Weighted Chairman Assignment and Flow-Time Scheduling

Victor Reis

joint work with
Siyue Liu

to appear in ITCS 2026

Microsoft Research

University of Washington Theory Seminar
November 18, 2025

Warmup I: discrepancy of two permutations

$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 7 & 4 & 5 & 2 & 3 & 1 \end{array}$$

Goal: Color numbers red and blue so that $|\#\text{red} - \#\text{blue}| \leq 1$ in all $2n$ prefixes

Warmup I: discrepancy of two permutations

$$1\ 2\ 3\ 4\ 5\ 6\ 7\ 8$$

$$6\ 7\ 4\ 5\ 2\ 3\ 1\ 8$$

Goal: Color numbers red and blue so that $|\#\text{red} - \#\text{blue}| \leq 1$ in all $2n$ prefixes

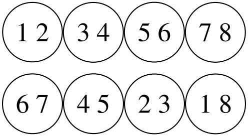
Warmup I: discrepancy of two permutations

1 2 | 3 4 | 5 6 | 7 8

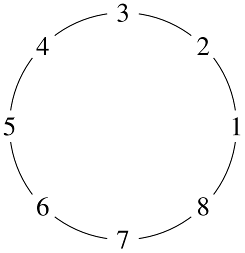
6 7 | 4 5 | 2 3 | 1 8

Goal: Color numbers red and blue so that $|\#\text{red} - \#\text{blue}| \leq 1$ in all $2n$ prefixes

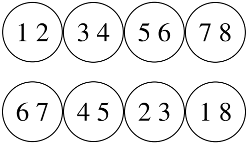
Warmup I: discrepancy of two permutations



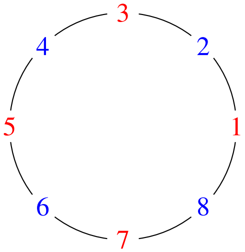
Goal: Color numbers red and blue so that $|\#\text{red} - \#\text{blue}| \leq 1$ in all $2n$ prefixes



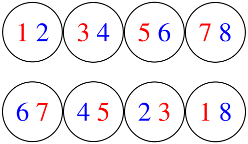
Warmup I: discrepancy of two permutations



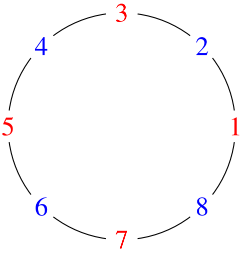
Goal: Color numbers red and blue so that |#red − #blue| ≤ 1 in all $2n$ prefixes



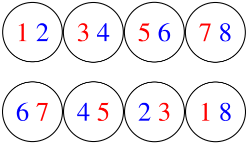
Warmup I: discrepancy of two permutations



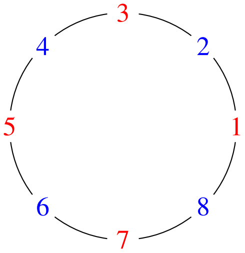
Goal: Color numbers red and blue so that $|\#\text{red} - \#\text{blue}| \leq 1$ in all $2n$ prefixes



Warmup I: discrepancy of two permutations



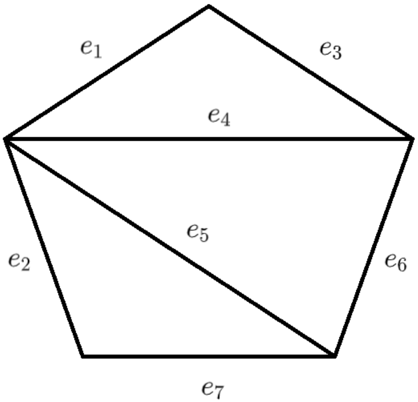
Goal: Color numbers red and blue so that $|\#\text{red} - \#\text{blue}| \leq 1$ in all $2n$ prefixes



Open problem

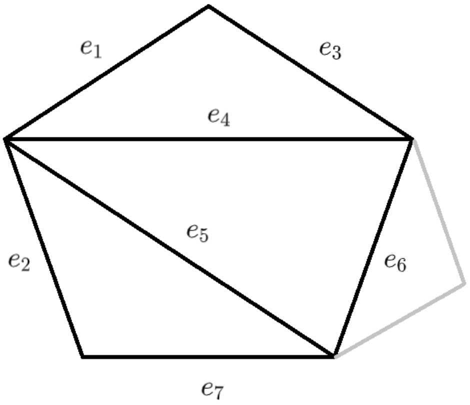
Given *weights* $w \in [0,1]^n$ can we get $|w(\text{red}) - w(\text{blue})| \leq O(1)$ in all $2n$ prefixes?

Warmup II: edge orientation



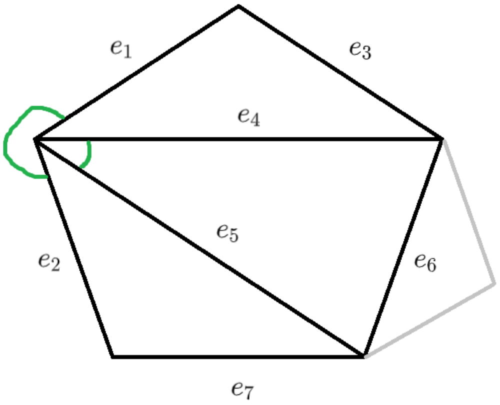
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



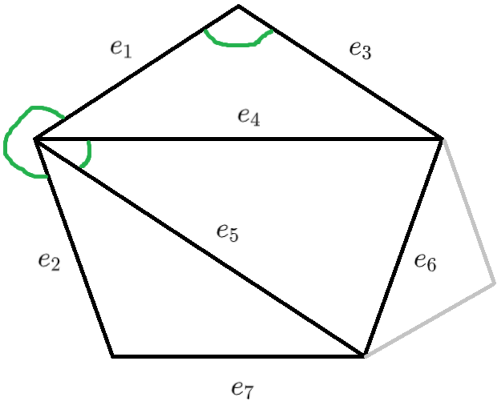
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



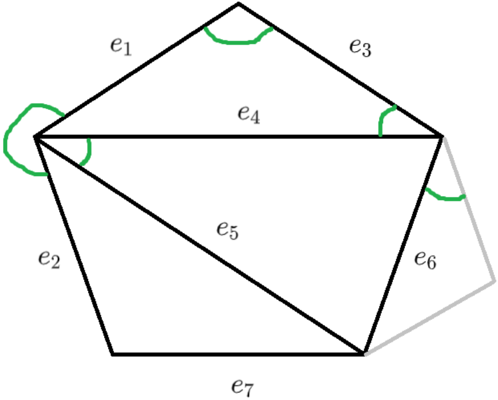
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



