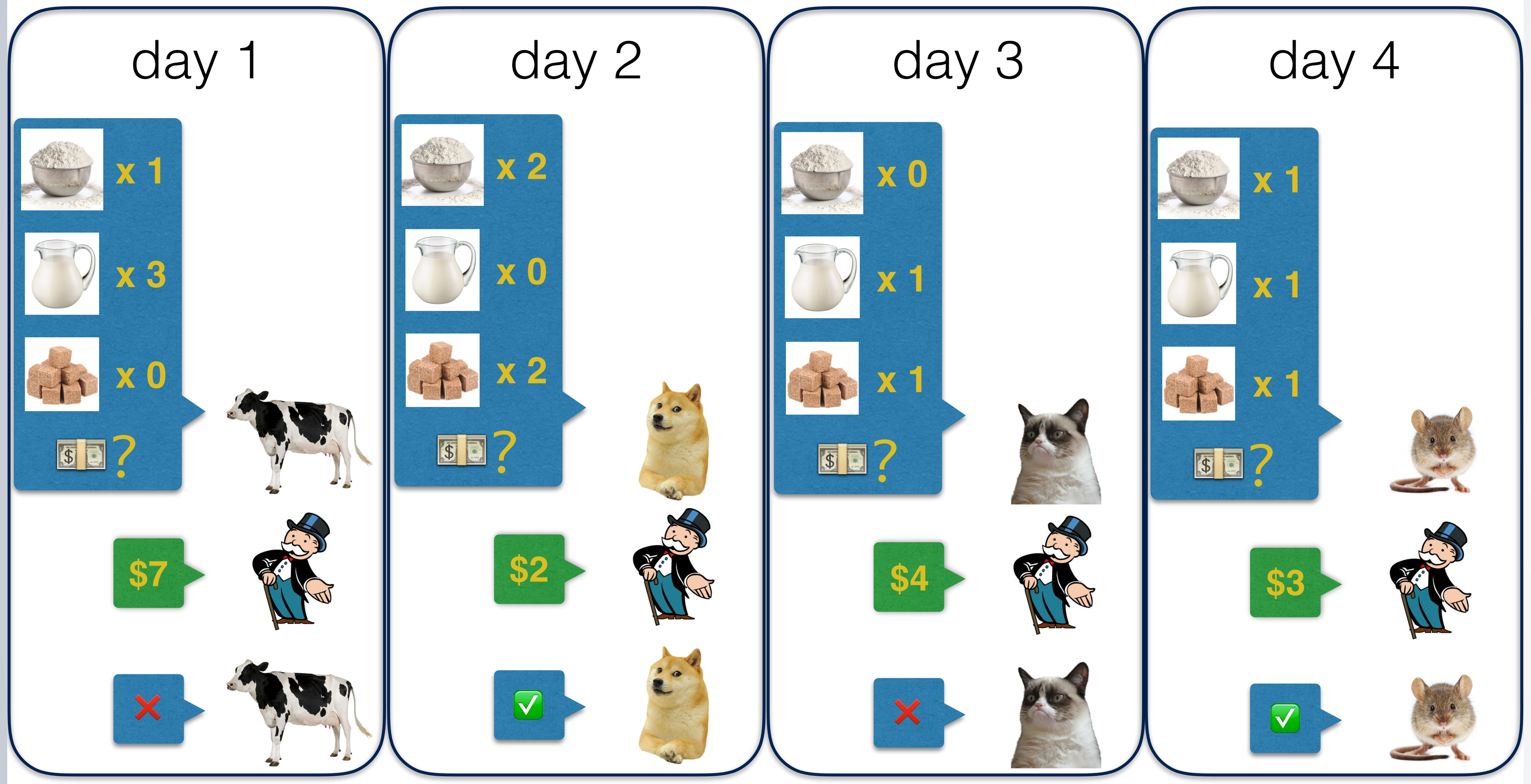
Multi-dimensional Binary Search with Applications to Pricing

