Parallel $(1 + \epsilon)$-Approximate Multi-Commodity Mincost Flow in Almost Optimal Depth and Work

To appear at FOCS'25

Yonggang Jiang (Max Planck Institute for Informatics)

Collaborators:

Bernhard Haeupler (ETH & INSAIT)

Yaowei Long, Thatchaphol Saranurak (Umich)

Shengzhe Wang (ETH)

Minimum Cost Flow

- Input: a directed graph with edge capacities and costs, two nodes s, t
  - For this talk: capacities and costs are non-negative integers
- Output: Max flow with Min cost.
  - Max flow: maximum (fractional) number of s-t flow path that respects capacity
  - Min cost: minimum summation of cost (length) of all the flow path
- Special cases:
  - Max flow
  - SSSP
  - Reachability
  - …

Parallel Computing

- PRAM: shared memory parallel model
  - Work: total number of unit operations
  - Depth: longest dependency chain of the algorithm

- Work-efficient: $\text{Work} = \tilde{O}(\text{best sequential running time})$
- Highly parallelizable: Depth$= \tilde{O}(1)$ and work-efficient
  - For this talk: Depth$= n^{o(1)}$ and work-efficient

What problems are highly parallelizable?

What problems are highly parallelizable?

- Reachability? ✖
  - Decades of attempts … stuck at $n^{0.5+o(1)}$ depth [JLS'19]
  - Lower bound of $n^{1/4}$ [BH'23] for shortcut type algorithms
  - Seems far from the answer …
- SSSP ✖
- Max flow ✖
- Min-cost flow ✖

Give up?

What problems are highly parallelizable?

- Reachability? ✖
  - Decades of attempts … stuck at $n^{0.5+o(1)}$ depth [JLS'19]
  - Lower bound of $n^{1/4}$ [BH'23] for shortcut type algorithms
  - Seems far from the answer …
- $(1+\epsilon)$- Approximate SSSP ✖
- $(1+\epsilon)$- Approximate Max flow ✖
- $(1+\epsilon)$- Approximate Min-cost flow ✖

Implies Reachability

Give up?

What problems are highly parallelizable?

- Undirected Reachability: $\tilde{O}(1)$ depth!
- Undirected SSSP ✖
- Undirected Max flow ✖
- Undirected Min-cost flow ✖

Implies **directed** Reachability

Give up?

What problems are highly parallelizable?

- Undirected $(1+\epsilon)$-Approximate SSSP
  - [Cohen, STOC'94] $n^{o(1)}$ depth via *hopset ("distance shortcut")*
  - [Li, STOC'20] $\tilde{O}(1)$ depth via *oblivious routing*
- Undirected $(1+\epsilon)$-Approximate Max flow
  - [Sherman, FOCS'13] First $\tilde{O}(m)$ *work sequential (congestion approximator + MWU)*
  - [Agarwal, Khanna, Li, Patil, Wang, White, Zhong, SODA'24] $\tilde{O}(1)$ depth following Sherman's approach
  - Sherman's approach does not work for Vertex Capacitary or Min-Cost
- Undirected $(1+\epsilon)$-Approximate Min-cost flow (with vertex capacity & cost)
  - [Bernstein, Gutenberg, Saranurak, FOCS'21] First $m^{1+o(1)}$ *work sequential (decremental SSSP + MWU)*
  - [Chen, Kyng, Liu, Peng, Gutenberg, Sachdeva, FOCS'22] $m^{1+o(1)}$ work (IPM for m iterations)
  - **Our result:** $m^{1+o(1)}$ work, $n^{o(1)}$ depth via "***LC-flow shortcut***"

* *Approximate means* $(1-\epsilon)$ *flow with* $(1+\epsilon)$ *cost for maximum flow*

Overview

1. Low-step $(1+\epsilon)$-Approximate min-cost flow in $m^{1+o(1)}$ work and $n^{o(1)}$ depth
   [Haeupler, Hershkowitz, Saranurak, STOC'23]
   - There exists an approximate flow and that every flow path has $n^{o(1)}$ edges.
   - Main techniques: flow path LP with $\exp\left(n^{o(1)}\right)$ constraints + MWU in $n^{o(1)}$ rounds