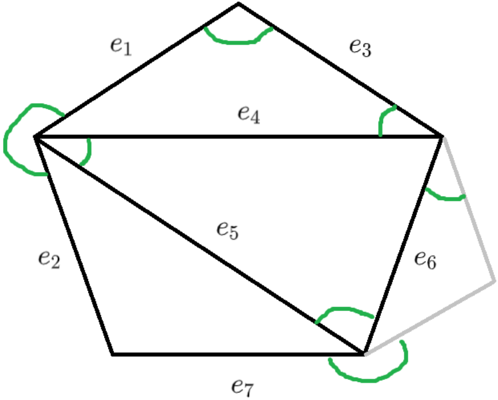
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



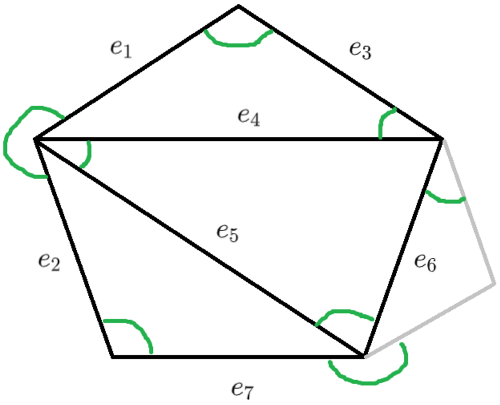
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



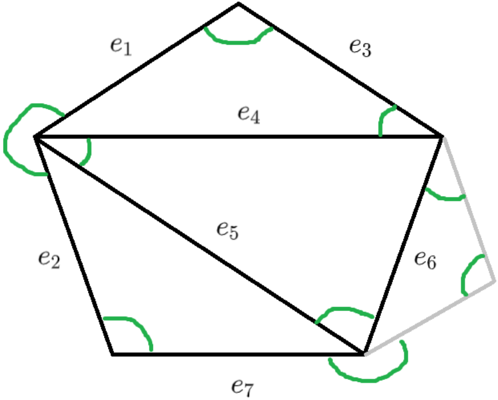
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



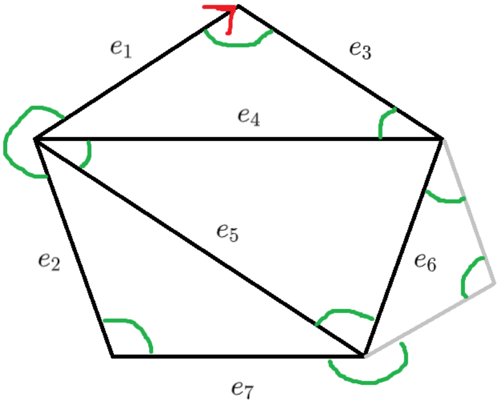
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



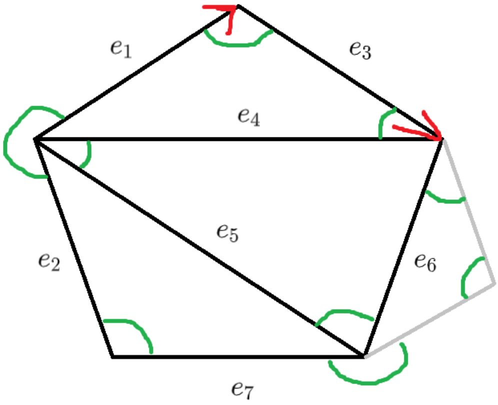
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



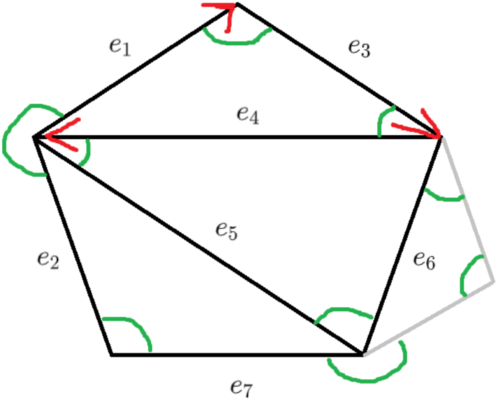
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \le 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



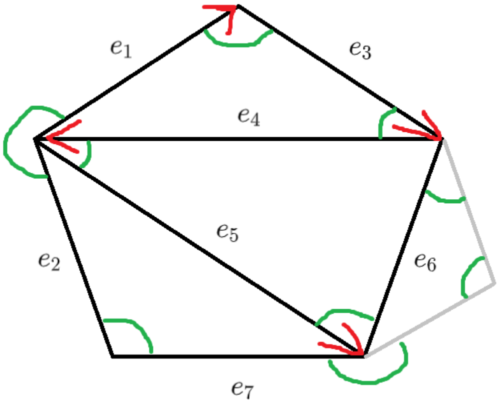
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



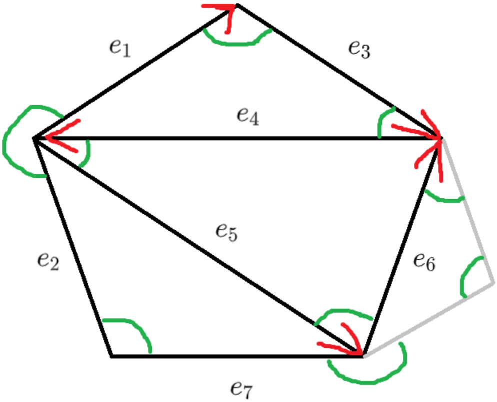
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \le 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



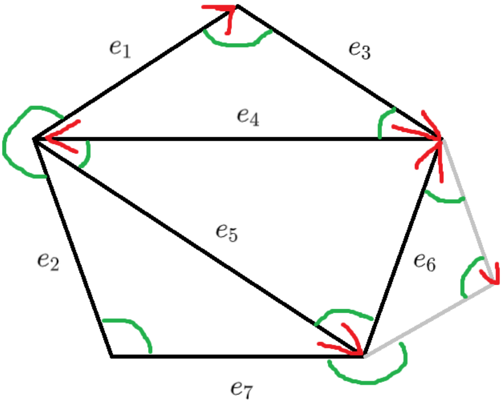
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



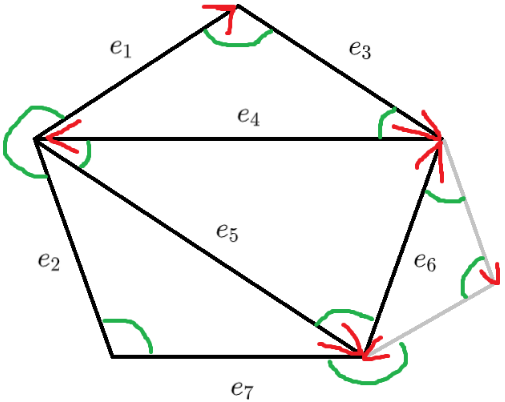
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



