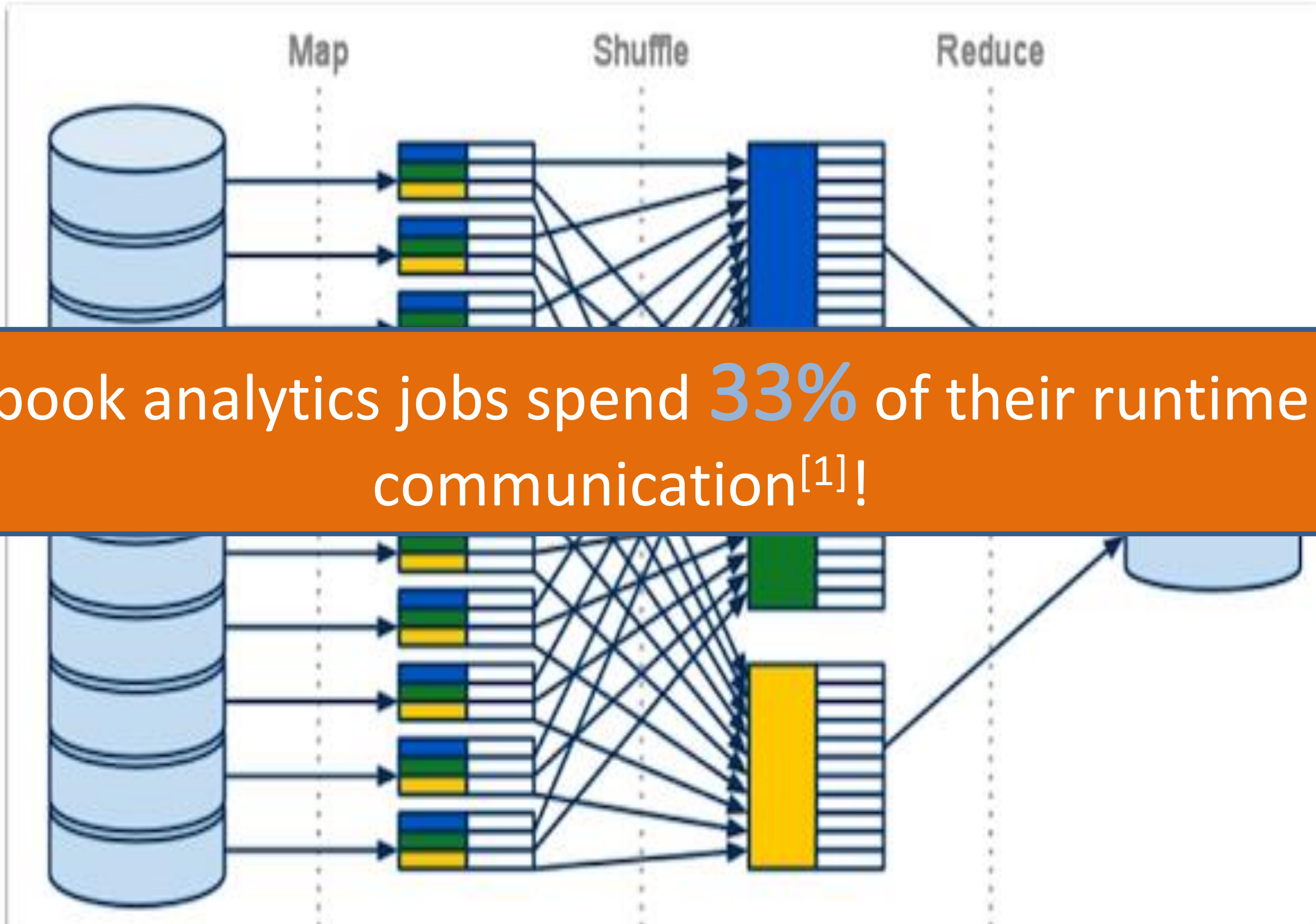
On Scheduling Coflows[1]

