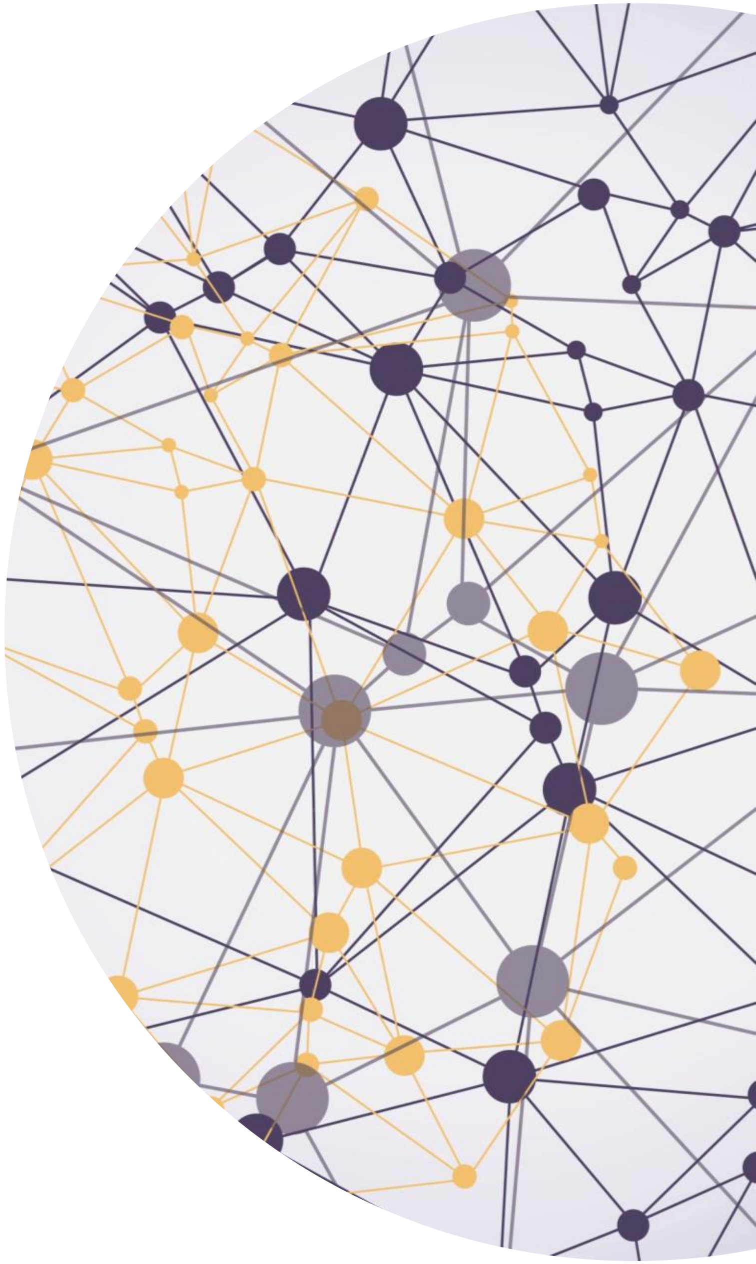
Improved Throughput for All-or-Nothing Multicommodity Flows with Arbitrary Demands

Anya Chaturvedi *Chandra Chekuri* *Andrea W. Richa*
Matthias Rost *Stefan Schmid* *Jamison Weber*

Arizona State University
University of Illinois Urbana Champaign
TU-Berlin & Frauenhofer Institute