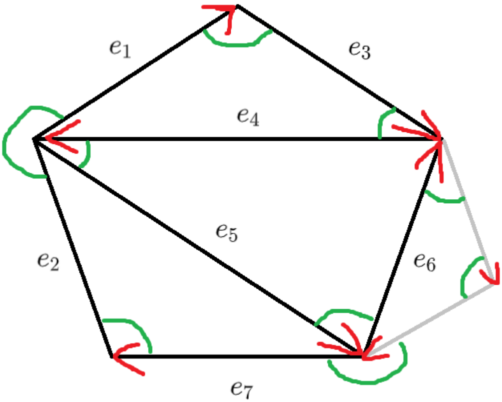
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



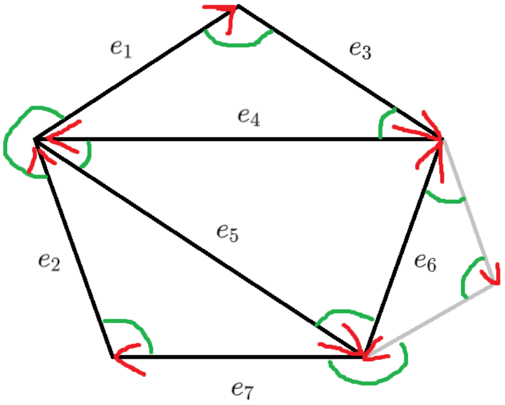
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



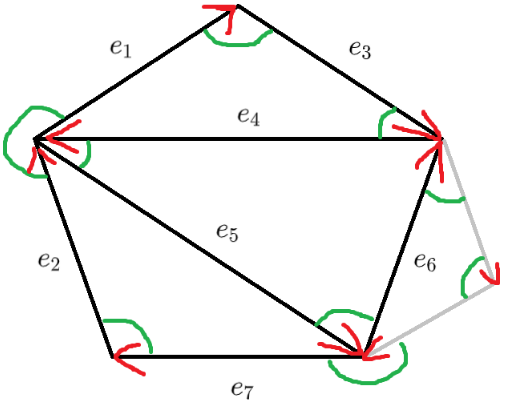
Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation



Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \leq 1$ in all $2|E|$ prefixes

Warmup II: edge orientation

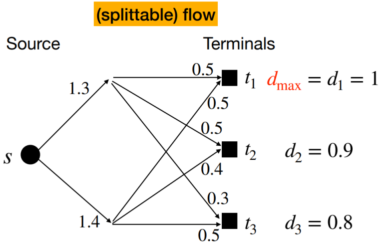


Goal: Orient edges so that $|\#\text{in} - \#\text{out}| \le 1$ in all $2|E|$ prefixes

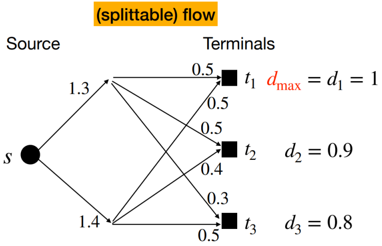
Open problem

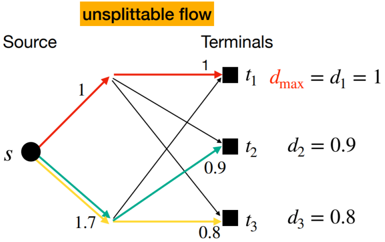
Given *weights* $w \in [0,1]^E$ can we get $|w(\text{in}) - w(\text{out})| \le O(1)$ in all $2|E|$ prefixes?

Single-source unsplittable flow (SSUF)

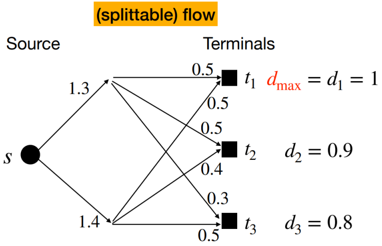


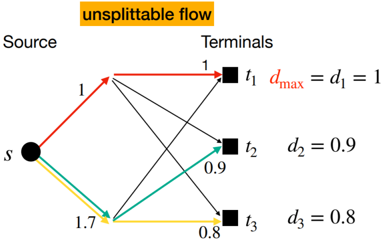
Single-source unsplittable flow (SSUF)





Single-source unsplittable flow (SSUF)

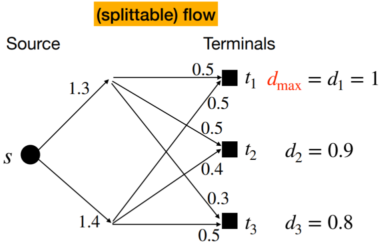


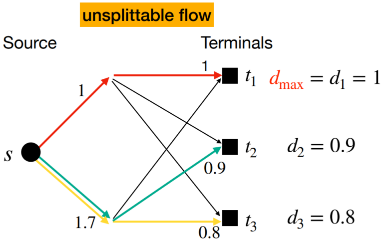


Theorem (Dinitz, Garg, Goemans, 1999)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $y(e) \leq x(e) + d_{\max}$ $\forall e \in E$.

Single-source unsplittable flow (SSUF)





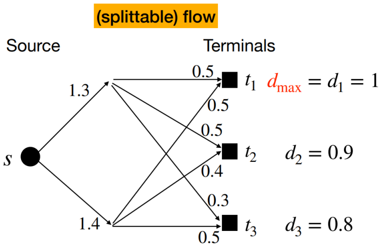
Theorem (Dinitz, Garg, Goemans, 1999)

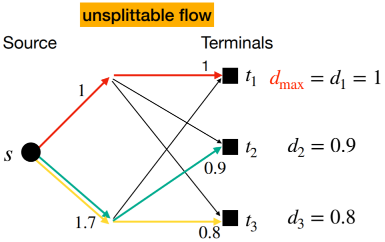
Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $y(e) \leq x(e) + d_{\max}\ \forall e \in E$.

Theorem (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $y(e) \geq x(e) - d_{\max}\ \forall e \in E$.

Single-source unsplittable flow (SSUF)





Theorem (Dinitz, Garg, Goemans, 1999)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $y(e) \leq x(e) + d_{\max}\ \forall e \in E$.

Theorem (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $y(e) \geq x(e) - d_{\max}\ \forall e \in E$.

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}\ \forall e \in E$.

Morell-Skutella conjecture

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}$ $\forall e \in E$.

Morell-Skutella conjecture

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}\ \forall e \in E$.

- Known for bipartite digraphs [Lenstra, Shmoys, Tardos, 1990]

Morell-Skutella conjecture

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}\ \forall e \in E$.