Saba Ahmadi, Samir Khuller, Manish Purohit, Sheng Yang

[1] Appeared in IPCO 2017

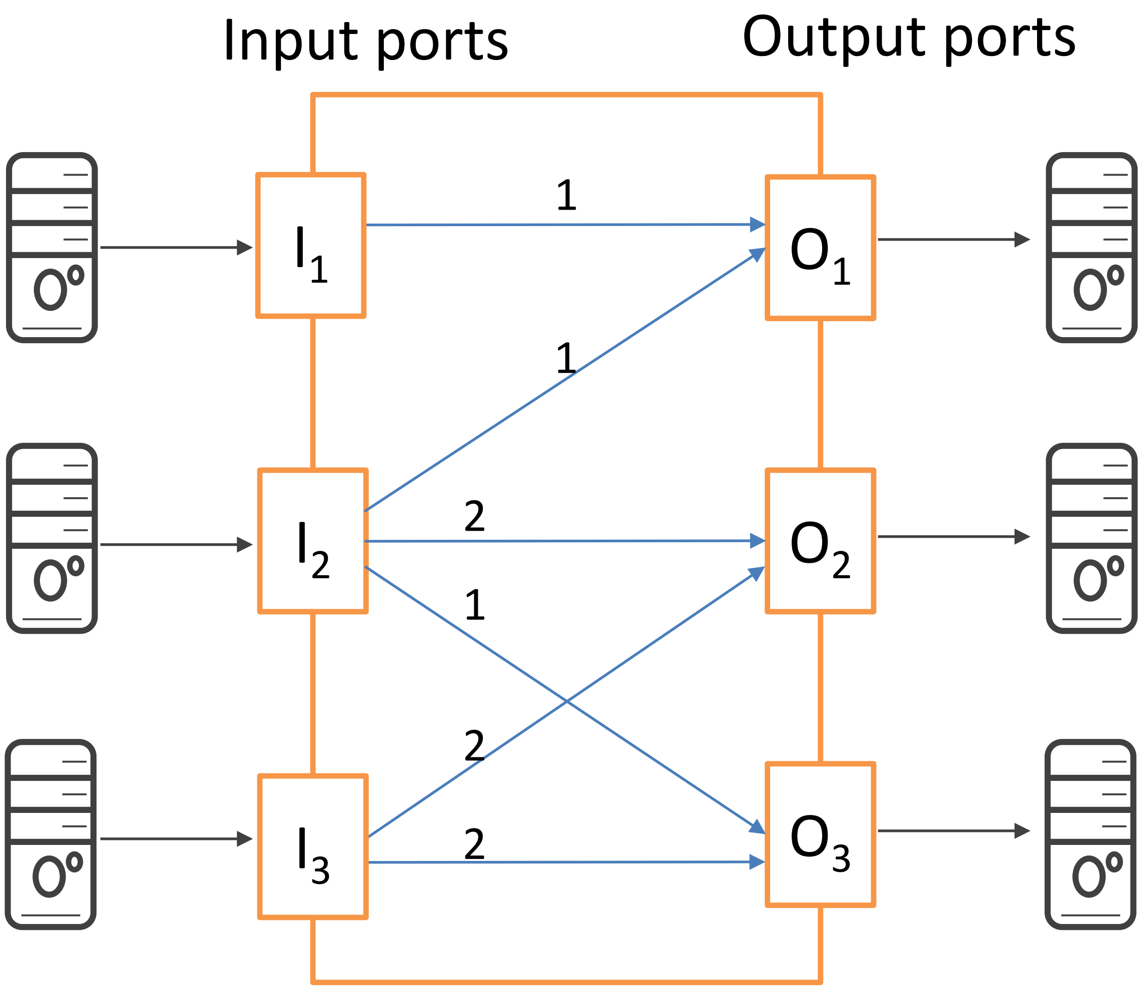
Communication is Crucial!



Facebook analytics jobs spend **33%** of their runtime in communication[1]!

[1] Chowdhury et al. Managing Data Transfers in Computer Clusters with Orchestra, SIGCOMM’2011

Model[1]



- m×m switch
- Coflows: Collection of Parallel flows for a Common Goal
- Coflow j is presented by a matrix D^j