All-or-Nothing Flow

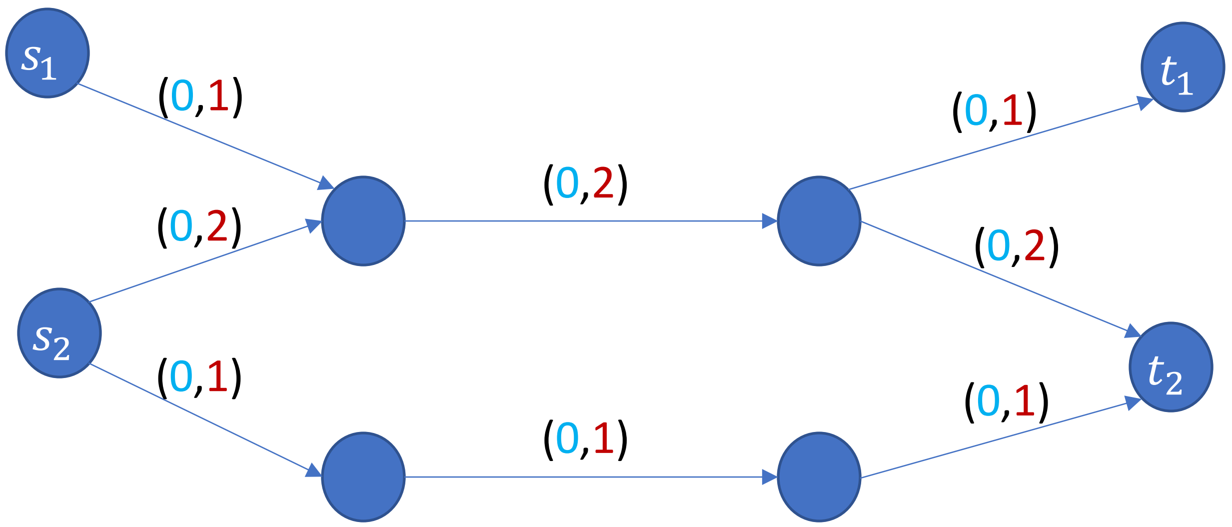
- Directed, connected edge-capacitated network $G = (V, E)$, where $n = |V|, m = |E|$
- k commodities: (source, sink)-pairs $(s_i, t_i), i \in [k]$ and $s_i, t_i \in V$
  with demand $d_i > 0$ and weight $w_i > 0$.

All-or-Nothing Multicommodity Flow (ANF):

- for each commodity i, route either d_i or 0 units from s_i to t_i.

[Chekuri et al., STOC 2004]

Example



Instance $G = (V, E)$

(Flow-Value,Edge-Capacity)

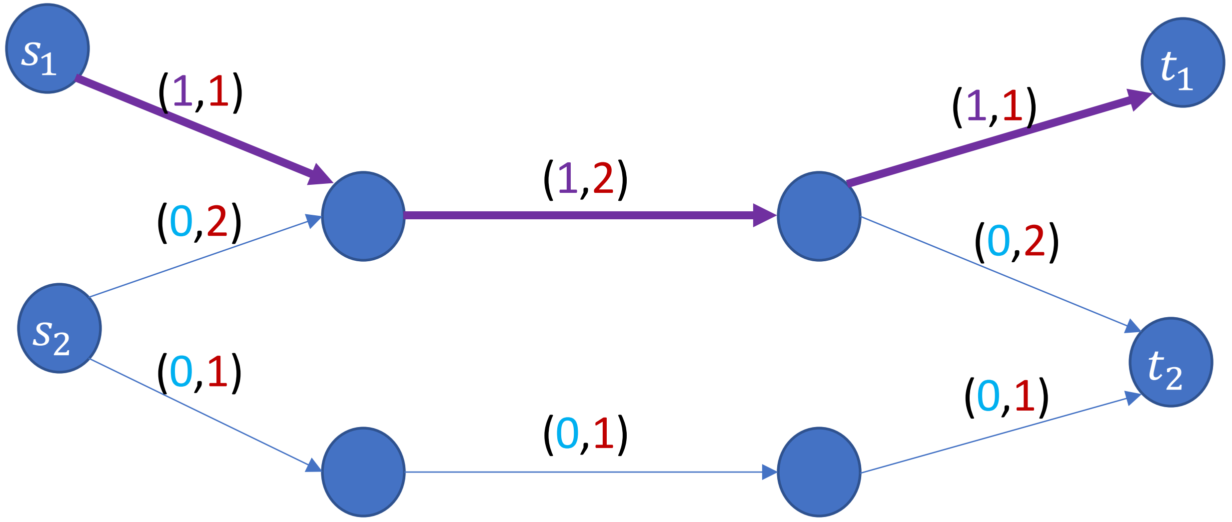
$d_1 = 1, w_1 = 5$
$d_2 = 3, w_2 = 3$

Demands are allowed to be bigger than edge capacities!

