Scheduling for Weighted Flow and Completion Times in Reconfigurable Networks

Michael Dinitz

JOHNS HOPKINS UNIVERSITY

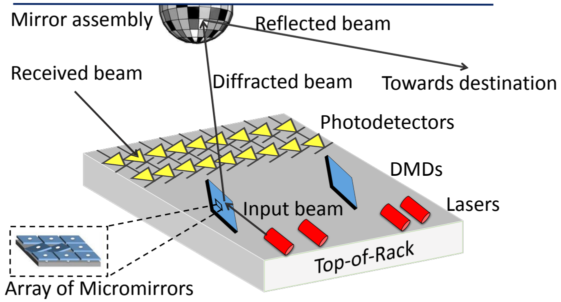
Benjamin Moseley

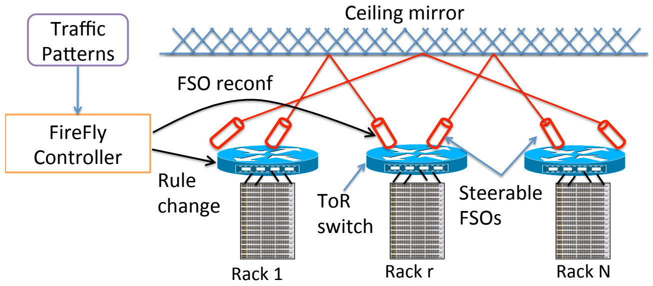
Carnegie Mellon University

Reconfigurable Networks

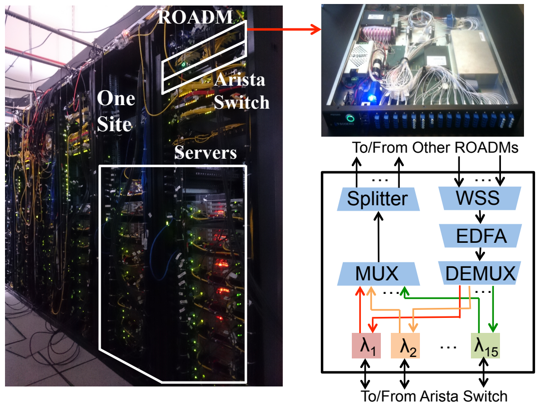
Can change network topology in software!

Datacenters





Optical WANs



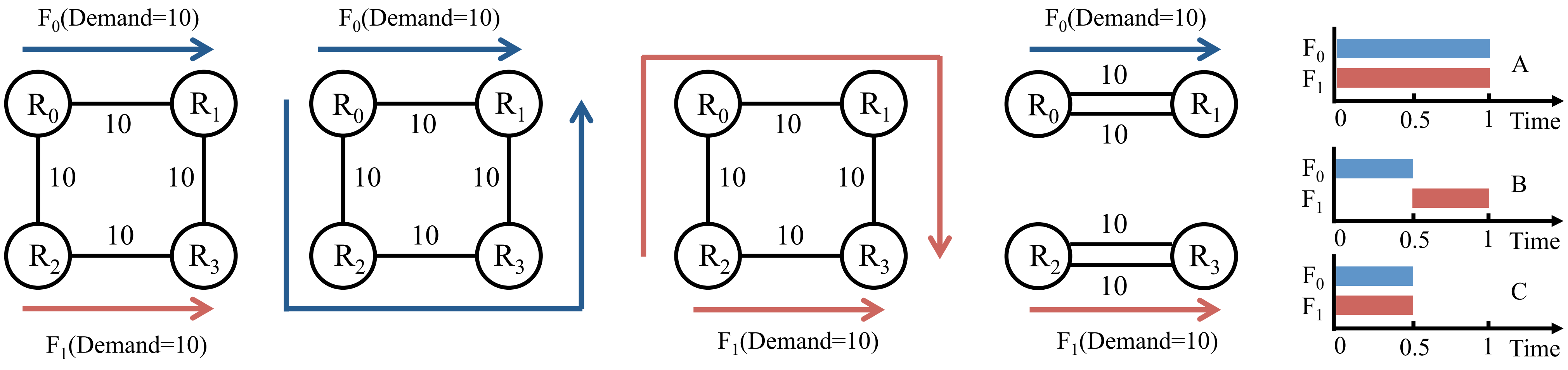
(a) Testbed.