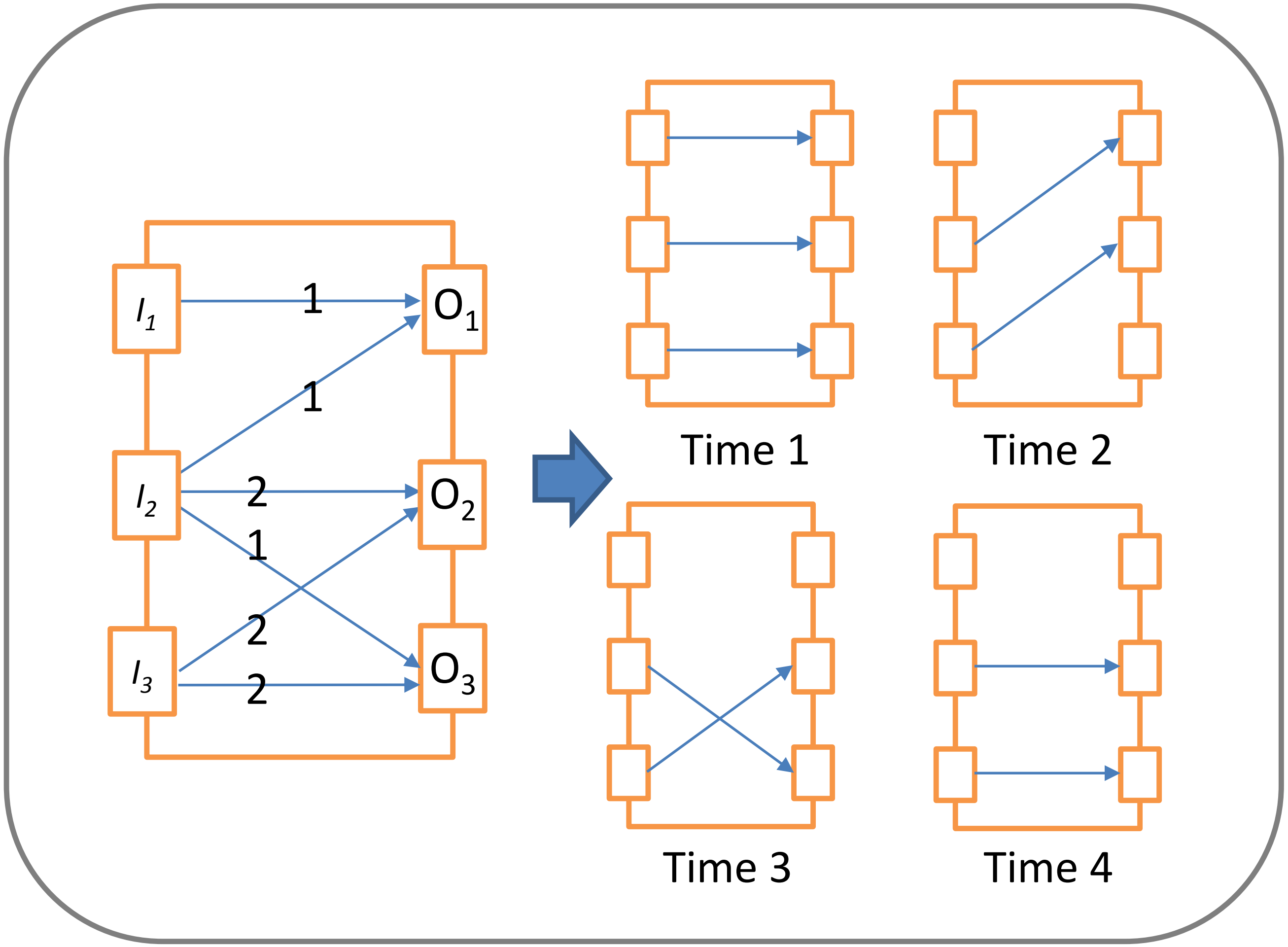
$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 2 \end{bmatrix}$$

- Capacity Constraint 1 at all ports.
- At any time slot, scheduled flows form a matching.

$$L_{ij} = deg_{G_j}(i)$$

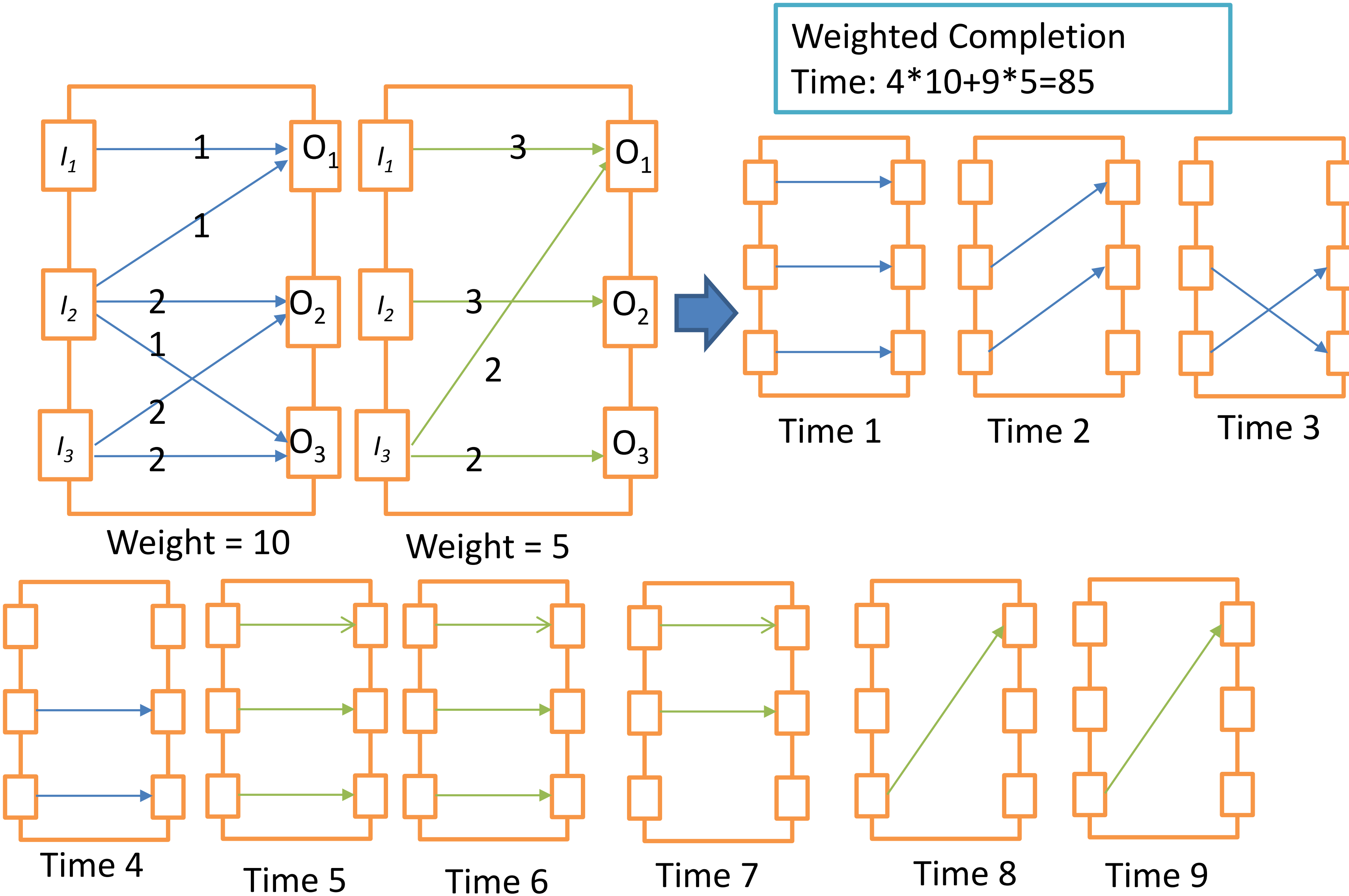
[1] Chowdhury et al. "Coflow: A networking abstraction for cluster applications.(HOTNETS'12)