- Known for bipartite digraphs [Lenstra, Shmoys, Tardos, 1990]
- Known for series-parallel digraphs [Almoghrabi, Skutella, Warode, 2024]

Morell-Skutella conjecture

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}\ \forall e \in E$.

- Known for bipartite digraphs [Lenstra, Shmoys, Tardos, 1990]
- Known for series-parallel digraphs [Almoghrabi, Skutella, Warode, 2024]
- Known for planar digraphs [Traub, Koch, Zenklusen, 2024]

Morell-Skutella conjecture

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}\ \forall e \in E$.

- Known for bipartite digraphs [Lenstra, Shmoys, Tardos, 1990]
- Known for series-parallel digraphs [Almoghrabi, Skutella, Warode, 2024]
- Known for planar digraphs [Traub, Koch, Zenklusen, 2024]
- Known when $d_{i+1}/d_i \in \mathbb{N}$ for each $i < n$ [Morell, Skutella, 2020]

Morell-Skutella conjecture

Conjecture (Morell, Skutella, 2020)

Given directed $D = (V, E)$ and flow $x \in \mathbb{R}^E_{\geq 0}$, there exists an *unsplittable* flow $y \in \mathbb{R}^E_{\geq 0}$ such that $|x(e) - y(e)| \leq d_{\max}$ $\forall e \in E$.

- Known for bipartite digraphs [Lenstra, Shmoys, Tardos, 1990]
- Known for series-parallel digraphs [Almoghrabi, Skutella, Warode, 2024]
- Known for planar digraphs [Traub, Koch, Zenklusen, 2024]
- Known when $d_{i+1}/d_i \in \mathbb{N}$ for each $i < n$ [Morell, Skutella, 2020]
- Known to imply a *cost-augmented* one-sided bound conjectured by Goemans [Swamy, Traub, Koch, Zenklusen, 2025]

Application in scheduling: makespan minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$

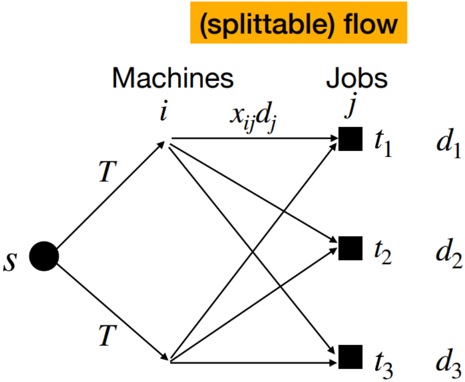
Application in scheduling: makespan minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Objective: minimize the time to complete all jobs

Application in scheduling: makespan minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Objective: minimize the time to complete all jobs

$$
\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j \in J} d_j x_{ij} \leq T, && \forall i \in M, \\
& x \geq 0, \\
& x_{ij} = 0, && \forall (i, j) \notin E.
\end{aligned}
$$

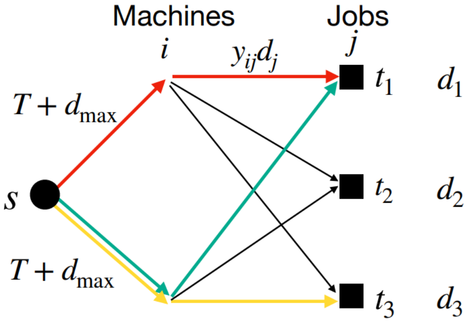


Application in scheduling: makespan minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Objective: minimize the time to complete all jobs

unsplittable flow

$$\begin{aligned}
\min\ & T \\
\text{s.t.}\ & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j \in J} d_j x_{ij} \le T, && \forall i \in M, \\
& x \ge 0, \\
& x_{ij} = 0, && \forall (i,j) \notin E.
\end{aligned}$$



$$\sum_{j \in J} d_j y_{ij} \le \sum_{j \in J} d_j x_{ij} + d_{\max} \le OPT + OPT = 2OPT$$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \leq (r_b - r_a) + T, && \forall i \in M,\ 1 \leq a \leq b \leq n, \\
& x \geq 0, x_{ij} = 0 \quad \forall (i, j) \notin E.
\end{aligned}
$$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \le r_2 \le \cdots \le r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \sum_{i \in M} x_{ij} = 1, & \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \le (r_b - r_a) + T, & \forall i \in M,\ 1 \le a \le b \le n, \\
& x \ge 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$

- Why is this an LP relaxation? Suppose we have an integral solution x

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \le r_2 \le \cdots \le r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min\ & T \\
\text{s.t.}\ & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \le (r_b - r_a) + T, && \forall i \in M,\ 1 \le a \le b \le n, \\
& x \ge 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$

- Why is this an LP relaxation? Suppose we have an integral solution x
- Fix job j with $x_{ij} = 1$ for some i and let t be the last time in $[0, r_j]$ when i idle

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \leq (r_b - r_a) + T, && \forall i \in M,\ 1 \leq a \leq b \leq n, \\
& x \geq 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$

- Why is this an LP relaxation? Suppose we have an integral solution x
- Fix job j with $x_{ij} = 1$ for some i and let t be the last time in $[0, r_j]$ when i idle
- May assume $t = r_a$ for some job $a \leq j$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \leq (r_b - r_a) + T, && \forall i \in M,\ 1 \leq a \leq b \leq n, \\
& x \geq 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$

- Why is this an LP relaxation? Suppose we have an integral solution x
- Fix job j with $x_{ij} = 1$ for some i and let t be the last time in $[0, r_j]$ when i idle
- May assume $t = r_a$ for some job $a \leq j$
- Machine busy in $[r_a, r_j]$ with total volume assigned $\leq r_j - r_a + T$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

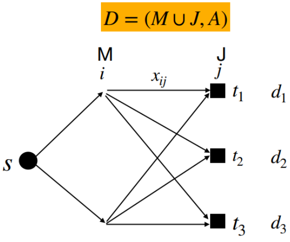
$$
\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \leq (r_b - r_a) + T, && \forall i \in M,\ 1 \leq a \leq b \leq n, \\
& x \geq 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$

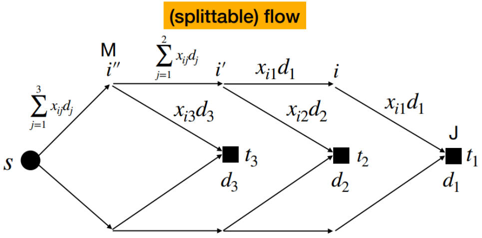
- Why is this an LP relaxation? Suppose we have an integral solution x
- Fix job j with $x_{ij} = 1$ for some i and let t be the last time in $[0, r_j]$ when i idle
- May assume $t = r_a$ for some job $a \leq j$
- Machine busy in $[r_a, r_j]$ with total volume assigned $\leq r_j - r_a + T$
- By time r_j, $\leq (r_j - r_a + T) - |[r_a, r_j]| = T$ volume left, so j finishes by $r_j + T$