(b) ROADM.

Many constraints depending on technology

Always: degree bounds

Reconfiguration Can Be Helpful



(a) Plan A. (b) Plan B-1. (c) Plan B-2. (d) Plan C. (e) Time series.

Image and example from Jin et al, SIGCOMM '16

Scheduling Bulk Transfers

System:

- Optimizing Bulk Transfers with Software-Defined Optical WAN [Jin et al. SIGCOMM ‘16]

Theory:

- Competitive Analysis for Online Scheduling in Software-Defined Optical WAN [Jia et al. INFOCOM ‘17]

Given bulk transfers (online), how should we schedule transfers & reconfigurations?

Model [Jia et al.]

Start:

- Nodes V, degree bounds d_v for each $v \in V$
- Transfers (jobs) S

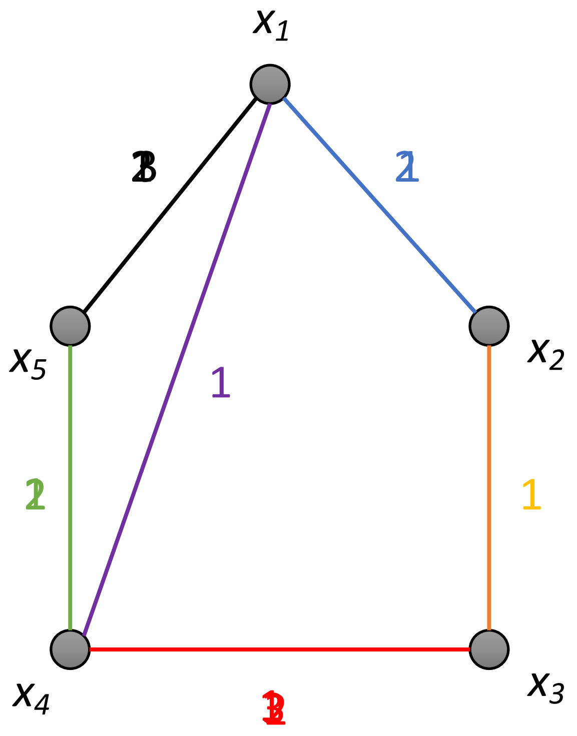
Transfer (job) i:

- Release time r_i, source u_i, destination v_i, size l_i, weight w_i (not in Jia et al)

Time t:

- Create graph $G_t = (V, E_t)$ obeying degree bounds
  - E_t subset of transfers S
- One unit of progress on jobs in E_t

Example



$d_v = 1$ for all v

Transfer	Release	Source	Destination	Size
1	1	x_1	x_5	3
2	1	x_1	x_2	2
3	1	x_2	x_3	1
4	2	x_5	x_4	2
5	2	x_4	x_3	3
6	4	x_1	x_4	1

Issues with Model

- No constraints on graphs other than degrees
  - Optical WANs: real constraints based on optical network
  - Datacenters: depending on technology

- Can only send data over direct connections
  - OWAN system uses multihop paths

Still a good start!

Objectives and Results (Jia et al)

Given schedule, each transfer *i* has **completion time** C_i

Makespan

- $max_i\ C_i$
- Time when last job completes
- 3-competitive algorithm

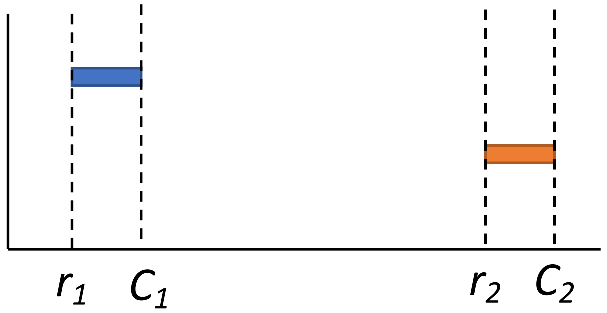
Sum of Completion Times

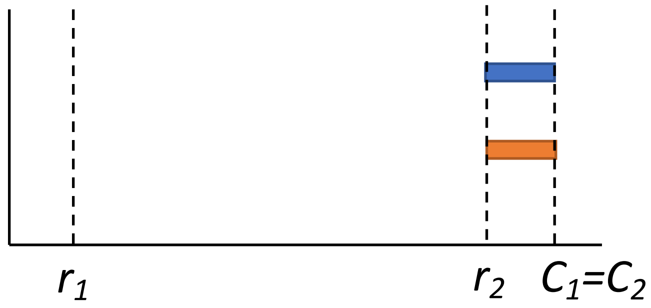
- $\sum_i C_i$
- 3α-competitive algorithm
    - α competitive ratio of SRPT for d-machine scheduling
    - At most 1.86
    - Assumes $d_v = d$ for all v