Adrian Vladu
MIT

Joint work with Ilan Lobel and Renato Paes Leme

PRICING ITEMS

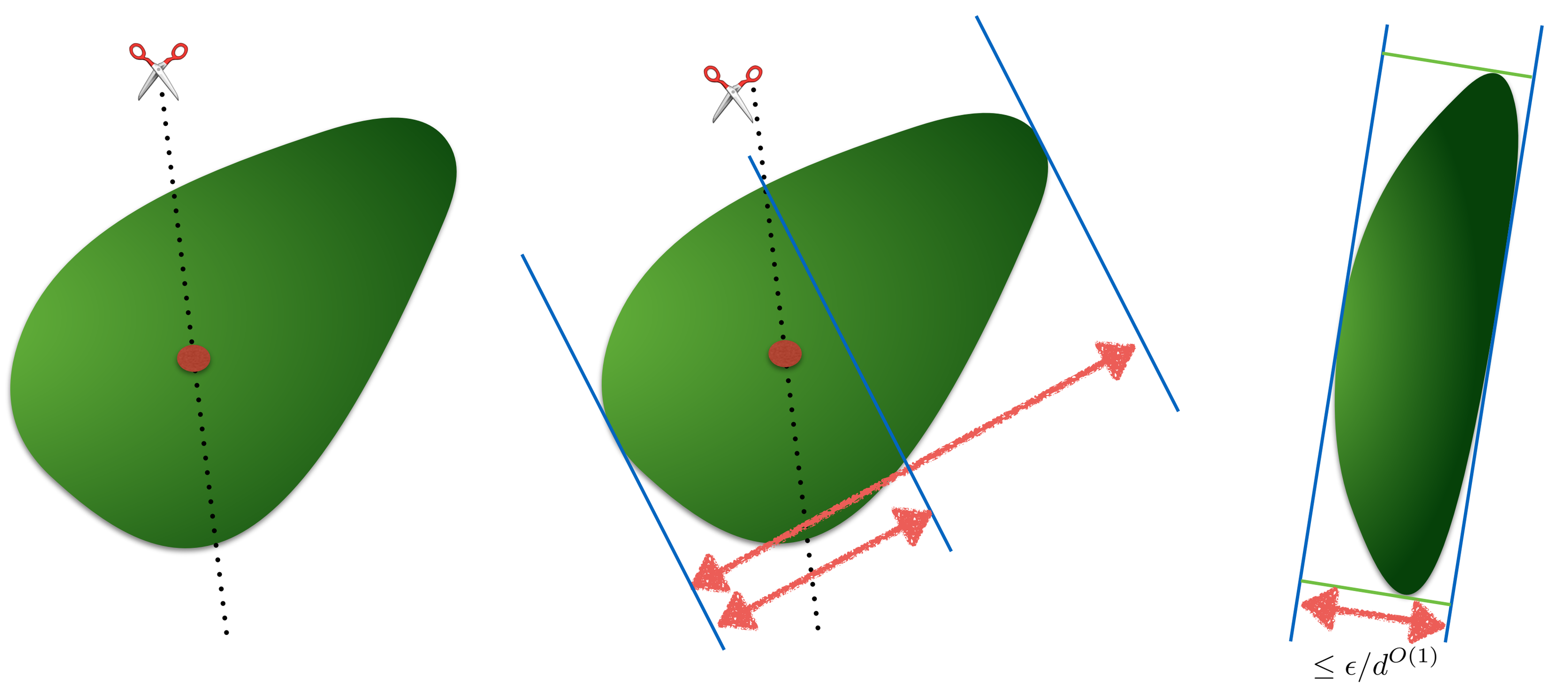


MAIN THEOREM

There exists a strategy for cutting, such that $w(K,u) \le \epsilon$, for all u, after $\tilde{O}(d \log 1/\epsilon)$ iterations.

Set $\epsilon = d \log T / T$, regret is at most $T \cdot \epsilon + \tilde{O}(d \log 1/\epsilon) = \tilde{O}(d \log T)$.

TOOLBOX



1. Grünbaum
2. Directional Grünbaum
3. Cylindrification

BASIC NOTIONS

- **Directional width** $w(K,u) = \max_{\overline{x} \in K} \overline{x}^\top u - \min_{\underline{x} \in K} \underline{x}^\top u$
- **Centroid** $c(x) = \int_K x \, dx$

