Minimum Cost Flow in the CONGEST Model

Tijn de Vos

Department of Computer Science
University of Salzburg

European Research Council
Established by the European Commission

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 947702) and is supported by the Austrian Science Fund (FWF): P 32863-N.

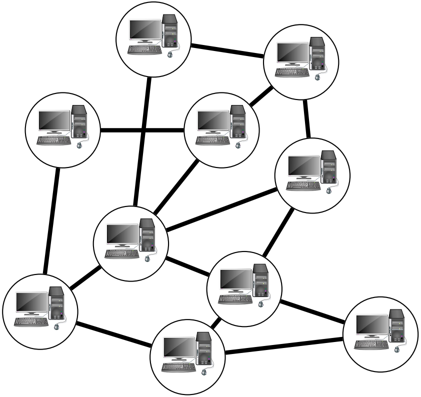
Der Wissenschaftsfonds.

One-Line Result

Theorem (Informal)

Min Cost Flow with $\tilde{O}(\sqrt{n})$ Laplacian solves in the CONGEST model.

The CONGEST Model



- $G = (V, E)$, $|V| = n$, $|E| = m$
- Communication over edges in synchronous rounds
- Bandwidth $O(\log n)$ bits per edge

Flow in the CONGEST Model

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m_{\geq 0}$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

- Exact unless stated otherwise
- Generalizes Max Flow and Negative Weight Shortest Path

Flow in the CONGEST Model

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m_{\geq 0}$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

- Exact unless stated otherwise
- Generalizes Max Flow and Negative Weight Shortest Path
- Lower bound: $\tilde{\Omega}(\sqrt{n} + D)$

Flow in the CONGEST Model

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m_{\geq 0}$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

- Exact unless stated otherwise
- Generalizes Max Flow and Negative Weight Shortest Path
- Lower bound: $\tilde{\Omega}(\sqrt{n} + D)$
- Recently first exact CONGEST algorithm: round complexity $> m^{3/7} n^{1/2}$ [FGLPSY '21]

Laplacian Paradigm

- Laplacian systems
- Spectral sparsifiers
- Electrical flow
- Effective resistance
- Expander decompositions
- Continuous optimization
- Interior-point methods
- Gradient descent
- Preconditioning
- ...

Laplacian Paradigm and Distributed Computing

Observation

Laplacian paradigm often yields inherently parallelizable algorithms

Laplacian Paradigm and Distributed Computing

Observation

Laplacian paradigm often yields inherently parallelizable algorithms

Basic operation:

- Vector $\mathbf{x}$: each node represents a coordinate
- Matrix $\mathbf{A}$: each edge represents a non-zero entry
- Matrix-vector multiplication $\mathbf{Ax}$: one round

Laplacian Paradigm and Distributed Computing

Observation

Laplacian paradigm often yields inherently parallelizable algorithms

Basic operation:

- Vector $\mathbf{x}$: each node represents a coordinate
- Matrix $\mathbf{A}$: each edge represents a non-zero entry
- Matrix-vector multiplication $\mathbf{Ax}$: one round

State of the art for (approximate) single-source shortest path, maximum flow, minimum-cost flow:
[GKKLPS '15] [BFKL '17] [Z '21] [AGL '21] [ZGYHS '22] [RGHZL '22] [F**V** '22] [F**V** '23]

Laplacian Systems

Goal

Solve linear system $\mathbf{L}\mathbf{x} = \mathbf{b}$ such that $\mathbf{L}$ is a Laplacian matrix.

Laplacian Systems

Goal

Solve linear system $\mathbf{Lx} = \mathbf{b}$ such that $\mathbf{L}$ is a Laplacian matrix.

Definition

The Laplacian matrix $\mathbf{L}_G$ is defined by $\mathbf{L}_G := \mathbf{D} - \mathbf{A}$, i.e.,

$$(\mathbf{L}_G)_{u,v} = \begin{cases} \sum_{(u,v') \in E} w_{u,v'} & \text{if } u = v, \\ -w_{u,v} & \text{otherwise.} \end{cases}$$

Laplacian Systems

Goal

Solve linear system $\mathbf{L}\mathbf{x} = \mathbf{b}$ such that $\mathbf{L}$ is a Laplacian matrix.

Definition

The Laplacian matrix $\mathbf{L}_G$ is defined by $\mathbf{L}_G := \mathbf{D} - \mathbf{A}$, i.e.,

$$(\mathbf{L}_G)_{u,v} = \begin{cases} \sum_{(u,v')\in E} w_{u,v'} & \text{if } u = v, \\ -w_{u,v} & \text{otherwise.} \end{cases}$$

High-precision solver: Approximation $\mathbf{x}$ of solution $\mathbf{x}^*$ s.t.

$$\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{L}(G)} \leq \epsilon \|\mathbf{b}\|_{\mathbf{L}(G)} .$$

Laplacian Systems

Goal

Solve linear system $\mathbf{Lx} = \mathbf{b}$ such that $\mathbf{L}$ is a Laplacian matrix.