Flow for Commodity 1

(Flow-Value,Edge-Capacity)

$d_1 = 1, w_1 = 5$
$d_2 = 3, w_2 = 3$

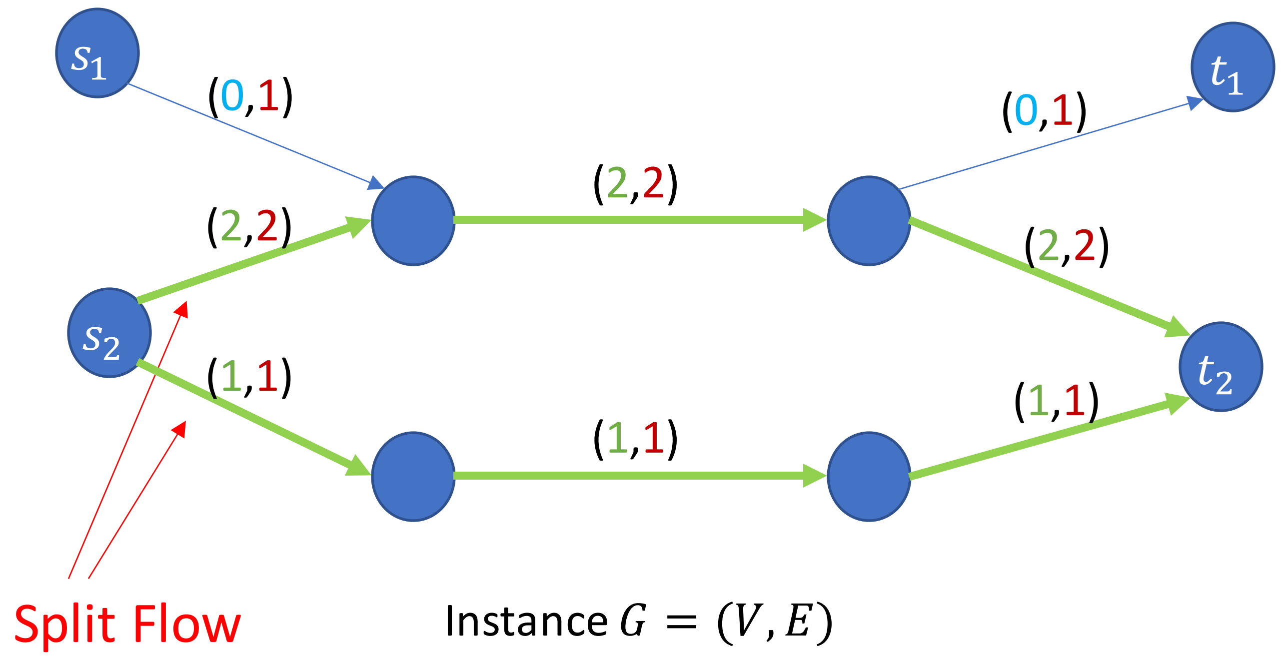


Instance $G = (V, E)$

Flow for Commodity 2

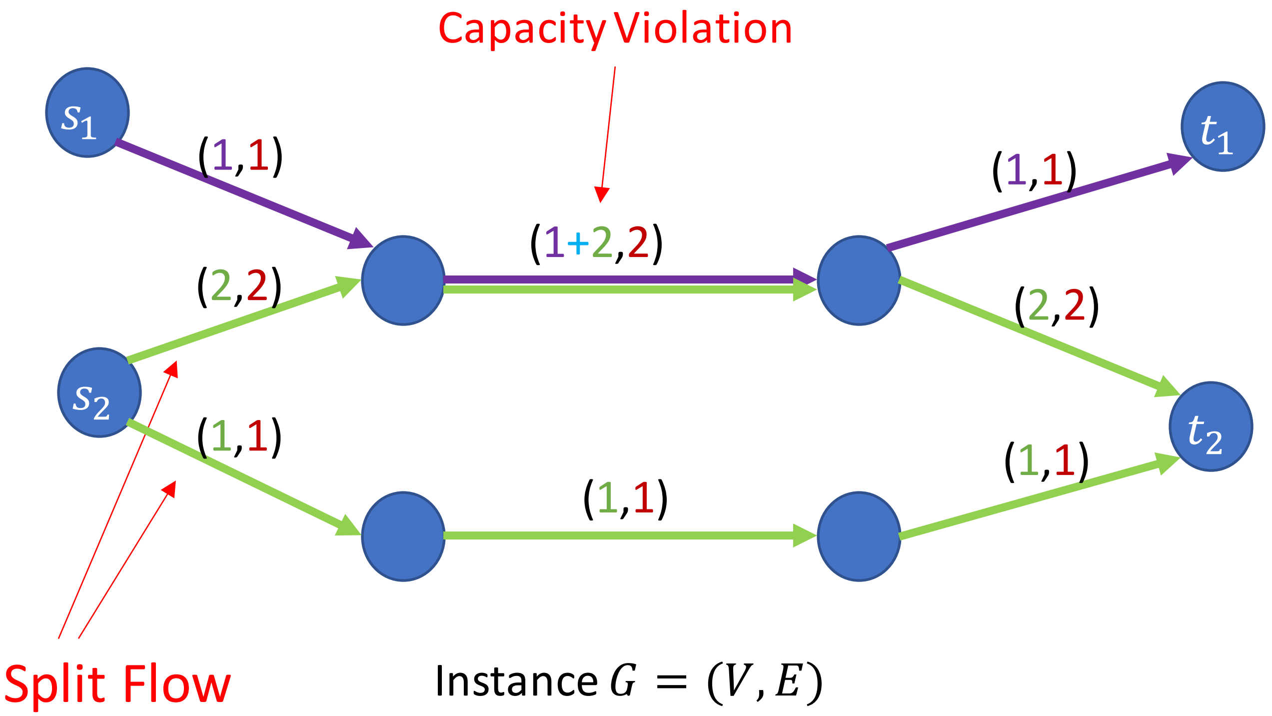
(Flow-Value,Edge-Capacity)