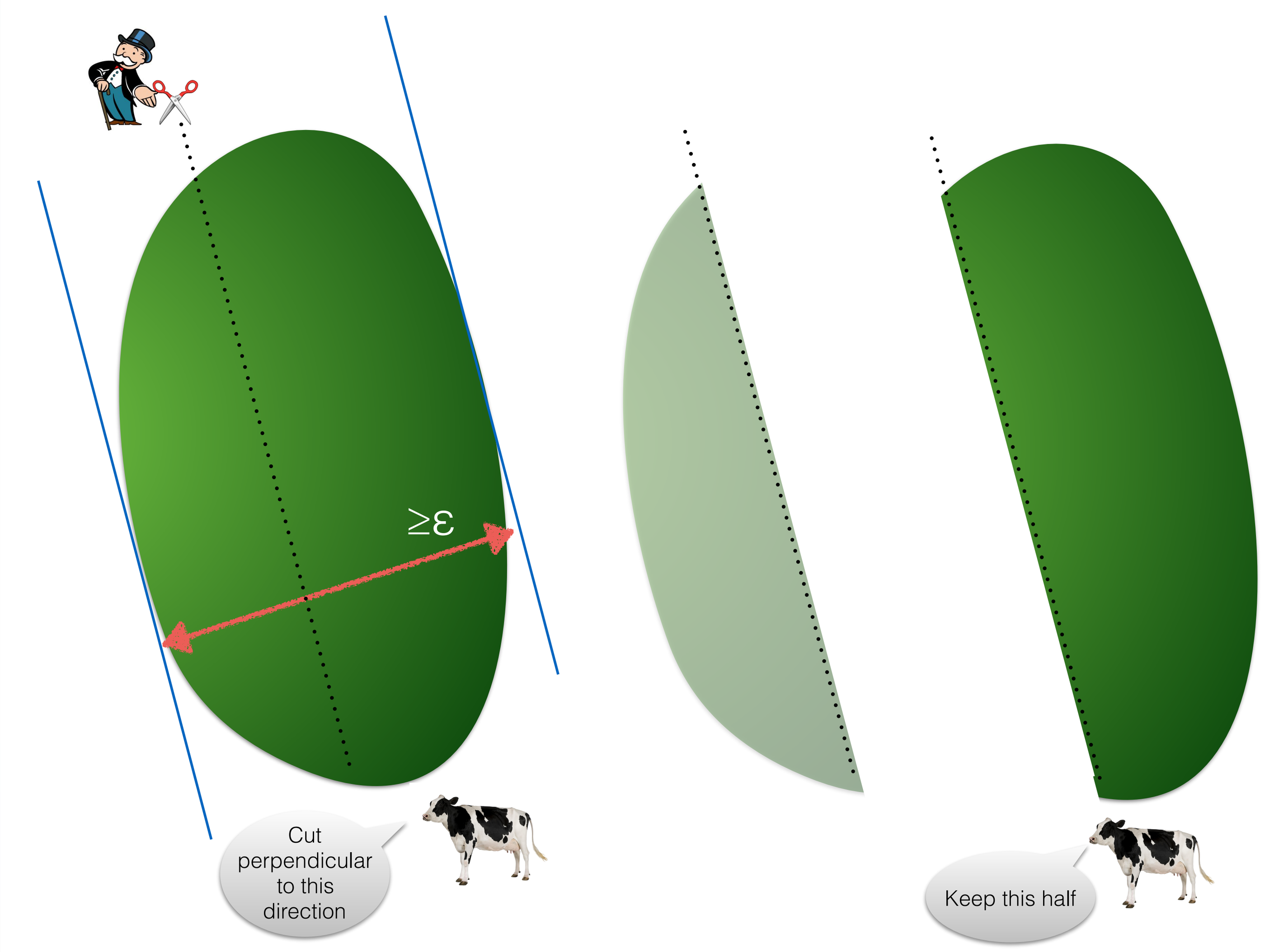
COMPARISON TO PREVIOUS WORK

- **[KL '03]** $d = 1$, regret $O(\log \log T)$
- **[CLP '16]** ellipsoid method, regret $O(d^2 \log T)$
- [this work] convex geometry, regret $\tilde{O}(d \log T)$

OPEN

Obtain regret $\tilde{O}(d \log \log T)$ or prove lower bound

MODEL



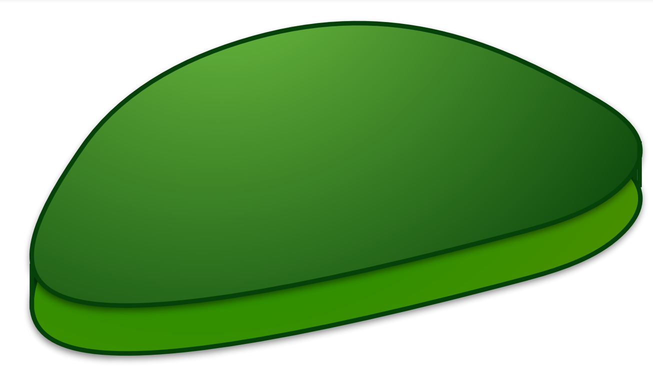
- start with d-dimensional convex body (possible prices for items)
- customer gives vector of item quantities u, $\|u\| = 1$, with the promise that $w(K,u) \ge \epsilon$, asks for price
- player chooses price c, and partitions $K = K_1 \cup K_2$, where $K_1 = \{x \in K : x^\top u \le c\}$ and $K_2 = K \setminus K_1$
- customer picks which half contains the true prices, update K accordingly
- repeat until no valid directions remaining

ALGORITHM AND PROOFS

1. **Grünbaum's Theorem** A cut through $c(K)$ partitions K into pieces with volume ratio at most e.
2. **Directional Grünbaum** (this work) A cut through $c(K)$ partitions $K = K_1 \cup K_2$ such that $w(K_i, u) \ge \frac{w(K,u)}{d+1}$, for all u.
3. **Cylindrification** (this work) If $w(K,u) \ge \delta$, then $Vol(Proj_{\perp u} K) \ge \frac{\delta}{(d+1)^2} Vol(K)$.

---

1. Cut through centroid.
2. When $w(K,u) \le \delta = \epsilon / d^{O(1)}$, cylindrify.
3. Small volume blow-up from $\le d$ cylindrifications, overall need to reduce volume by a factor of $(d/\epsilon)^{O(d)}$.



POLYNOMIAL TIME ALGORITHM

1. Compute approximate centroid, requires almost uniform sampling **[LS '93]**
2. Robust versions of our theorems