Model: Scheduling a Single Coflow



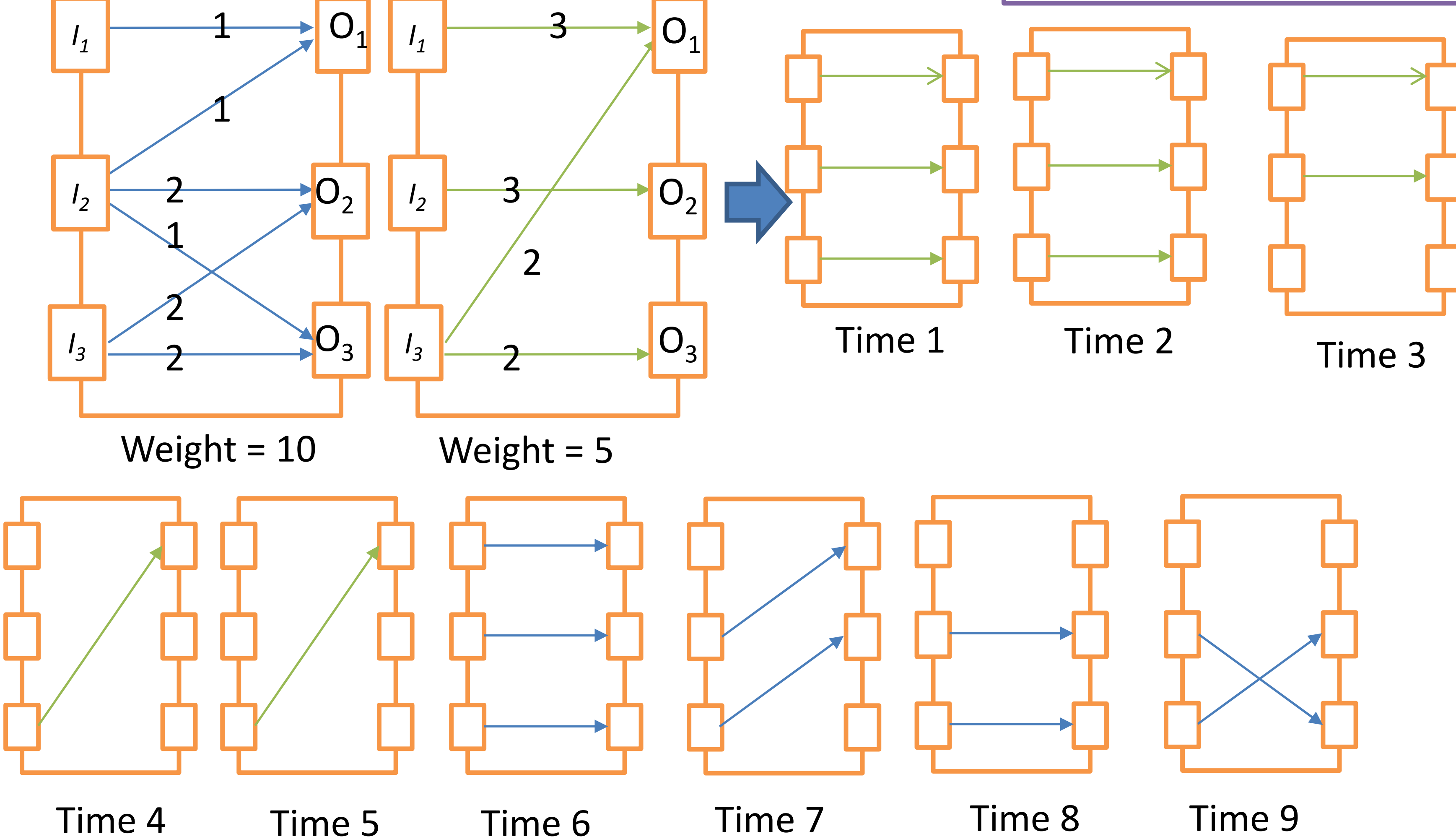
Application of Hall's Theorem

Model: Scheduling Multiple Coflows



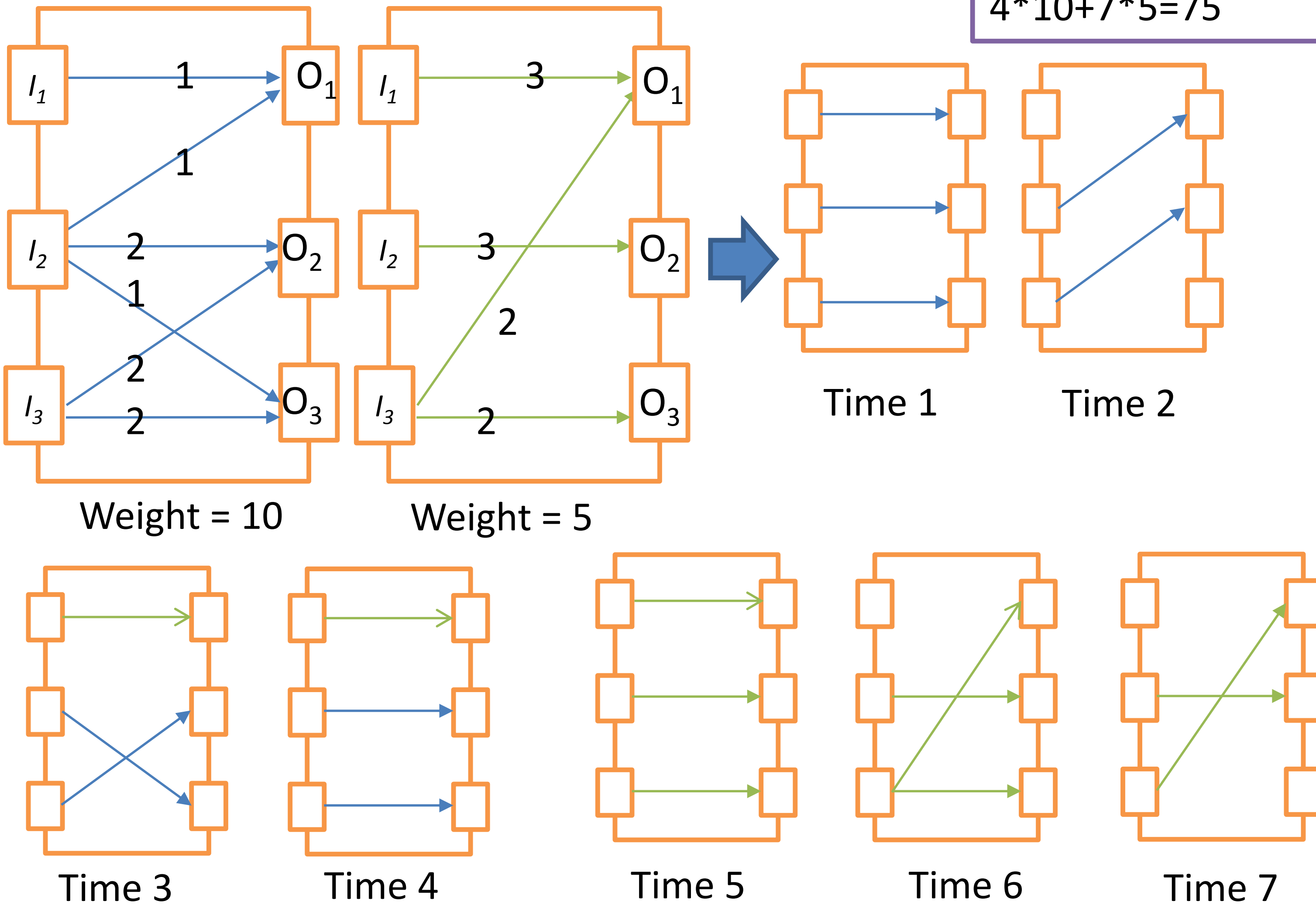
Model: Scheduling Multiple Coflows

Weighted Completion Time:
5*5+9*10=115



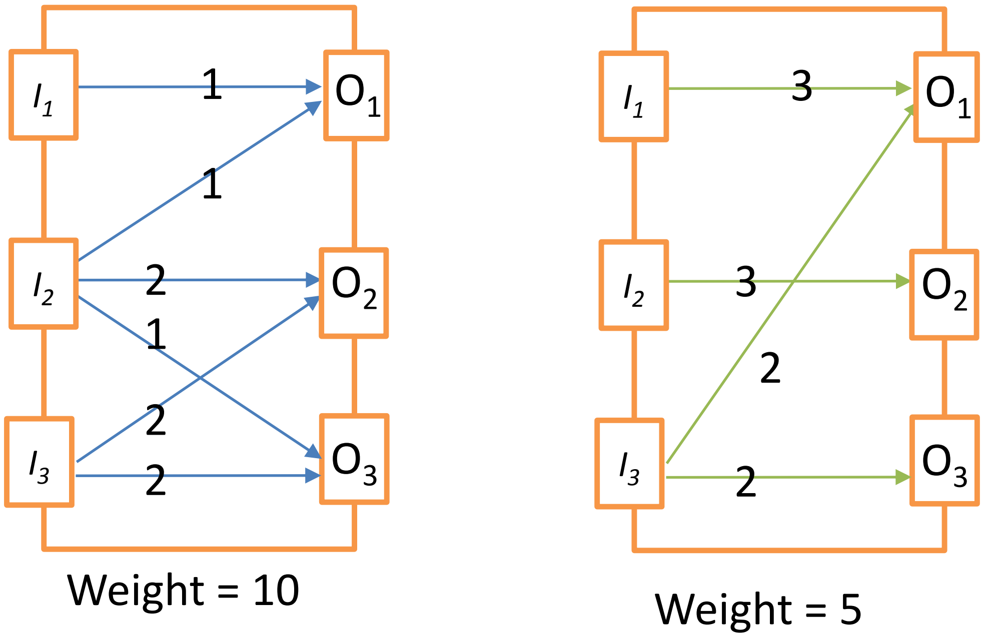