$d_1 = 1, w_1 = 5$
$d_2 = 3, w_2 = 3$



Instance $G = (V, E)$

Routing Both Commodities



Instance $G = (V, E)$

(Flow-Value,Edge-Capacity)

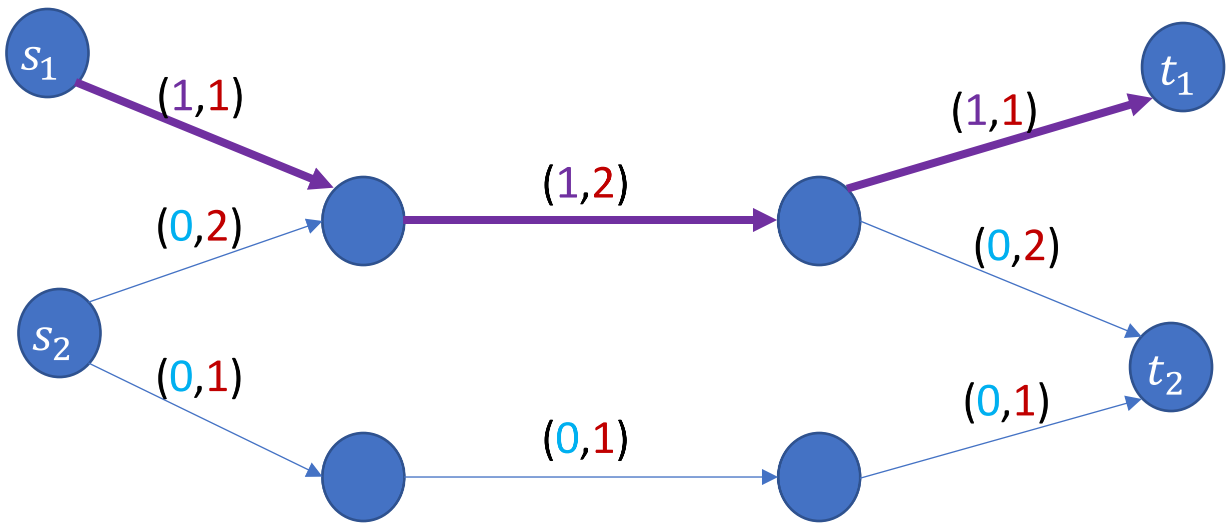
$d_1 = 1, w_1 = 5$
$d_2 = 3, w_2 = 3$

Infeasible Flow!!

Optimum Solution

(Flow-Value,Edge-Capacity)