α-competitive: at most α factor worse than offline optimum

Flow Time

In online setting, do these objectives make sense?





Makespan unchanged, sum of completion times only doubled!

New Objective: Sum of (Weighted) **Flow Times**

- Flow time of job *i*: $F_i = C_i - r_i$
- Sojourn time, waiting time, response time
- $\sum_i w_i(C_i - r_i)$

Our Results:

Lower bound: Every online algorithm has competitive ratio at least $\Omega(\sqrt{n})$

Upper bound: need **resource augmentation / speedup**

- Allow faster transfer compared to OPT
  - Our solution uses 200 Gbps links, compare to OPT using 100Gbps links
- *$O(1/\varepsilon^2)$*-competitive algorithm with *$(2+\varepsilon)$*-speedup

Corollary: *O(1)*-competitive algorithm for **weighted** sum of completion times, **different** degree bounds (no speedup)

Algorithm: Highest-Density First

- Density of job i: $h_i = \frac{w_i}{l_i}$
- At time *t*:
    - Order jobs in nonincreasing order of density
    - Schedule job *i* (add $u_i - v_i$ edge) if u_i and v_i not already full

Easy to state, tricky to analyze!

- Reduce to unit-length jobs (via “fractional” flow time): cost $O(1/\varepsilon)$
- Dual Fitting: cost $O(1/\varepsilon)$

LP relaxation (unit length)

$$
\begin{aligned}
\min \quad & \sum_{i \in S} \sum_{t \geq r_i} w_i (t - r_i) x_{i,t} & \\
\text{s.t.} \quad & \sum_{t \geq r_i} x_{i,t} \geq 1 & \forall i \in S \\
& \sum_{i \in S : |\{u_i, v_i\} \cap \{w\}| = 1} x_{i,t} \leq d_w & \forall w \in V, \ \forall t \in \mathbb{N} \\
& x_{i,t} \geq 0 & \forall i \in S, \ \forall t \in \mathbb{N}
\end{aligned}
$$