Application in scheduling II: flow-time minimization

- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \le r_2 \le \cdots \le r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min\ & T \\
\text{s.t.}\ & \sum_{i \in M} x_{ij} = 1, && \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \le (r_b - r_a) + T, && \forall i \in M,\ 1 \le a \le b \le n, \\
& x \ge 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$



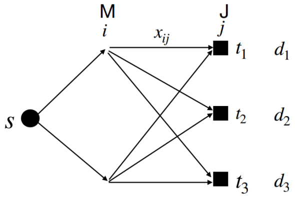


Application in scheduling II: flow-time minimization

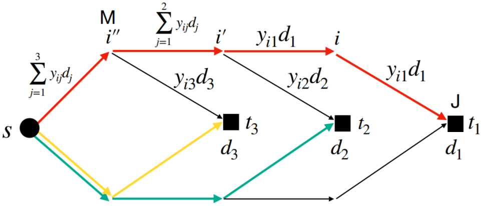
- Jobs J with processing times d_j, machines M, bipartite graph $D = (M \cup J, E)$
- Jobs ordered by release time $r_1 \leq r_2 \leq \cdots \leq r_n$
- Minimize the max *flow-time* $\max_{j \in J}(C_j - r_j)$ where C_j is the completion time

$$
\begin{aligned}
\min\ & T \\
\text{s.t.}\ & \sum_{i \in M} x_{ij} = 1, & \forall j \in J, \\
& \sum_{j=a}^{b} d_j x_{ij} \leq (r_b - r_a) + T, & \forall i \in M,\ 1 \leq a \leq b \leq n, \\
& x \geq 0, x_{ij} = 0 \quad \forall (i,j) \notin E.
\end{aligned}
$$

$D = (M \cup J, A)$



unsplittable flow



Application in scheduling II: flow-time minimization

Special case of Morell-Skutella's conjecture

Given bipartite $D = (M \cup J, E)$, and a fractional assignment $x \in [0,1]^{m \times n}$, there is an assignment $y \in \{0,1\}^{m \times n}$ such that

$$\left| \sum_{j=1}^{t} d_j x_{ij} - \sum_{j=1}^{t} d_j y_{ij} \right| \leq d_{\max}, \qquad \forall i \in [m],\ t \in [n].$$

Application in scheduling II: flow-time minimization

Special case of Morell-Skutella's conjecture

Given bipartite $D = (M \cup J, E)$, and a fractional assignment $x \in [0,1]^{m \times n}$, there is an assignment $y \in \{0,1\}^{m \times n}$ such that

$$\left| \sum_{j=1}^{t} d_j x_{ij} - \sum_{j=1}^{t} d_j y_{ij} \right| \leq d_{\max}, \qquad \forall i \in [m],\ t \in [n].$$

- Given a feasible LP solution x, can round it to y so that for any $1 \leq a \leq b \leq n$,

$$\begin{aligned}\sum_{j=a}^{b} d_j y_{ij} = \sum_{j=1}^{b} d_j y_{ij} - \sum_{j=1}^{a-1} d_j y_{ij} &\leq \left(\sum_{j=1}^{b} d_j x_{ij} + d_{\max} \right) - \left(\sum_{j=1}^{a-1} d_j x_{ij} - d_{\max} \right) \\ &= \sum_{j=a}^{b} d_j x_{ij} + 2d_{\max} \\ &\leq (r_b - r_a) + OPT + 2d_{\max} \\ &\leq (r_b - r_a) + 3OPT.\end{aligned}$$

Application in scheduling II: flow-time minimization

Special case of Morell-Skutella's conjecture

Given bipartite $D = (M \cup J, E)$, and a fractional assignment $x \in [0,1]^{m \times n}$, there is an assignment $y \in \{0,1\}^{m \times n}$ such that

$$\left| \sum_{j=1}^{t} d_j x_{ij} \ - \ \sum_{j=1}^{t} d_j y_{ij} \right| \leq \ d_{\max}, \qquad \forall i \in [m],\ t \in [n].$$

- Given a feasible LP solution x, can round it to y so that for any $1 \leq a \leq b \leq n$,

$$\begin{aligned} \sum_{j=a}^{b} d_j y_{ij} = \sum_{j=1}^{b} d_j y_{ij} - \sum_{j=1}^{a-1} d_j y_{ij} &\leq \Big(\sum_{j=1}^{b} d_j x_{ij} + d_{\max} \Big) - \Big(\sum_{j=1}^{a-1} d_j x_{ij} - d_{\max} \Big) \\ &= \sum_{j=a}^{b} d_j x_{ij} + 2 d_{\max} \\ &\leq (r_b - r_a) + OPT + 2 d_{\max} \\ &\leq (r_b - r_a) + 3 OPT. \end{aligned}$$

- Known up to $O(\sqrt{\log n})$ [Bansal, Svensson, Rohwedder STOC 2022]

Main results

Theorem (Liu, R., 2026)

Given a fractional assignment $x \in [0,1]^{m\times n}$ and weights $d \in \mathbb{R}^n_{>0}$, there exists an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,

$$\left|\sum_{j=1}^{t} d_j x_{ij} - \sum_{j=1}^{t} d_j y_{ij}\right| < \max_{j\in[n]} d_j.$$

Main results

Theorem (Liu, R., 2026)

Given a fractional assignment $x \in [0,1]^{m\times n}$ and weights $d \in \mathbb{R}^n_{>0}$, there exists an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,

$$\left|\sum_{j=1}^{t} d_j x_{ij} - \sum_{j=1}^{t} d_j y_{ij}\right| < \max_{j\in[n]} d_j .$$

- The RHS can be improved to $(1 - \frac{1}{2m-2}) \cdot \max_{j\in[n]} d_j$, which is tight:

$$\begin{array}{c|cccc|}
 & 1 & 2 & \cdots & m-1 \\
\hline
1 & \frac{1}{2m-2} & \frac{1}{m-1} & \cdots & \frac{1}{m-1} \\
2 & \frac{1}{2m-2} & \frac{1}{m-1} & \cdots & \boxed{\frac{1}{m-1}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
m-1 & \frac{1}{2m-2} & \boxed{\frac{1}{m-1}} & \cdots & \frac{1}{m-1} \\
m & \boxed{\frac{1}{2}} & 0 & \cdots & 0 \\
\hline
\end{array}$$

Application to flow-time scheduling

Corollary (Liu, R., 2026)

A 3-approx. algorithm for maximum flow-time minimization on identical machines.

Application to flow-time scheduling

Corollary (Liu, R., 2026)

A 3-approx. algorithm for maximum flow-time minimization on identical machines.