Model: Scheduling Multiple Coflows

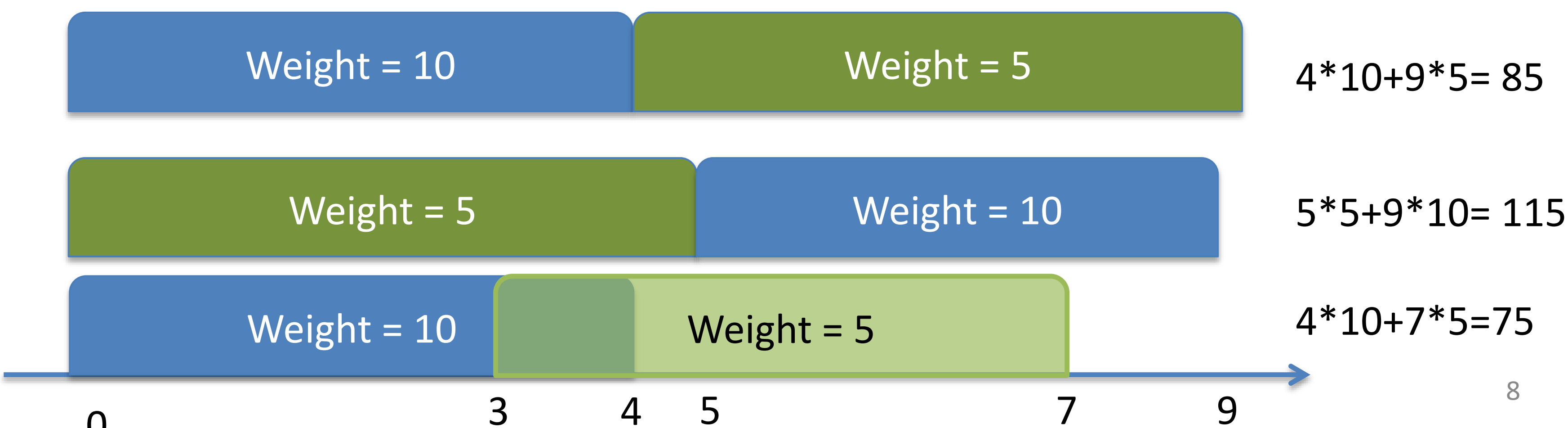
Weighted Completion Time:
4*10+7*5=75



Model

Given n coflows, find a feasible schedule which minimizes total weighted completion time, i.e $\sum_{j=1}^{n} w_j C_j$





Prior Work

Systems