$d_1 = 2, w_1 = 5$
$d_2 = 3, w_2 = 3$

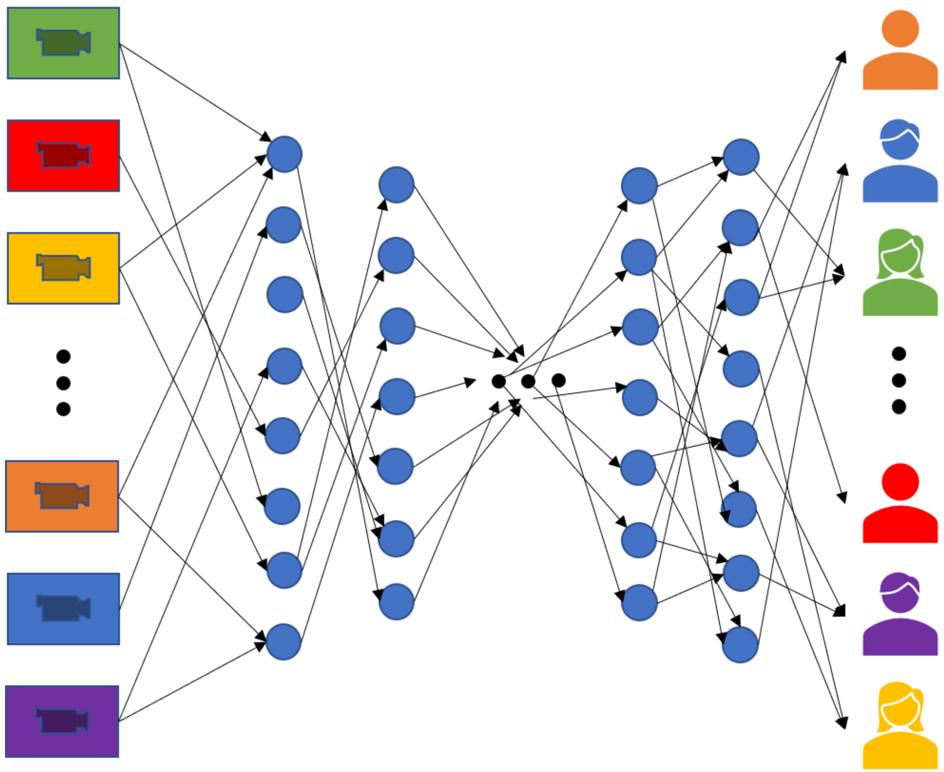


Instance $G = (V, E)$

ANF Problem

- Find a maximum weight routable subset of commodities $S \subseteq [k]$.

- (Optimal) throughput: $\sum_{i \in S} w_i$



Our Contributions

Deeper understanding of packing and compact edge-flow ANF LP formulations and their equivalence.

Randomized Rounding performance improved over state-of-the-art:

- Tighter theoretical guarantees
- Lower space requirements
- Allows for more constrained extensions

Experimental Evaluation

The Packing Formulation

- Let $\mathcal{F}_i$ be the set of all valid canonical d_i-flows for commodity i.
- $|\mathcal{F}_i|$ is exponential
- LP-relaxation can be solved in poly-time via multiplicative-weight updates (MWU)

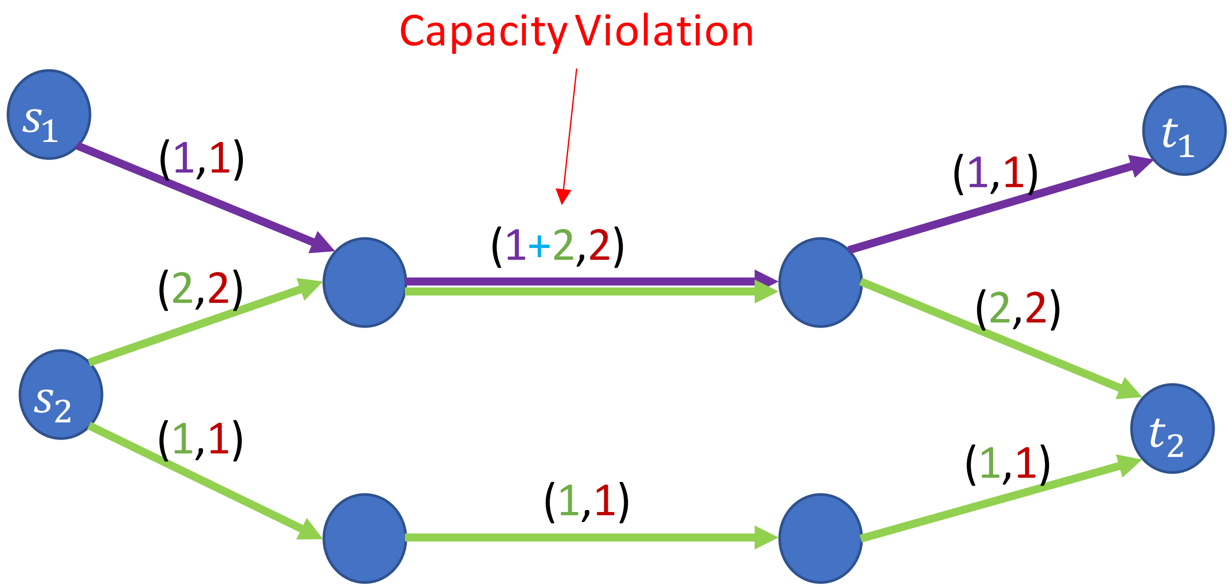
Each commodity flow is expressed as a convex combination of flows in $\mathcal{F}_i$.