Weighted flow time

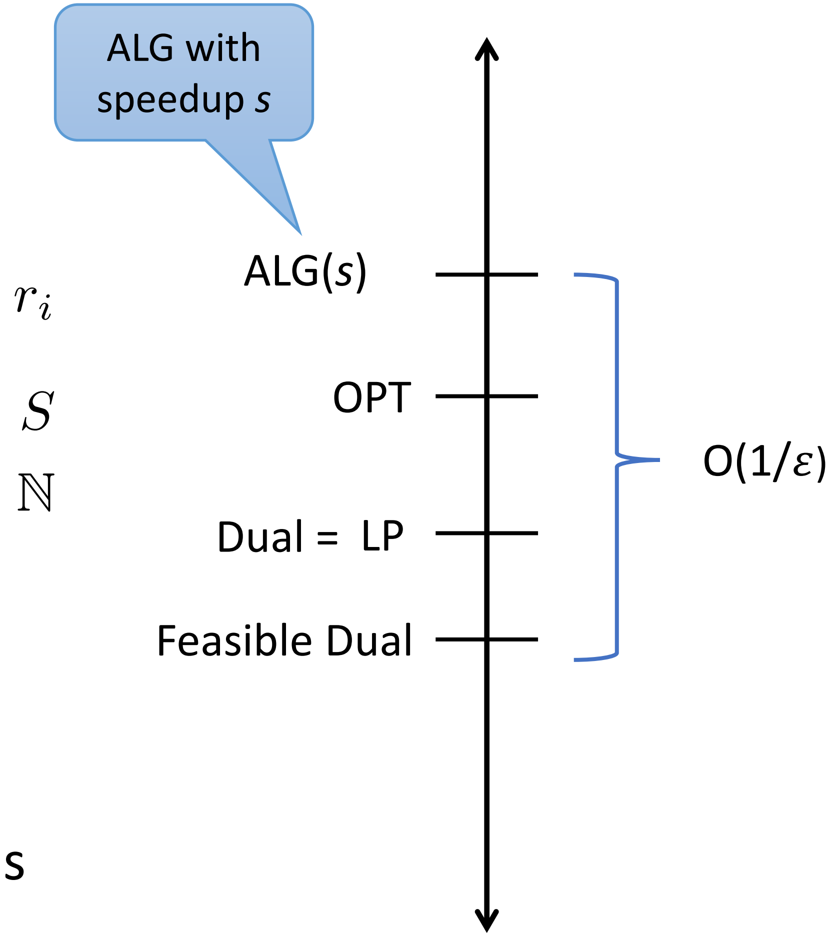
Every job gets scheduled

Degree bounds

1 if job *i* scheduled at time *t*

Dual

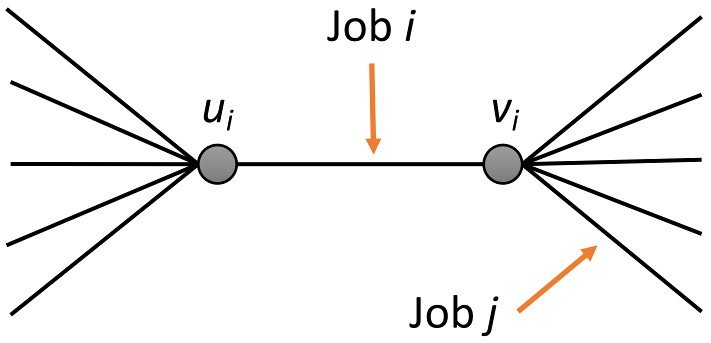
$$
\begin{aligned}
\max \quad & \sum_{i \in S} \alpha_i - \sum_{u \in V} \sum_{t \in \mathbb{N}} \beta_{u,t} & \\
\text{s.t.} \quad & \alpha_i - \frac{\beta_{u_i,t}}{d_{u_i}} - \frac{\beta_{v_i,t}}{d_{v_i}} \leq w_i(t - r_i) & \forall i \in S,\ \forall t \geq r_i \\
& \alpha_i \geq 0 & \forall u \in S \\
& \beta_{i,t} \geq 0 & \forall i \in S,\ \forall t \in \mathbb{N}
\end{aligned}
$$



- Dual fitting: common in flow time scheduling problems
- Intuition:
  - α_i = increase in algorithm's cost due to transfer *i* when it is released
  - $\beta_{u,t}$ = remaining work at node *u* at time *t*

Dual Solution: α

α_i = increase in algorithm's cost due to transfer *i* when it is released



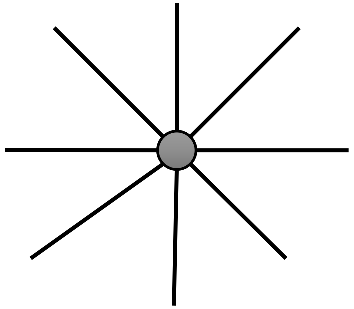
Job *j* with $w_j > w_i$: scheduled before *i* ⇒ increase in total weighted flow is w_i

Job *j* with $w_j < w_i$: scheduled after *i* ⇒ increase in total weighted flow is w_j

$$\alpha_i := \frac{1}{2s}\left(\frac{1}{d_{u_i}}\left(\sum_{j \in U_i(r_i): w_i < w_j} w_i + \sum_{j \in U_i(r_i): w_i > w_j} w_j\right) + \frac{1}{d_{v_i}}\left(\sum_{j \in V_i(r_i): w_i < w_j} w_i + \sum_{j \in V_i(r_i): w_i > w_j} w_j\right)\right)$$