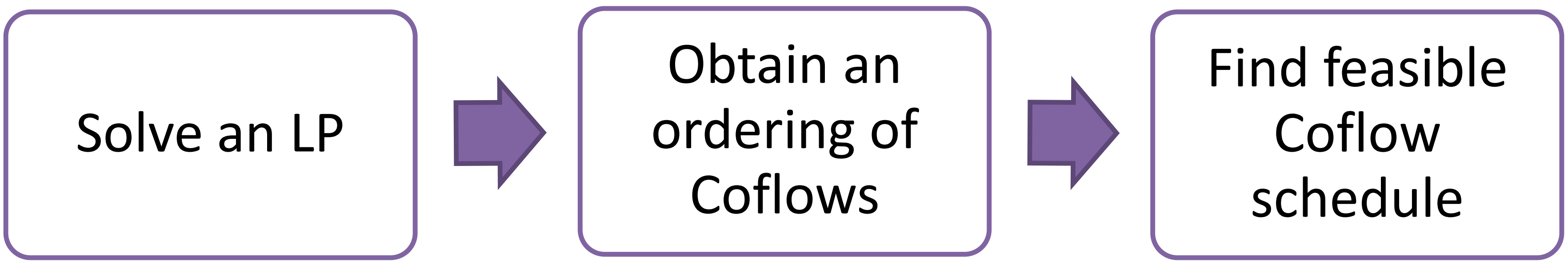
- Chowdhury and Stoica (HOTNETS 2012), Coflow: A Networking Abstraction for Cluster Applications
- Chowdhury, Zhong and Stoica (SIGCOMM 2014), Efficient Coflow Scheduling with Varys
- Chowdhury and Stoica (SIGCOMM 2015), Effective Coflow Scheduling Without Prior Knowledge