- Already known: sort jobs by release times and schedule on least busy machine [Mastrolilli, 2004]

Application to flow-time scheduling

Corollary (Liu, R., 2026)

A 3-approx. algorithm for maximum flow-time minimization on identical machines.

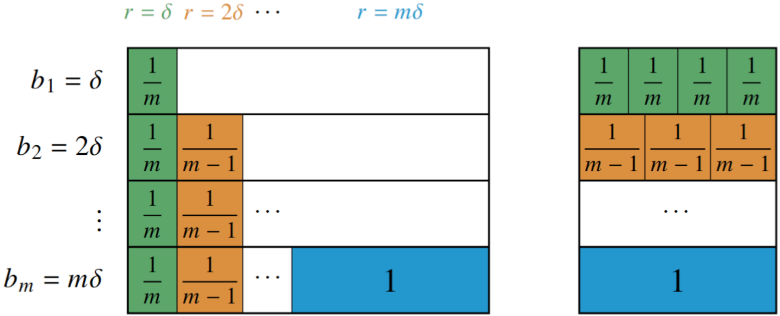
- Already known: sort jobs by release times and schedule on least busy machine [Mastrolilli, 2004]
- Suppose now each machine i has a *closing time* b_i

Application to flow-time scheduling

Corollary (Liu, R., 2026)

A 3-approx. algorithm for maximum flow-time minimization on identical machines.

- Already known: sort jobs by release times and schedule on least busy machine [Mastrolilli, 2004]
- Suppose now each machine i has a *closing time* b_i
- Can schedule job j on machine i if $r_j \leq b_i$

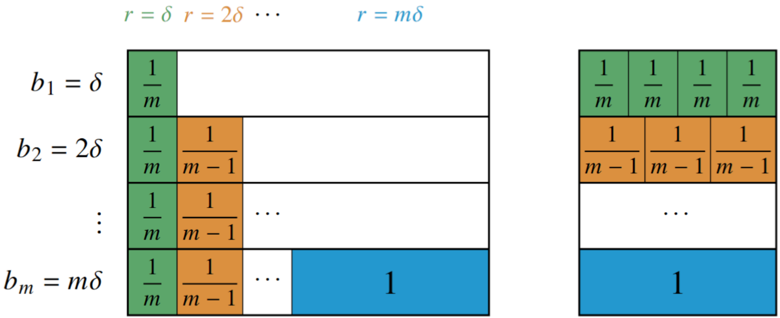


Application to flow-time scheduling

Corollary (Liu, R., 2026)

A 3-approx. algorithm for maximum flow-time minimization on ~~identical~~ machines with closing times.

- Already known: sort jobs by release times and schedule on least busy machine [Mastrolilli, 2004]
- Suppose now each machine i has a *closing time* b_i
- Can schedule job j on machine i if $r_j \leq b_i$



Algorithm

Given fractional assignment $x \in [0,1]^{m\times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

Algorithm

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

Algorithm 1 EARLIEST DEADLINE ALGORITHM

for $t \in [n]$ **do**
  $C(t) \leftarrow \{i \in [m] : P_t(i) > N_{t-1}(i)\}$ ▹ set of candidates at time t
  for $i \in [m]$ **do**
    $D_t(i) \leftarrow \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$ ▹ deadline of i at time t
  end for
  Choose $s_t \in \arg\min_{i \in C(t)} D_t(i)$
  $y_{s_t,t} \leftarrow 1$ and $y_{i,t} \leftarrow 0$ for $i \in [m] \setminus \{s_t\}$
end for

Algorithm

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

Algorithm 1 Earliest Deadline Algorithm

for $t \in [n]$ **do**
  $C(t) \leftarrow \{i \in [m] : P_t(i) > N_{t-1}(i)\}$ ▹ set of candidates at time t
  for $i \in [m]$ **do**
    $D_t(i) \leftarrow \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$ ▹ deadline of i at time t
  end for
  Choose $s_t \in \arg\min_{i \in C(t)} D_t(i)$
  $y_{s_t,t} \leftarrow 1$ and $y_{i,t} \leftarrow 0$ for $i \in [m] \setminus \{s_t\}$
end for

Claim 1: The algorithm outputs a valid assignment, i.e. $C(t) \neq \emptyset$ for all $t \in [n]$.

Algorithm

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

Algorithm 1 Earliest Deadline Algorithm

```
for t ∈ [n] do
    C(t) ← {i ∈ [m] : P_t(i) > N_{t−1}(i)}                ▷ set of candidates at time t
    for i ∈ [m] do
        D_t(i) ← min{T ≥ t : P_T(i) ≥ N_{t−1}(i) + 1}      ▷ deadline of i at time t
    end for
    Choose s_t ∈ arg min_{i∈C(t)} D_t(i)
    y_{s_t,t} ← 1 and y_{i,t} ← 0 for i ∈ [m] \ {s_t}
end for
```

Claim 1: The algorithm outputs a valid assignment, i.e. $C(t) \neq \emptyset$ for all $t \in [n]$.

$$\sum_{i=1}^{m} \big(P_t(i) - N_{t-1}(i)\big) = \sum_{j=1}^{t-1} d_j \Big(\sum_{i=1}^{m} x_{ij} - \sum_{i=1}^{m} y_{ij} \Big) + d_t \sum_{i=1}^{m} x_{ij} = d_t > 0.$$

Algorithm

Given fractional assignment $x \in [0,1]^{m\times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

Algorithm 1 EARLIEST DEADLINE ALGORITHM

---

for $t \in [n]$ **do**
  $C(t) \leftarrow \{i \in [m] : P_t(i) > N_{t-1}(i)\}$ ▷ set of candidates at time t
  for $i \in [m]$ **do**
    $D_t(i) \leftarrow \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$ ▷ deadline of i at time t
  end for
  Choose $s_t \in \arg\min_{i \in C(t)} D_t(i)$
  $y_{s_t,t} \leftarrow 1$ and $y_{i,t} \leftarrow 0$ for $i \in [m] \setminus \{s_t\}$
end for

Claim 1: The algorithm outputs a valid assignment, i.e. $C(t) \neq \emptyset$ for all $t \in [n]$.
Claim 2: $\Delta_t(i) > -1$ for all $i \in [m]$, $t \in [n]$.

