1 Comparison of Auction Algorithms

1.1 Nima-McAfee Auction

Algorithm 1: Nima-McAfee Auction

Sort sellers ascending: $v_1^S < v_2^S < \cdots < v_n^S$
buyers descending: $v_1^B > v_2^B > \cdots > v_m^B$
Find largest L and K satisfying: $v_L^B \geq v_L^S$, and $\sum_{i=1}^{L} q_{B_i} \leq \sum_{i=1}^{K} q_{S_i}$
$\gamma \leftarrow \frac{1}{2}(v_{L+1}^S + v_{K+1}^B)$
if $\gamma \in [v_L^S, v_K^B]$ **then**
  $\Theta_{\mathrm{Pr}} \leftarrow \min(\sum_{i=1}^{L} q_{B_i}, \sum_{i=1}^{K} q_{S_i})$
  Set uniform trade price: $p = \gamma$
else
  $\Theta_{Pr} \leftarrow \min(\sum_{i=1}^{L-1} q_{B_i}, \sum_{i=1}^{K-1} q_{S_i})$
  Set buyer price: $p_B = v_L^B$, and seller price: $p = v_L^S$

1.2 SBBA Auction

Algorithm 2: SBBA Auction (Strongly Budget Balanced)

Sort buyers descending: $b_1 \geq b_2 \geq \cdots \geq b_n$
Sort sellers ascending: $s_1 \leq s_2 \leq \cdots \leq s_n$
Find largest k s.t. $b_k \geq s_k$
if $s_{k+1} \leq b_k$ **then**
  Set price $p \leftarrow s_{k+1}$
  Trade all k matched pairs at price p
else
  Set price $p \leftarrow b_k$
  Randomly exclude one seller among cheapest k sellers
  Trade remaining $k-1$ matched pairs at price p

1.3 MUDA Auction

Algorithm 3: MUDA Double Auction (Segal-Halevi et al. [**?**])

Randomly partition buyers into two groups: B_1, B_2
Randomly partition sellers into two groups: S_1, S_2
Compute prices: $p_1 \leftarrow$ market-clearing price for (B_1, S_1)
$p_2 \leftarrow$ market-clearing price: (B_2, S_2)
Cross-match at computed prices:
- Match buyers B_1 with sellers S_2 at price p_2
- Match buyers B_2 with sellers S_1 at price p_1

2 Comparative Market Design Analysis

- **Price Formation:**
  - **Nima-McAfee:** Uses the midpoint between the marginal unsuccessful buyer and seller valuations; sometimes results in a surplus.
  - **SBBA:** Uses a single uniform price, strongly budget balanced; may randomly exclude one seller.
  - **MUDA:** Uses random market partitioning and computes two cross-market prices, ensuring truthful behavior.
- **Budget Balance:**
  - **Nima-McAfee:** Weakly budget-balanced; can leave surplus.
  - **SBBA:** Strongly budget-balanced; no surplus left.
  - **MUDA:** Budget-balanced in expectation due to random partitions.

Truthfulness: All algorithms ensure truthfulness.