2. Add "LC-flow shortcut" edges H to the graph G so that
   - Every flow in G *maps* to a low-step flow in $G \cup H$
   - Every flow in $G \cup H$ *maps* to a flow in G
   - Flow *mapping* has
     - Cost slack $(1+\epsilon)$
     - Flow value (congestion) slack $n^{o(1)}$

Our focus

3. We get: min-cost flow with $(1+\epsilon)$-cost approximation and $n^{o(1)}$-flow approximation
   - Use MWU to boost the flow approximation to $(1-\epsilon)$

LC-Flow shortcut: Plan

1. A "distance shortcut" preserving $(1+\epsilon)$ distance (no congestion) [Cohen, STOC'94]
   - Low diameter decomposition

2. A "congestion shortcut" preserving only $n^{o(1)}$ congestion (no cost)
   - Expander decomposition hierarchy

3. A "LC-flow shortcut" preserving both $n^{o(1)}$ congestion and $(1+\epsilon)$ distance
   - Combining the ideas of 1 and 2
   - Length-constraint expander decomposition hierarchy

Distance Shortcut [Cohen, STOC'94]

- LDD: Low-diameter decomposition (with a parameter d):
  - Partition vertex set into *clusters* with diameter at most d
  - Any path *(we care about)* of length d crosses at most $\tilde{O}(1)$ clusters
    - To be precise: each edges is crossing clusters with probability $\tilde{O}(\frac{1}{d})$

- A simple distance shortcut with slack $\tilde{O}(1)$
  - For every scale of d: compute LDD with parameter d, add a d-star to each cluster

- It is possible to boosting distance slack to $(1 + \epsilon)$

Congestion Shortcut

- Expander decomposition (with a parameter $\phi = 1/n^{o(1)}$):
  - Partition vertex sets into *clusters* with expansion at least ϕ *(any degree-respecting demand can be routed with congestion $1/\phi$ satisfying)*
  - At most $\tilde{O}(\phi \cdot m)$ crossing cluster edges

- Add a star to each cluster (capacity equals to degree): congestion slack $n^{o(1)}$

- For crossing cluster edges: run *terminal expander decomposition*
  - Repeat $O(\log_{\frac{1}{\phi}} n)$ layers

Combining Distance and Congestion: LC-Flow Shortcut

- How to define a cluster? (Think of combining Low-diameter and Expander)
  - Every degree-respecting demand can be routed by a flow with congestion ϕ and *average cost* d

- What is the guarantee of decomposition?
  - $\tilde{O}(m\phi)$ edges must cross clusters, and then
  - Each edge becomes crossing with probability $\tilde{O}(\frac{1}{d})$

- Making this intuition to be continuous: “crossing” -> “increasing length”
  - Length-constraint expander decomposition [Haeupler, Raecke, Ghaffari STOC’22]

- Add stars with (1) degree as capacity (2) d as cost -> LC-Flow shortcut

Summary and Open Problems

- Undirected $(1+\epsilon)$-Approximate Min-cost flow (with vertex capacity & cost)
  - $m^{1+o(1)}$ work $n^{o(1)}$ depth

Open problems:

- How to improve to $\tilde{O}(m)$ work and $\tilde{O}(1)$ depth? (Even sequential is unknow)
  - Shortcut based approach cannot work, $(1+\epsilon)$-hopset has lower bound $n^{o(1)}$
  - [Li, STOC'20] circumvent the barrier for SSSP using *oblivious routing*

- Removing "undirected" or "approximation"?
  - Requires solving reachability (shortcut upper / lower bound)
  - Match the performance of reachability ($\tilde{O}(m)$ work $n^{0.5+o(1)}$ depth)
    - Linear size LC-flow shortcut for directed graph with depth $n^{1-\epsilon}$?
    - Min-cost flow in almost linear work and h depth for h-length bound flow path?