x_i indicates whether to route commodity i

Hardness

- Hard to approximate within constant factor
- Idea: Allow congestion > 1, i.e., allow bounded edge capacity violations
- Even with constant factor congestion, still hard to approximate within polynomial factor

[Chuzhoy et al., STOC 2007]



Congestion Ratio: 3/2
Throughput: 5

Theoretical Results

- (α,β)-approximation: A feasible solution with
  - $\geq \alpha$ fraction of the optimal throughput
  - $\leq \beta$ factor bound on largest congestion

Theorem: For $m \geq 9, \epsilon > 1/m$ there exists a polynomial time randomized algorithm that yields a $\left(1-\epsilon, O\left(\frac{\ln m}{\ln \ln m}\right)\right)$-approximation with high probability.

- Improves over $O\left(\frac{1}{3}, O(\sqrt{\mathrm{k} \cdot \log n}\,)\right)$-approximation of [Liu et al., INFOCOM 2019]

The Compact Edge-Flow Formulation

- Polynomial size (polynomial # of variables)
- Yields easier randomized rounding
- Equivalence between the compact edge-flow and the packing formulations

given a feasible solution to one relaxed LP we can obtain a feasible solution to the other relaxation of the same flow and objective values, hence theoretical guarantees carry over!

[Liu et al., INFOCOM 2019]

$$
\begin{aligned}
\max \sum_{i=1}^{k} w_i f_i & \\
\sum_{(s_i,v)\in E} f_{i,(s_i,v)} = f_i & \qquad \forall i \in [k] \\
\sum_{(u,v)\in E} f_{i,(u,v)} = \sum_{(v,u)\in E} f_{i,(v,u)} & \qquad \forall i \in [k], \forall v \in V - \{s_i, t_i\} \\
\sum_{i=1}^{k} f_{i,(u,v)} \cdot d_i \le c_{(u,v)} & \qquad \forall (u,v) \in E \\
\boxed{f_{i,(u,v)} \cdot d_i \le f_i \cdot c_{(u,v)} \qquad \forall i \in [k], \forall (u,v) \in E} & \\
f_{i,(u,v)} \ge 0 & \qquad \forall i \in [k], \forall (u,v) \in E \\
f_i \in \{0,1\} & \qquad \forall i \in [k]
\end{aligned}
$$

Randomized Rounding

- Simple to implement and fast
- Special case of randomized rounding for the packing formulation, so same bounds still apply.
- Derandomization:
  - Deterministic guarantees on approximation bounds
  - Much slower in practice than randomized rounding

Algorithm 1: Randomized Rounding Algorithm

Input : Directed graph $G(V, E)$ with edge capacities $c_e > 0, \forall e \in E$; set of k pairs of commodities (s_i, t_i), each with demand $d_i \geq 0$ and weight $w_i \geq 0$; $\epsilon \in (0, 1]$

Output: The final values of f_i and $f_{i,e}$ and $\sum w_i f_i$

1 Let $\tilde{f}_i, \tilde{f}_{i,e}, \forall i \in [k], \forall e \in E$, be a feasible solution tc compact LP.

2 For each $i \in [k]$, independently, set $f_i = 1$ with probability $\tilde{f}_i$, otherwise set $f_i = 0$.

3 Rescale the fractional flow $\tilde{f}_{i,e}$ from the LP solutior on edge e for commodity i by $\frac{1}{\tilde{f}_i}$: I.e., $f_{i,e} = \frac{\tilde{f}_{i,e}}{\tilde{f}_i} \cdot$ and the flow for commodity i on e is given by $f_{i,e}c$

4 If $\sum_i w_i f_i \geq (1-\epsilon) \sum w_i \tilde{f}_i$ and $\sum_i f_{i,e} d_i \leq (3b \ln m / \ln \ln m) c(e)$ for all $e \in E$, return the corresponding flow assignments given b f_i and $f_{i,e}, \forall i \in [k]$ and $e \in E$. Otherwise, repeat steps 2 and 3, $O((\ln m)/\epsilon^2)$ times.