Algorithm

Given fractional assignment $x \in [0,1]^{m\times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

Algorithm 1 Earliest Deadline Algorithm

for $t \in [n]$ **do**
  $C(t) \leftarrow \{i \in [m] : P_t(i) > N_{t-1}(i)\}$ ▷ set of candidates at time t
  for $i \in [m]$ **do**
    $D_t(i) \leftarrow \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$ ▷ deadline of i at time t
  end for
  Choose $s_t \in \arg\min_{i\in C(t)} D_t(i)$
  $y_{s_t,t} \leftarrow 1$ and $y_{i,t} \leftarrow 0$ for $i \in [m] \setminus \{s_t\}$
end for

Claim 1: The algorithm outputs a valid assignment, i.e. $C(t) \neq \emptyset$ for all $t \in [n]$.
Claim 2: $\Delta_t(i) > -1$ for all $i \in [m]$, $t \in [n]$.
Claim 3: $\Delta_t(i) < 1$ for all $i \in [m]$, $t \in [n]$.

Lower bound proof

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 2

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) > -1$.

Lower bound proof

Given fractional assignment $x \in [0,1]^{m\times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 2

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) > -1$.

Proof. If i has not been selected at any time up to t, then $N_t(i) = 0$ and

$$\Delta_t(i) = P_t(i) \geq 0 > -1.$$

Lower bound proof

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 2

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) > -1$.

Proof. If i has not been selected at any time up to t, then $N_t(i) = 0$ and

$$\Delta_t(i) = P_t(i) \geq 0 > -1.$$

Otherwise, let $j \in [t]$ be the latest time such that $s_j = i$. Then

$$\Delta_t(i) = P_t(i) - N_t(i)$$

Lower bound proof

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 2

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) > -1$.

Proof. If i has not been selected at any time up to t, then $N_t(i) = 0$ and

$$\Delta_t(i) = P_t(i) \geq 0 > -1.$$

Otherwise, let $j \in [t]$ be the latest time such that $s_j = i$. Then

$$\begin{aligned}\Delta_t(i) &= P_t(i) - N_t(i) \\ &= P_t(i) - N_{j-1}(i) - d_j\end{aligned}$$

Lower bound proof

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 2

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) > -1$.

Proof. If i has not been selected at any time up to t, then $N_t(i) = 0$ and

$$\Delta_t(i) = P_t(i) \geq 0 > -1.$$

Otherwise, let $j \in [t]$ be the latest time such that $s_j = i$. Then

$$\begin{aligned}
\Delta_t(i) &= P_t(i) - N_t(i) \\
&= P_t(i) - N_{j-1}(i) - d_j \\
&\geq P_j(i) - N_{j-1}(i) - d_j
\end{aligned}$$

Lower bound proof

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 2

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) > -1$.

Proof. If i has not been selected at any time up to t, then $N_t(i) = 0$ and

$$\Delta_t(i) = P_t(i) \geq 0 > -1.$$

Otherwise, let $j \in [t]$ be the latest time such that $s_j = i$. Then

$$\begin{aligned}
\Delta_t(i) &= P_t(i) - N_t(i) \\
&= P_t(i) - N_{j-1}(i) - d_j \\
&\geq P_j(i) - N_{j-1}(i) - d_j \\
&> -d_j \geq -1.
\end{aligned}$$

Upper bound proof I

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 3 (Main technical lemma)

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) < 1$.

Upper bound proof I

Given fractional assignment $x \in [0, 1]^{m \times n}$, $d \in (0, 1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 3 (Main technical lemma)

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) < 1$.

Proof. Assuming otherwise $I^+ := \{i \in [m] : \Delta_t(i) \geq 1\} \neq \emptyset$ for some $t \in [n]$.

Upper bound proof I

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 3 (Main technical lemma)

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) < 1$.

Proof. Assuming otherwise $I^+ := \{i \in [m] : \Delta_t(i) \geq 1\} \neq \emptyset$ for some $t \in [n]$.

$$\sum_{i \in [m]} \Delta_t(i) = 0 \implies \{i \in [m] : \Delta_t(i) < 0\} \neq \emptyset.$$

Upper bound proof I

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 3 (Main technical lemma)

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) < 1$.

Proof. Assuming otherwise $I^+ := \{i \in [m] : \Delta_t(i) \geq 1\} \neq \emptyset$ for some $t \in [n]$.

$$\sum_{i \in [m]} \Delta_t(i) = 0 \implies \{i \in [m] : \Delta_t(i) < 0\} \neq \emptyset.$$

So $t^- := \max\{j \leq t : \Delta_t(s_j) < 0\}$ is well-defined.

Upper bound proof I

Given fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$, define

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

$$C(t) := \{i \in [m] : P_t(i) > N_{t-1}(i)\}, \quad D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1\}$$

Claim 3 (Main technical lemma)

For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) < 1$.

Proof. Assuming otherwise $I^+ := \{i \in [m] : \Delta_t(i) \geq 1\} \neq \emptyset$ for some $t \in [n]$.

$$\sum_{i \in [m]} \Delta_t(i) = 0 \implies \{i \in [m] : \Delta_t(i) < 0\} \neq \emptyset.$$

So $t^- := \max\{j \leq t : \Delta_t(s_j) < 0\}$ is well-defined.
We will need to define another time $t^- \leq t' \leq t$ and analyze what happens in $(t', t]$.

Upper bound proof II

$t^- := \max\{j \leq t : \Delta_t(s_j) < 0\}$ is well-defined.

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$P_t(s_{t^-}) < N_t(s_{t^-})$$

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

t' := $\max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \end{aligned}$$

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \\ &= N_{t^- -1}(s_{t^-}) + d_{t^-} \end{aligned}$$

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \\ &= N_{t^- -1}(s_{t^-}) + d_{t^-} \\ &\le N_{t^- -1}(s_{t^-}) + 1. \quad \square \end{aligned}$$

Upper bound proof II

$t^- := \max\{j \leq t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \leq t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \\ &= N_{t^- -1}(s_{t^-}) + d_{t^-} \\ &\leq N_{t^- -1}(s_{t^-}) + 1. \quad \square \end{aligned}$$

- By maximality, for any $j \in (t', t]$, $P_t(s_j) \geq N_{j-1}(s_j) + 1 \geq N_{t'-1}(s_j) + 1$.

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \\ &= N_{t^- - 1}(s_{t^-}) + d_{t^-} \\ &\le N_{t^- - 1}(s_{t^-}) + 1. \quad \square \end{aligned}$$

- By maximality, for any $j \in (t', t]$, $P_t(s_j) \ge N_{j-1}(s_j) + 1 \ge N_{t'-1}(s_j) + 1$.
- Define $I := \{i \in [m] : P_t(i) \ge N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \le t\}$.

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \\ &= N_{t^- - 1}(s_{t^-}) + d_{t^-} \\ &\le N_{t^- - 1}(s_{t^-}) + 1. \quad \square \end{aligned}$$

- By maximality, for any $j \in (t', t]$, $P_t(s_j) \ge N_{j-1}(s_j) + 1 \ge N_{t'-1}(s_j) + 1$.
- Define $I := \{i \in [m] : P_t(i) \ge N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \le t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t]$.

