Sorting Networks Under Restricted Topology

Avah Banerjee, Dana Richards, Igor Shinkar

LSU, GMU, SFU

ibanerjee@lsu.edu

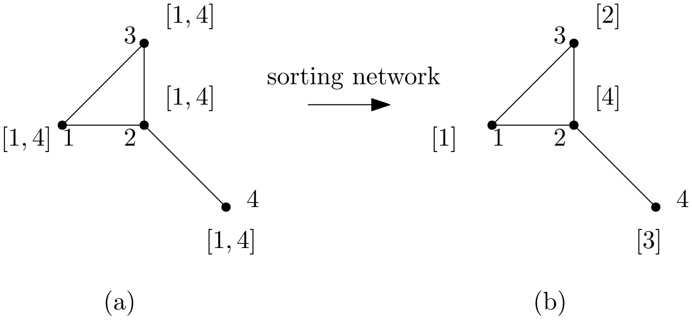

Figure: (a) G with unknown ranks on pebbles. (b) after passing through a sorting network ranks are known. The sorted order is given by the permutation (1)(243).

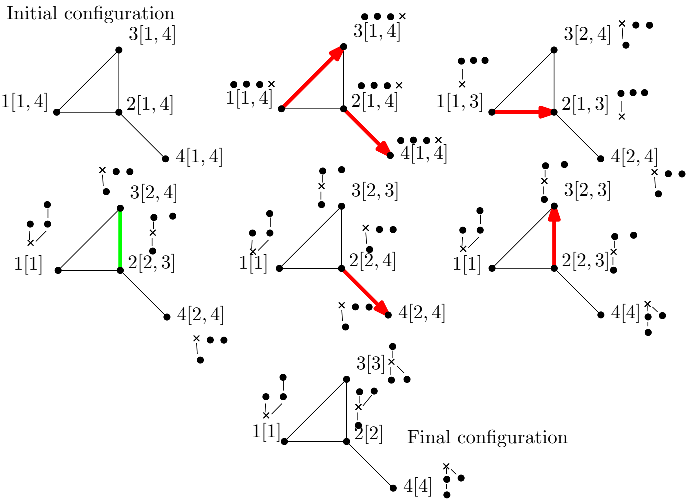

Figure: A Sorting network w.r.t identity permutation for the previous graph.

Definitions: Sorting Network

A sorting network of *depth t* is a triple $\mathcal{S}(G, \mathcal{M}, \pi)$:

1. $G = (V, E)$ is a connected graph with a bijection $\pi : V \to \{1, \ldots, n\}$ specifying the sorted order on the vertices.
2. $\mathcal{M}$ is a sequence of matchings in G where some edges may be directed.
   (direction indicates position of max)
3. After t stages the vertex labeled i contains the pebble whose rank is $\pi(i)$ in the sorted order.
4. This must hold for all $(n!)$ initial arrangements.

Sorting number st(G) of G is the minimum depth sorting network over all permutations.

Definitions: Routing

1. G is a connected labeled graph with a pebble on each vertex.
2. Each pebble has a unique destination specified by the permutation π.
3. We route the pebbles via a sequence of matchings.
4. Pebbles are swapped along matched edges.
5. Routing time is the minimum number of steps necessary to route all pebbles to their destination.
6. After completion of routing all pebbles must be at their destination.

Routing number $rt(G)$ of G is the maximum routing time over all permutations.

Motivation

- Sorting networks have been studied for more than half a century.
- Initial motivation was to design physical networks for sorting.
- If topology of the network is fixed, then comparison-exchanges should be oblivious (precomputed).
- With emergence of FPGAs and GPUs sorting networks are still relevant today.
- Many privacy preserving algorithms employ oblivious permutation networks.

Motivation

- Previous studies on this subject were principally motivated to design optimal or simple sorting networks.
- Topology of these networks were an emerging property of the design.
- In this study we introduce a new paradigm.
- If the topology is fixed what is the best we could do.
- This is a natural question to ask if one wishes to design a *fault-resiliant* sorting network.