Solving the Relaxed LP

- CPLEX:
  - Very fast
  - Solves optimally
  - Relatively high space complexity
- Multiplicative Weight Update (MWU):
  - Low Space Complexity
  - Solves the LP to arbitrarily high precision with a trade off with the running time
- Permutation Routing:
  - Very low space complexity
  - Fast heuristic based on MWU (slower than CPLEX)
  - Works well in practice

Algorithm 1: MWU for Multi-Commodity ANF Problem

Inputs: Directed graph G(V,E), $c : E \longrightarrow \mathbb{R}^+$, a set S of k pairs of commodities (s_i, t_i) each with demand d_i and $\gamma \in \mathbb{R}^+$

Change G by adding dummy terminal s_i' and edge(s_i', s_i) with capacity d_i. This ensures that we don't route more than d_i units for pair i. We will assume this has been done and simply use (s_i, t_i) instead of (s_i', t_i)

Output: Total flow f_e on each e. $f(s_i', s_i)/d_i$ gives the fraction of commodity i that is routed

Define a length/cost function $\ell : E \longrightarrow \mathbb{R}^+$ and initialize $\ell_e \longleftarrow 1, \forall e \in E$

Define a function $f : E \longrightarrow \mathbb{R}_{\geq 0}$ and initialize $f_e \longleftarrow 0, \forall e \in E$

Define $\eta \longleftarrow \frac{\ln |E|}{\gamma}$

repeat

    for each commodity $i \in S$ **do**

        Compute min-cost flow of d_i units from s_i to t_i with capacities $c(e)$ and cost given by ℓ. (If no feasible flow then pair i can be dropped.) Let this flow be defined by $g_i(e), e \in E$ and let cost of this flow be $\rho(i) = \sum_e \ell(e) g_i(e)$

    Set $i^* \longleftarrow \text{argmin}_{i \in S} \frac{\rho(i)}{w_i}$

    Compute $\delta \longleftarrow \min_e \frac{\gamma}{\eta} \cdot \frac{g_{i^*}(e)}{c(e)}$