Dual Solution: β

$\beta_{u,t}$ = remaining work at node *u* at time *t*



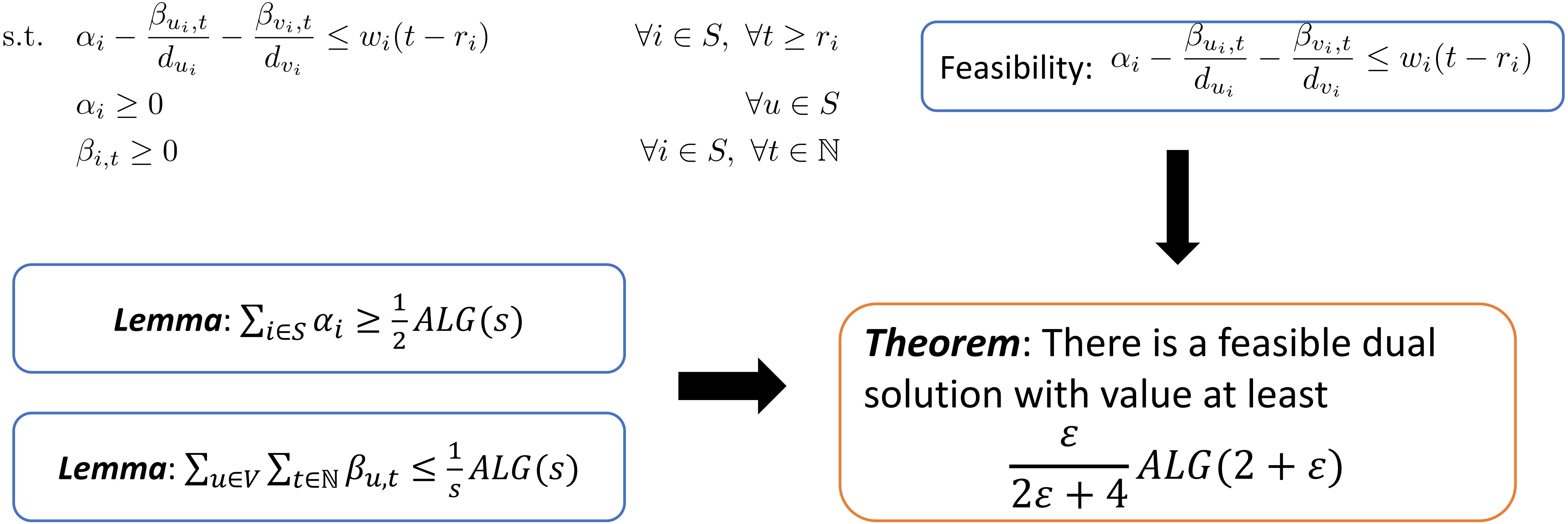
$$\beta_{u,t} = \frac{w_u(t)}{2s}$$

Total weight of jobs at *u* at time *t*

Speedup (2+ε)

Main Result

$$\begin{aligned}
\max \quad & \sum_{i\in S}\alpha_i - \sum_{u\in V}\sum_{t\in\mathbb{N}}\beta_{u,t} & \\
\text{s.t.} \quad & \alpha_i - \frac{\beta_{u_i,t}}{d_{u_i}} - \frac{\beta_{v_i,t}}{d_{v_i}} \le w_i(t-r_i) & \forall i\in S,\ \forall t \ge r_i \\
& \alpha_i \ge 0 & \forall u \in S \\
& \beta_{i,t} \ge 0 & \forall i\in S,\ \forall t\in\mathbb{N}
\end{aligned}$$



Feasibility: $\alpha_i - \frac{\beta_{u_i,t}}{d_{u_i}} - \frac{\beta_{v_i,t}}{d_{v_i}} \le w_i(t-r_i)$

Lemma: $\sum_{i\in S}\alpha_i \ge \frac{1}{2}ALG(s)$

Lemma: $\sum_{u\in V}\sum_{t\in\mathbb{N}}\beta_{u,t} \le \frac{1}{s}ALG(s)$

Theorem: There is a feasible dual solution with value at least

$$\frac{\varepsilon}{2\varepsilon+4}ALG(2+\varepsilon)$$

Conclusion & Open Questions

Our work:

- Model of scheduling transfers in reconfigurable networks from Jia et al. [INFOCOM '17]
- In online setting, flow times make more sense than completion times
- First nontrivial approx for flow times, with small speedup (necessary)
- Corollary: first O(1)-competitive algorithm for completion times

Future work:

- More realistic model of reconfigurable networks!
- Speedup $1+\varepsilon$ instead of $2+\varepsilon$?

Thanks!