Previous Results

Graph	Lower Bound	Upper Bound	Remark
Complete Graph (K_n)	$\log n$	$O(\log n)$	AKS Network
Hypercube (Q_n)	$\Omega(\frac{\log n \log\log n}{\log\log\log n})$	$2^{O(\sqrt{\log\log n})} \log n$	Plaxton et. al
Path (P_n)	$n-1$	n	OETS.
Mesh ($P_n \times P_n$)	$3n - 2\sqrt{n} - 3$	$3n + O(n^{3/4})$	Schnorr et al.
d-dimensional Mesh	$\Omega(dn)$	$O(dn)$	Kunde

Table: Known bounds on the sorting number for various graphs

Trees

- As long as the network is connected we should be able to sort in principle.
- Trees are minimally connected graphs.
- Any fault tolerant network should be able to handle the tree topology.
- We use two parameters to bound the depth of a sorting network, 1) diameter d 2) max degree Δ

Trees

Given a tree with n nodes:

- There is a $O(n \log(n/d))$ depth sorting network.
- There is a $O(\Delta n)$ depth sorting network.
- These bounds are optimal (w.r.t d and Δ) for the Star and Path graphs.

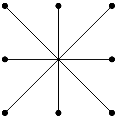

Figure: $st(K_{1,n-1}) = O(n \log n)$ and $st(P_n) = n - 1$.

High Level Idea: $O(n \log(n/d))$-Network

- A tree with diameter d must have a path (subgraph) P_d.
- The path is used as sorting network to sort d pebbles at a time.
- We interleave sorting and routing phases to achieve the desired bound.

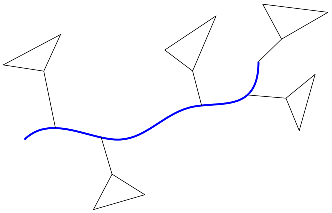

Figure: Path-like parts are easier to sort than Star-like parts.

High Level Idea: $O(\Delta n)$-Network

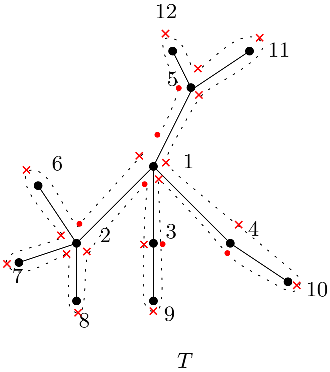

Figure: The contour (dotted line) of the tree T.

High Level Idea: $O(\Delta n)$-Network

- Contour of a tree T is a path P_T ($|P_T| = 2n - 1$) that traverses all edges exactly twice.
- We can simulate an odd-even transposition sorter on this path P_T.
- Pebbles are not consecutive in P_T.
- We use routing to move pebbles adjacent to each other determined by odd or even sorting round.
- Pebbles are located ≤ 3 distance apart and takes ≤ 5 rounds to compare a pair.
- All edges which share a vertex can be matched with $O(\Delta)$ rounds.

Building Networks Using Subgraphs

- H be a subgraph of G.
- If $st(H)$ is small and G is well connected then $st(G)$ is also small.

1. $st(G) = O((n/|H|)\log(n/|H|)(rt_H(G) + st(H)))$
2. $rt_H(G) \leq rt(G) = O((n/\kappa)rt(H'))$.
   $rt_H(G) =$ routing number of G w.r.t destination subgraph H.

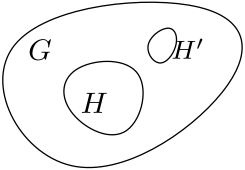

Figure: κ is the connectivity of G, $|H'| = Theta(\kappa)$

More Resuts

1. Highly routable graphs with good concurrency = good sorter.

$$st(G) = O\left((n \log n)\frac{rt(G)}{\nu(G)}\right)$$

   $\nu(G)$ = matching number (a measure of concurrency).

2. If G is the cartesian product of G_1 and G_2 then,

$$st(G) \leq O(\min(\log|V_1|(rt(G)+st(G_2)), \log|V_2|(rt(G)+st(G_1))).$$

The Pyramid Graph

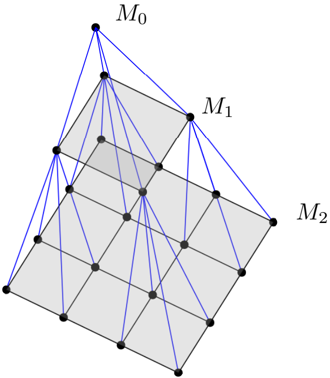

Figure: A pyramid $\triangle_{3,2}$ with 3 layers. Dimension $d = 2$ and Size $N = 21$

Sorting Number of $\triangle = O(dN^{1/d})$

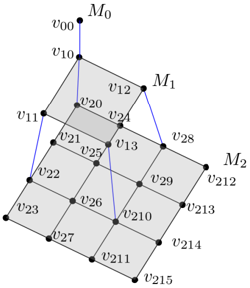

Figure: A multi-grid formed after stripping way some edges from $\triangle_{3,2}$

Use vertical paths of length k to move pebbles up to level k (from the base).

Questions?