Definition

The Laplacian matrix $\mathbf{L}_G$ is defined by $\mathbf{L}_G := \mathbf{D} - \mathbf{A}$, i.e.,

$$(\mathbf{L}_G)_{u,v} = \begin{cases} \sum_{(u,v')\in E} w_{u,v'} & \text{if } u = v, \\ -w_{u,v} & \text{otherwise.} \end{cases}$$

High-precision solver: Approximation $\mathbf{x}$ of solution $\mathbf{x}^*$ s.t.

$$\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{L}(G)} \leq \epsilon \|\mathbf{b}\|_{\mathbf{L}(G)} .$$

Round complexity $T_{\text{Laplacian}}$

- $T_{\text{Laplacian}} = n^{o(1)}(\sqrt{n} + D)$ in general [FGLPSY '21]
- $T_{\text{Laplacian}} = n^{o(1)}D$ in planar graphs, expander graphs, $n^{o(1)}$-genus graphs, $n^{o(1)}$-treewidth graphs, and excluded-minor graphs [ALHZG '22]

Flow in the CONGEST Model: Our Result

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

Flow in the CONGEST Model: Our Result

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \leq M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

Flow in the CONGEST Model: Our Result

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \le M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

- $T_{\text{Laplacian}} = n^{o(1)}(\sqrt{n} + D)$ in general [FGLPSY '21]
- $T_{\text{Laplacian}} = n^{o(1)} D$ in planar graphs, expander graphs, $n^{o(1)}$-genus graphs, $n^{o(1)}$-treewidth graphs, and excluded-minor graphs [ALHZG '22]

Flow in the CONGEST Model: Our Result

Definition

$G = (V, E)$ directed, capacities $c \in \mathbb{Z}^m$, costs $q \in \mathbb{Z}^m$, source and target $s, t \in V$. The minimum cost (maximum) flow problem is to find the $s - t$ flow $f \in \mathbb{R}^m$ of minimum cost, among all flows of maximum value.

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \leq M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

- $T_{\text{Laplacian}} = n^{o(1)}(\sqrt{n} + D)$ in general [FGLPSY '21]
- $T_{\text{Laplacian}} = n^{o(1)} D$ in planar graphs, expander graphs, $n^{o(1)}$-genus graphs, $n^{o(1)}$-treewidth graphs, and excluded-minor graphs [ALHZG '22]
- Also: Max Flow and Negative Weight Shortest Path
- Obtain result by solving a more general class of LPs

Flow in the CONGEST Model: Previous Work

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \leq M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

- Unit Capacity Min-Cost Flow: $m^{3/7+o(1)}(\sqrt{n}D^{1/4} + D)$ rounds [FGLPSY '21]

Flow in the CONGEST Model: Previous Work

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \leq M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

- Unit Capacity Min-Cost Flow: $m^{3/7+o(1)}(\sqrt{n}D^{1/4} + D)$ rounds [FGLPSY '21]
- Max Flow: $\tilde{O}(m^{3/7}M^{1/7}(\sqrt{n}D^{1/4} + D) + \sqrt{m})$ rounds [FGLPSY '21]

Flow in the CONGEST Model: Previous Work

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \leq M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

- Unit Capacity Min-Cost Flow: $m^{3/7+o(1)}(\sqrt{n}D^{1/4} + D)$ rounds [FGLPSY '21]
- Max Flow: $\tilde{O}(m^{3/7} M^{1/7}(\sqrt{n}D^{1/4} + D) + \sqrt{m})$ rounds [FGLPSY '21]
- Undirected Approximate Unit Capacity Min-Cost Flow: $\tilde{O}(n/\epsilon^2)$ rounds [BFKL '21]

Flow in the CONGEST Model: Previous Work

Theorem

$G = (V, E)$ directed, $||c||_\infty, ||q||_\infty \leq M$. We can solve the minimum cost maximum flow problem in $\tilde{O}(\sqrt{n} T_{\text{Laplacian}} \log^3 M)$ rounds in the CONGEST model.

- Unit Capacity Min-Cost Flow: $m^{3/7+o(1)}(\sqrt{n}D^{1/4} + D)$ rounds [FGLPSY '21]
- Max Flow: $\tilde{O}(m^{3/7} M^{1/7}(\sqrt{n}D^{1/4} + D) + \sqrt{m})$ rounds [FGLPSY '21]
- Undirected Approximate Unit Capacity Min-Cost Flow: $\tilde{O}(n/\epsilon^2)$ rounds [BFKL '21]
- Undirected Approximate Max Flow: $n^{o(1)}(\sqrt{n} + D)/\epsilon^3$ rounds [GKKLPS '18]

Flow in Other Distributed Models

Broadcast Congested Clique:

- Minimum Cost Flow: $\tilde{O}(\sqrt{n})$ rounds [F**V** '22]

Deterministic Congested Clique:

- Maximum Flow: $m^{3/7+o(1)}M^{1/7}$ rounds [F**V** '23]
- Unit Capacity Minimum Cost Flow: $\tilde{O}(m^{3/7}(n^{0.158} + n^{o(1)}\text{poly}\log M))$ rounds [F**V** '23]

LP Solver: Main Idea and Challenges

Linear Programming

Minimize $\mathbf{c}^{T}\mathbf{x}$ subject to $\mathbf{M}\mathbf{x} = \mathbf{b}$

LP Solver: Main Idea and Challenges

Linear Programming

Minimize $\mathbf{c}^\mathsf{T}\mathbf{x}$ subject to $\mathbf{M}\mathbf{x} = \mathbf{b}$

Implementation of [LS '14] in CONGEST:

- Interior point method with $\tilde{O}(\sqrt{\text{rank}})$ iterations involving
  - $\tilde{O}(1)$ matrix-vector multiplications
  - $\tilde{O}(1)$ linear system solves
  - Leverage score computation with Johnson-Lindenstrauss
  - Projection on a mixed norm ball:

$$\arg\max_{||\mathbf{x}||_2 + ||\ell^{-1}\mathbf{x}||_\infty \leq 1} \mathbf{a}^T\mathbf{x} \qquad \text{for some } \mathbf{a}, \ell \in \mathbb{R}^m$$

LP Solver: Main Idea and Challenges

Linear Programming

Minimize $\mathbf{c}^{\mathsf{T}}\mathbf{x}$ subject to $\mathbf{M}\mathbf{x} = \mathbf{b}$

Implementation of [LS '14] in CONGEST:

- Interior point method with $\tilde{O}(\sqrt{\text{rank}})$ iterations involving
  - $\tilde{O}(1)$ matrix-vector multiplications
  - $\tilde{O}(1)$ linear system solves
  - Leverage score computation with Johnson-Lindenstrauss
  - Projection on a mixed norm ball:

$$\arg\max_{||\mathbf{x}||_2 + ||\ell^{-1}\mathbf{x}||_\infty \leq 1} \mathbf{a}^T\mathbf{x} \qquad \text{for some } \mathbf{a}, \ell \in \mathbb{R}^m$$

Theorem

Can solve LP with $\mathbf{M}$ *expressed in* $\mathbf{L}_G$ *in* $\tilde{O}(\sqrt{rank})$ *rounds.*

LP Solver: Main Idea and Challenges

Linear Programming

Minimize $\mathbf{c}^{\mathsf{T}}\mathbf{x}$ subject to $\mathbf{M}\mathbf{x} = \mathbf{b}$

Implementation of [LS '14] in CONGEST:

- Interior point method with $\tilde{O}(\sqrt{\text{rank}})$ iterations involving
  - $\tilde{O}(1)$ matrix-vector multiplications
  - $\tilde{O}(1)$ linear system solves
  - Leverage score computation with Johnson-Lindenstrauss
  - Projection on a mixed norm ball:

$$\arg\max_{||\mathbf{x}||_2 + ||\ell^{-1}\mathbf{x}||_\infty \leq 1} \mathbf{a}^T\mathbf{x} \qquad \text{for some } \mathbf{a}, \ell \in \mathbb{R}^m$$

Theorem

Can solve LP with $\mathbf{M}$ *expressed in* $\mathbf{L}_G$ *in* $\tilde{O}(\sqrt{rank})$ *rounds.*

Minimum Cost Flow: [DS '08]

- Rank $= O(\#\text{nodes}) = O(n)$
- Round approximate solution to exact solution

Does faster Max Flow transfer to CONGEST?

[LS '14]:
#iterations: $\tilde{O}(\sqrt{n})$
Time per iteration: $\tilde{O}(m)$

[CKLPPGS '22]:
#iterations: $m^{1+o(1)}$
Time per iteration: $m^{o(1)}$

Does faster Max Flow transfer to CONGEST?

[LS '14]:
#iterations: $\tilde{O}(\sqrt{n})$
Time per iteration: $\tilde{O}(m)$

Iteration count carries over to round complexity

[CKLPPGS '22]:
#iterations: $m^{1+o(1)}$
Time per iteration: $m^{o(1)}$

Running time improvement does not improve round complexity

Does faster Max Flow transfer to CONGEST?

[LS '14]:
#iterations: $\tilde{O}(\sqrt{n})$
Time per iteration: $\tilde{O}(m)$

Iteration count carries over to round complexity

[CKLPPGS '22]:
#iterations: $m^{1+o(1)}$
Time per iteration: $m^{o(1)}$

Running time improvement does not improve round complexity

Question

Is $\tilde{\Theta}(\sqrt{n})$ the right iteration count for min-cost flow LP?

Conclusion

Our Contributions

- CONGEST LP Solver
- CONGEST Minimum Cost Flow

Open Problems

- LP Solver with Fewer Iterations
- CONGEST LP Solver with Fewer *Global* Iterations
- Combinatorial Minimum Cost Flow
- Use LP solver for other problems