Prior Work

Theory

	Deterministic, No release time	Deterministic, With release time	Randomized, No release time	Randomized, With release time
Qiu,Stein and Zhong- SPAA15	64/3	67/3	15.5	16.5
Khuller and Purohit-SPAA16	8	12	$3+\sqrt{2}$	
This paper-IPCO17	4	5		
Shafiee, Ghaderi – SPAA17	4	5		

Overview of our Approach



Step 1: Solve an LP

$$L_{ij} = deg_{G_j}(i)$$

$$M = I \cup O$$

$$\min \sum_{j \in J} w_j C_j$$

subject to, $\quad C_j \geq r_j + L_{i,j} \qquad \forall j \in J, \forall i \in M$

$$\sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2}\left(\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2\right) \qquad \forall i \in M, \forall S \subseteq J$$

How can we get the second constraint?

Step 1: Solve an LP

[1] $$\sum_{j\in S} L_{i,j}C_j \geq \frac{1}{2}\Big(\sum_{j\in S} L_{i,j}^2 + (\sum_{j\in S} L_{i,j})^2\Big) \qquad \forall S \subseteq J, \forall i \in M$$

$S \subseteq J$
w.l.o.g assume:
$S = \{1, 2, \cdots, x\}$
$C_1 \leq C_2 \leq \cdots \leq C_x$

$$\begin{aligned} C_1 &\geq L_{i,1} \\ C_2 &\geq L_{i,1} + L_{i,2} \\ C_3 &\geq L_{i,1} + L_{i,2} + L_{i,3} \\ &\cdots \\ C_x &\geq L_{i,1} + L_{i,2} + \cdots + L_{i,x} \end{aligned}$$

$$\begin{aligned} L_{i,1}C_1 &\geq L_{i,1}(L_{i,1}) \\ L_{i,2}C_2 &\geq L_{i,2}(L_{i,1} + L_{i,2}) \\ L_{i,3}C_3 &\geq L_{i,3}(L_{i,1} + L_{i,2} + L_{i,3}) \\ &\cdots \\ L_{i,x}C_x &\geq L_{i,x}(L_{i,1} + L_{i,2} + \cdots + L_{i,x}) \end{aligned}$$

$$\begin{aligned} L_{i,1}C_1 &\geq L_{i,1}^2 \\ L_{i,2}C_2 &\geq L_{i,2}L_{i,1} + L_{i,2}^2 \\ L_{i,3}C_3 &\geq L_{i,3}L_{i,1} + L_{i,3}L_{i,2} + L_{i,3}^2 \\ &\cdots \\ L_{i,x}C_x &\geq L_{i,x}L_{i,1} + L_{i,x}L_{i,2} + \cdots + L_{i,x}^2 \end{aligned}$$

$$\sum_{j\in S} L_{i,j}C_j \geq \sum_{j\in S} L_{i,j}^2 + \sum_{j<k, j,k\in S} L_{i,j}L_{i,k} = \frac{1}{2}\Big(\sum_{j\in S} L_{i,j}^2 + (\sum_{j\in S} L_{i,j})^2\Big)$$

Second Constraint

[1] M. Queyranne. Structure of a simple scheduling polyhedron. Mathematical Programming, 1993.

Step 1: Solve the LP

$$\min \sum_{j \in J} w_j C_j$$

$$\text{subject to,} \quad C_j \geq r_j + L_{i,j} \qquad \forall j \in J, \forall i \in M$$