Upper bound proof II

$t^- := \max\{j \le t : \Delta_t(s_j) < 0\}$ is well-defined.

Claim 3.1.

$t' := \max\{j \le t : P_t(s_j) < N_{j-1}(s_j) + 1\}$ is well-defined.

Proof. It suffices to show that t^- satisfies the inequality:

$$\begin{aligned} P_t(s_{t^-}) &< N_t(s_{t^-}) \\ &= N_{t^-}(s_{t^-}) \\ &= N_{t^- - 1}(s_{t^-}) + d_{t^-} \\ &\le N_{t^- - 1}(s_{t^-}) + 1. \quad \square \end{aligned}$$

- By maximality, for any $j \in (t', t]$, $P_t(s_j) \ge N_{j-1}(s_j) + 1 \ge N_{t'-1}(s_j) + 1$.
- Define $I := \{i \in [m] : P_t(i) \ge N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \le t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t]$.
- $I \ne \emptyset$ since $I^+ \subseteq I$: $P_t(i) \ge N_t(i) + 1 \ge N_{t'-1}(i) + 1$ for any $i \in I^+$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Contradiction assuming Claim 3.2.:

$$\sum_{j \in (t', t]} d_j < 1 + \sum_{j \in (t', t]} d_j$$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Contradiction assuming Claim 3.2.:

$$\begin{aligned}\sum_{j \in (t', t]} d_j &< 1 + \sum_{j \in (t', t]} d_j \\ &= 1 + \sum_{i \in I} (N_t(i) - N_{t'-1}(i))\end{aligned}$$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Contradiction assuming Claim 3.2.:

$$\begin{aligned}
\sum_{j \in (t', t]} d_j &< 1 + \sum_{j \in (t', t]} d_j \\
&= 1 + \sum_{i \in I} (N_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - N_{t'-1}(i))
\end{aligned}$$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Contradiction assuming Claim 3.2.:

$$
\begin{aligned}
\sum_{j \in (t',t]} d_j &< 1 + \sum_{j \in (t',t]} d_j \\
&= 1 + \sum_{i \in I} (N_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - P_{t'}(i))
\end{aligned}
$$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Contradiction assuming Claim 3.2.:

$$\begin{aligned}
\sum_{j \in (t', t]} d_j &< 1 + \sum_{j \in (t', t]} d_j \\
&= 1 + \sum_{i \in I} (N_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - P_{t'}(i)) \\
&\leq \sum_{i \in [m]} (P_t(i) - P_{t'}(i))
\end{aligned}$$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Contradiction assuming Claim 3.2.:

$$\begin{aligned}
\sum_{j \in (t',t]} d_j &< 1 + \sum_{j \in (t',t]} d_j \\
&= 1 + \sum_{i \in I} (N_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - N_{t'-1}(i)) \\
&\leq \sum_{i \in I} (P_t(i) - P_{t'}(i)) \\
&\leq \sum_{i \in [m]} (P_t(i) - P_{t'}(i)) \\
&= \sum_{j \in (t',t]} d_j. \quad \square
\end{aligned}$$

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Proof. Fix $i \in I$. Either i is not chosen in $(t', t]$: $P_t(i) \geq N_{t'-1}(i) + 1 = N_t(i) + 1$.

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Proof. Fix $i \in I$. Either i is not chosen in $(t', t]$: $P_t(i) \geq N_{t'-1}(i) + 1 = N_t(i) + 1$.

OR $s_j = i$ for some $j \in (t', t]$: then $j > t' \geq t^- \implies P_t(i) \geq N_t(i)$.

Upper bound proof (last part!)

- Define $I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1\} = \{i \in [m] : D_{t'}(i) \leq t\}$.
- $s_{t'} \notin I, s_j \in I \quad \forall j \in (t', t], \quad I \neq \emptyset$

Claim 3.2.

For any $i \in I$ we have $P_t(i) \geq N_t(i)$ and $P_{t'}(i) \leq N_{t'-1}(i)$.

Proof. Fix $i \in I$. Either i is not chosen in $(t', t]$: $P_t(i) \geq N_{t'-1}(i) + 1 = N_t(i) + 1$.

OR $s_j = i$ for some $j \in (t', t]$: then $j > t' \geq t^- \implies P_t(i) \geq N_t(i)$.

For the second inequality: $D_{t'}(s_{t'}) > t \implies I \cap C(t') = \emptyset$. □

Other conjectures

Non-uniform weights

Let $d \in \mathbb{R}_{>0}^{m\times n}$. For every fractional assignment $x \in [0,1]^{m\times n}$, there is an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,

$$\Big|\sum_{j\in[t]} d_{ij}(x_{ij} - y_{ij})\Big| \leq \max_{i\in[m], j\in[n]} d_{ij}.$$

Other conjectures

Non-uniform weights

Let $d \in \mathbb{R}_{>0}^{m \times n}$. For every fractional assignment $x \in [0,1]^{m \times n}$, there is an assignment $y \in \{0,1\}^{m \times n}$ so that for every $i \in [m]$ and $t \in [n]$,

$$\left| \sum_{j \in [t]} d_{ij}(x_{ij} - y_{ij}) \right| \leq \max_{i \in [m], j \in [n]} d_{ij}.$$

Weighted committee assignment

Let $d \in \mathbb{R}_{>0}^{n}$. For every $x \in [0,1]^{m \times n}$ so that $n_j := \sum_{i \in [m]} x_{ij} \in \mathbb{N}$ for all $j \in [n]$, there is $y \in \{0,1\}^{m \times n}$ so that $\sum_{i \in [m]} y_{ij} = n_j$ for all $j \in [n]$ and

$$\left| \sum_{j \in [t]} d_j(x_{ij} - y_{ij}) \right| \leq \max_{j \in [n]} d_j \quad \forall i \in [m], t \in [n].$$

Other conjectures

Non-uniform weights

Let $d \in \mathbb{R}^{m\times n}_{>0}$. For every fractional assignment $x \in [0,1]^{m\times n}$, there is an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,

$$\Big|\sum_{j\in[t]} d_{ij}(x_{ij} - y_{ij})\Big| \leq \max_{i\in[m],j\in[n]} d_{ij}.$$

Weighted committee assignment

Let $d \in \mathbb{R}^{n}_{>0}$. For every $x \in [0,1]^{m\times n}$ so that $n_j := \sum_{i\in[m]} x_{ij} \in \mathbb{N}$ for all $j \in [n]$, there is $y \in \{0,1\}^{m\times n}$ so that $\sum_{i\in[m]} y_{ij} = n_j$ for all $j \in [n]$ and

$$\Big|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})\Big| \leq \max_{j\in[n]} d_j \quad \forall i \in [m], t \in [n].$$

Thanks for your attention!