt{m})$
s-CA Problem	Disjoint, size at most s	$s+1$
Convex Bundles	Disjoint convex areas in plane, max aspect ratio R.	$O(R^{4/3})$ [BB01]

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- Applies to many greedy algorithms for variants of the CA problem.

Problem	**Feasibility condition**	**Approximation Ratio**
CA Problem	Disjoint sets	$O(\sqrt{m})$
s-CA Problem	Disjoint, size at most s	$s+1$
Convex Bundles	Disjoint convex areas in plane, max aspect ratio R.	$O(R^{4/3})$ [BB01]
Unsplittable Flow	Possible to route flow between graph vertices, max capacity B	$O(B\,m^{1/(B-1)})$ [BKV05]

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- Applies to many greedy algorithms for variants of the CA problem.

Problem	Feasibility condition	~~Approximation Ratio~~ Price of Total Anarchy
CA Problem	Disjoint sets	$O(\sqrt{m})$
s-CA Problem	Disjoint, size at most s	$s+2$
Convex Bundles	Disjoint convex areas in plane, max aspect ratio R.	$O(R^{4/3})$ [BB01]
Unsplittable Flow	Possible to route flow between graph vertices, max capacity B	$O(B\, m^{1/(B-1)})$ [BKV05]

Byzantine Players

- Our argument is resilient to the presence of byzantine agents who do not necessarily minimize regret.
- We think of byzantine players as not being savvy enough to understand how to bid “well” in the auction.
- If agents do not over-bid, and some subset of the agents minimize regret, then mechanism M(A) obtains a (c+1) approximation to the optimal social welfare obtainable by the regret-minimizing agents.

Best Response

Best Response

- On each round, an agent is chosen uniformly at random.
- That agent is free to change his declared valuation to the optimal, given the current declarations of the other bidders.

Best Response

- On each round, an agent is chosen uniformly at random.
- That agent is free to change his declared valuation to the optimal, given the current declarations of the other bidders.
- The *price of sinking* of mechanism M is

$$\max \frac{SW_{opt}}{E_D[SW_{avg}(D)]}$$

where the maximum is taken over agent types, and the expectation is over histories corresponding to best-response dynamics (randomized by the agent selection process) [Goemans-Mirrokni-Vetta 05].

Best Response

- On each round, an agent is chosen uniformly at random.
- That agent is free to change his declared valuation to the optimal, given the current declarations of the other bidders.
- The *price of sinking* of mechanism M is

$$\max \frac{SW_{opt}}{E_D[SW_{avg}(D)]}$$

where the maximum is taken over agent types, and the expectation is over histories corresponding to best-response dynamics (randomized by the agent selection process) [Goemans-Mirrokni-Vetta 05].

- Theorem: There is a mechanism for the general CA problem with $O(\sqrt{m})$ price of sinking.

Conclusions

- Applied repeated-game solution concepts to the problem of designing mechanisms for repeated combinatorial auctions.
- There is a general reduction from a broad class of approximation algorithms to approximation mechanisms, under the assumption that agents minimize external regret.
- For the general CA problem, it is possible to obtain an $O(\sqrt{m})$ approximation when agents apply best-response dynamics.

Future Work

- Does the general reduction used for regret-minimizing bidders also yield an O [illegible] approximation for best-response bidders?
- Generalize to broader classes of algorithms.
- Generalize to broader classes of problems.
  - Problems that apply restrictions on the agents' valuation functions, e.g. submodular CAs.

Thank You

Regret Minimization

- Given history $D = (d^1, \ldots, d^T)$, the *external regret* for agent i is

$$\max_{d^*} \left\{ \frac{1}{T} \sum u_i(d^*, d^t_{-i}) \right\} - u_i(D)$$

- Agent i *minimizes external regret* if his regret tends to 0 as $T \to \infty$.
- Simple* algorithms (e.g. follow-the-leader) can be used to minimize external regret [Kalai-Vempala 05].
- The *price of total anarchy* of mechanism M is

$$\max_D \frac{SW_{opt}}{SW_{avg}(D)}$$

with the max taken over histories in which agents minimize regret [Blum-Hajiaghayi-Ligett-Roth 08].

- Can we implement a mechanism for the CA problem with price of total anarchy $O(\sqrt{m})$?

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- **monotone**: if agent i makes a single-minded declaration for set S and wins it, then he would also win if he increased his declared value for S.
- **loser-independent**: if a change in agent i's declaration leads to a change in the algorithm's allocation from $\langle S_1, \ldots, S_n \rangle$ to $\langle S_1', \ldots, S_n' \rangle$, then he must have changed his declared value for either S_i or S_i'.

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- **monotone**: if agent i makes a single-minded declaration for set S and wins it, then he would also win if he increased his declared value for S.
- **loser-independent**: if a change in agent i's declaration leads to a change in the algorithm's allocation from $\langle S_1, \ldots, S_n \rangle$ to $\langle S_1', \ldots, S_n' \rangle$, then he must have changed his declared value for either S_i or S_i'.

- A generalization of max-in-range algorithms.

Mechanism M(A)

- Theorem: If A is a **monotone**, **loser-independent** c-approximation algorithm, then M(A) has price of total anarchy (c+1).

- **monotone**: if agent i makes a single-minded declaration for set S and wins it, then he would also win if he increased his declared value for S.
- **loser-independent**: if a change in agent i's declaration leads to a change in the algorithm's allocation from $\langle S_1, \ldots, S_n \rangle$ to $\langle S_1', \ldots, S_n' \rangle$, then he must have changed his declared value for either S_i or S_i'.

- A generalization of max-in-range algorithms.
- Includes many greedy-like algorithms.