    for each e **do**

        Set $f_e \longleftarrow f_e + \delta g_{i^*}(e)$

        if $f_e > c_e$ **then**

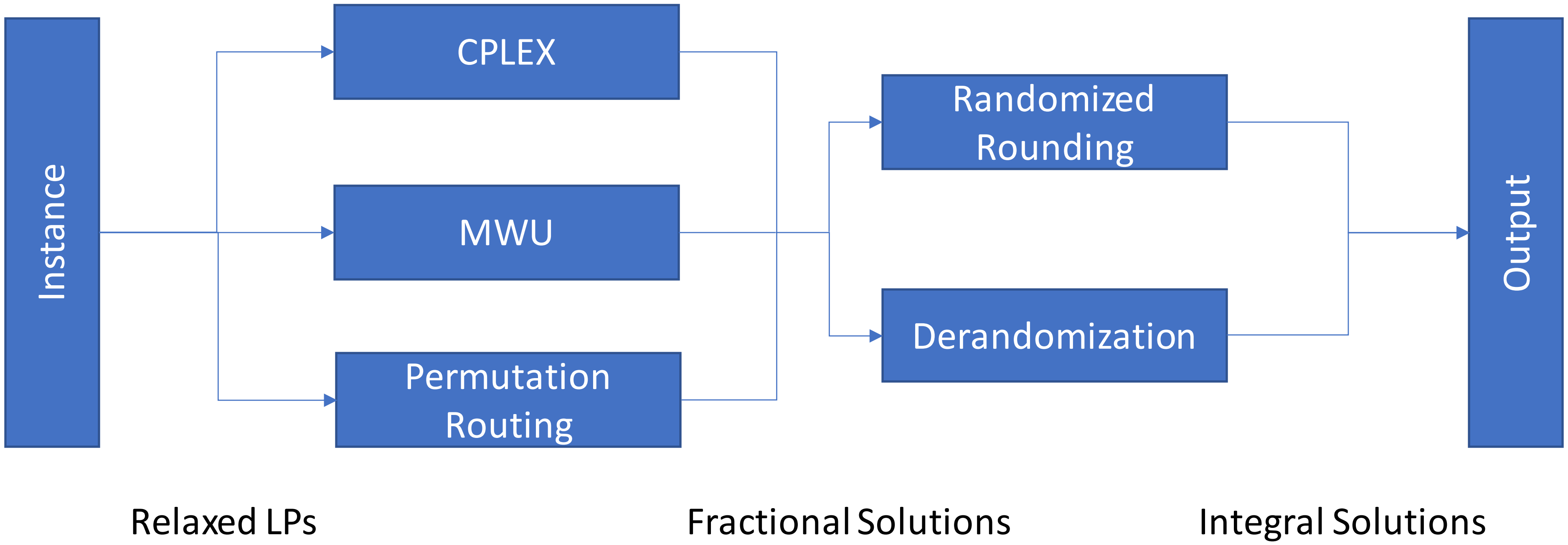
            Output f and halt

        else

            Update $\ell_e \longleftarrow \exp(\eta f_e / c_e)$

until termination

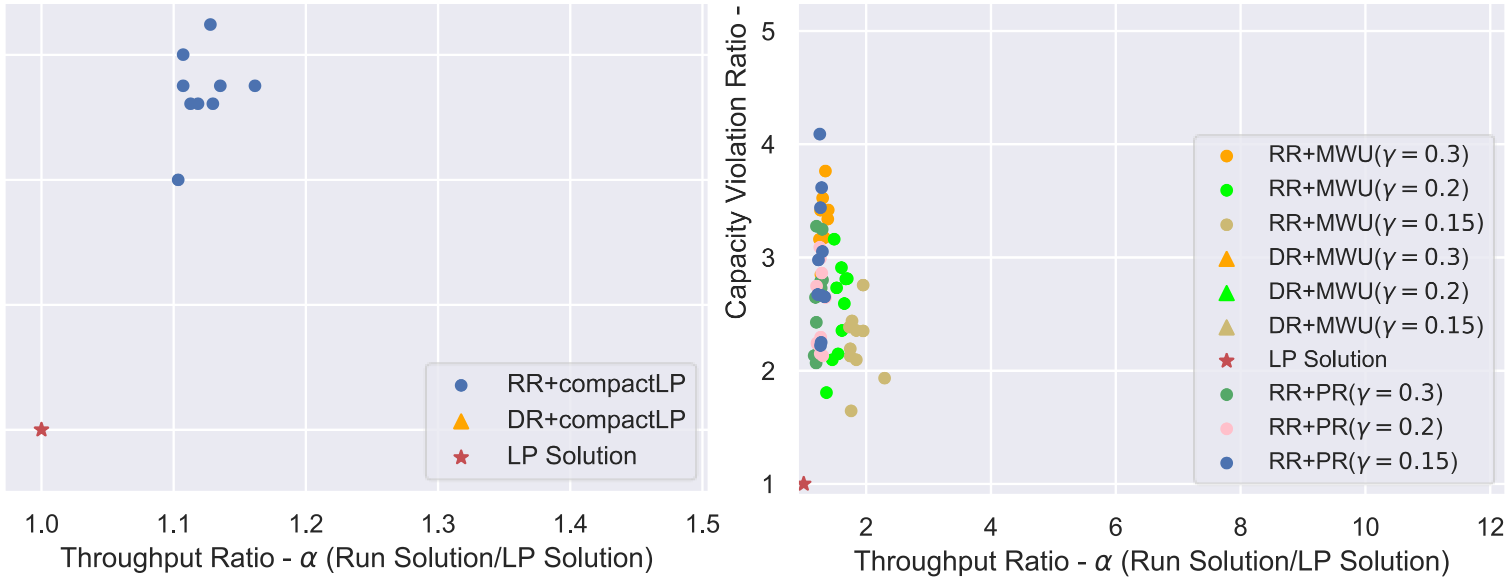
Algorithmic Architecture



Experimental Design

- Germany50 network from SNDlib.
  - Relatively large compared to other networks, with many commodities.
- Execute all combinations of LP solvers and algorithms for integral solutions under various parameters.
- For each experiment, compute:
  - Throughput ratio $\alpha = \frac{Output\ of\ Experiment}{Optimal\ LP\ Solution}$
  - Edge capacity violation ration $\beta = \max_{e \in E}\left(\sum_{i=1}^{k} f_i(e)/c(e)\right)$
- Note that recorded $\alpha > 1$ is possible since $\beta > 1$.

Experimental Results & Findings



Vertices: 50, Edges: 176, Commodities: 662

Future Work

- Experiments on larger networks (for which CPLEX fails)
- Experiments with packing formulation extensions
- Improved Randomized Rounding Performance via resampling techniques (Lovasz-Local-Lemma)