Parallel/Distributed Auction Algorithms for the Assignment Problem

Xin Jin

- **Assignment Problem:** n persons and n items. Item j has payoff a_{ij} (could be $-\infty$) to person i:

	item 1	item 2	item 3
person 1	7	9	8
person 2	8	5	7
person 3	1	6	6

  What is the optimal assignment (each person gets exactly one item) that generates the largest total payoff.
- A linear programming problem. >10 variants of algorithms for this problem. Most famous one is Hungarian algorithm. But auction algorithm is easy to parallelize.

- Dealing with the dual problem:

$$\min_{\{p_j\}} \left\{ \sum_{i=1}^{n} \max_{j} \{a_{ij} - p_j\} + \sum_{j=1}^{n} p_j \right\}.$$

- Resembles English auction: people make bids, and the person with the highest bid gets the item.

- Mainly two ways to be parallel/distributed:
  1. Divide items into p partitions and distribute among machines/processors. Each iteration deals with one person.
     - ★ Computation time on each machine: $O(n/p)$
     - ★ Sync cost (merge results) $O(p)$
     - ★ Communication cost: $O(p)$

     More iterations. Requires fewer storage for cluster (n^2/p).
  2. Each iteration involves multiple persons across machines/processors.
     - ★ Computation time on each machine: $O(n)$
     - ★ Sync cost 0 (PRAM) or $O(p)$ (cluster)
     - ★ Communication cost: $O(p)$

     Fewer iterations. Requires more storage for cluster (n^2): one-to-all broadcast of the whole payoff matrix.

- **Results:** For smaller p, first version is faster. For enough p, second is faster for PRAM. Unclear for distributed version.