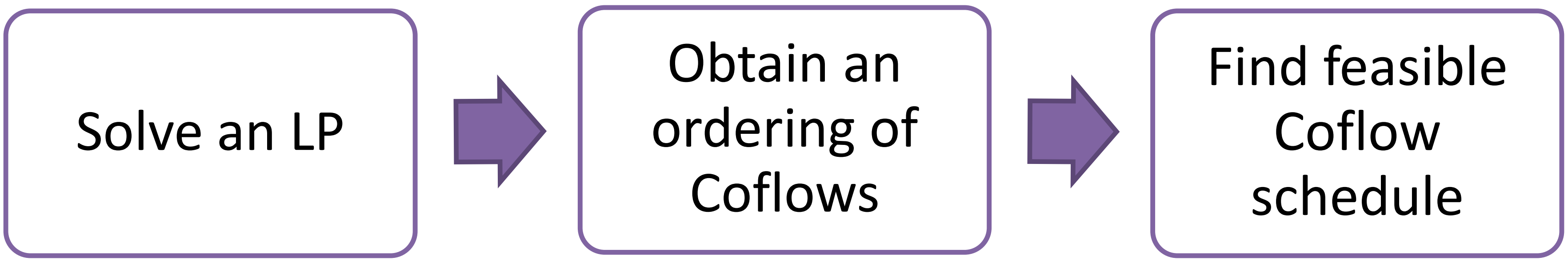
$$\sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2} \left(\sum_{j \in S} L_{i,j}^2 + \big(\sum_{j \in S} L_{i,j}\big)^2 \right) \qquad \forall i \in M, \forall S \subseteq J$$

The number of constraints is exponential, we can solve this LP using ellipsoid method.

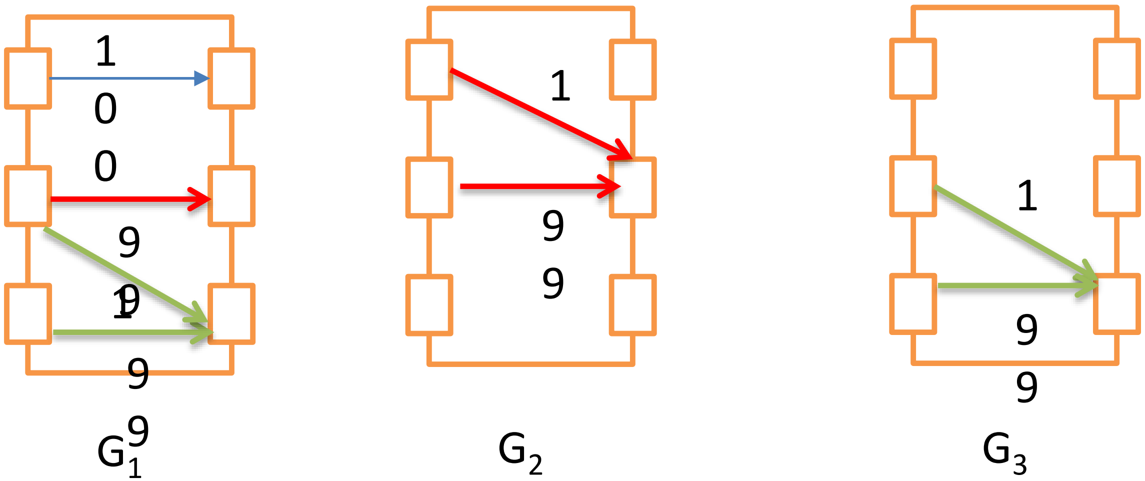
Separation Oracle:
Given a solution C, w.l.o.g assume $C_1 \leq C_2 \cdots \leq C_n$. Let $S_1 = \{1\}$, $S_2 = \{1, 2\}$,..., $S_n = \{1, 2, \cdots, n\}$. It's sufficient to check whether the constraints are violated for the n sets $S_1, \cdots, S_n$.

Not Practical!

Overview of our Approach



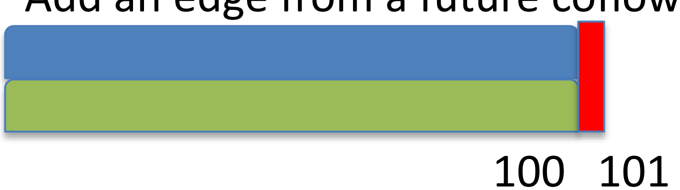
Step 3: Find a Feasible Coflow Schedule



Scheduling coflows sequentially:

Total weighted completion time = 100*1+200*1+300*1=600

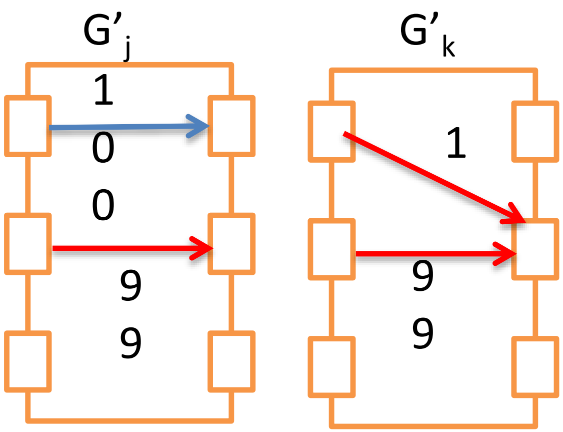
Add an edge from a future coflow if it does not increase degree of the current coflow.



Total weighted Completion Time:
100*1+101*1+100*1 =301

Algorithm

Solve the LP to get the completion times of Coflows.
Assume $C_1 \le C_2 \cdots \le C_n$
Consider the Coflows in the same order: $G_1, \cdots, G_n$

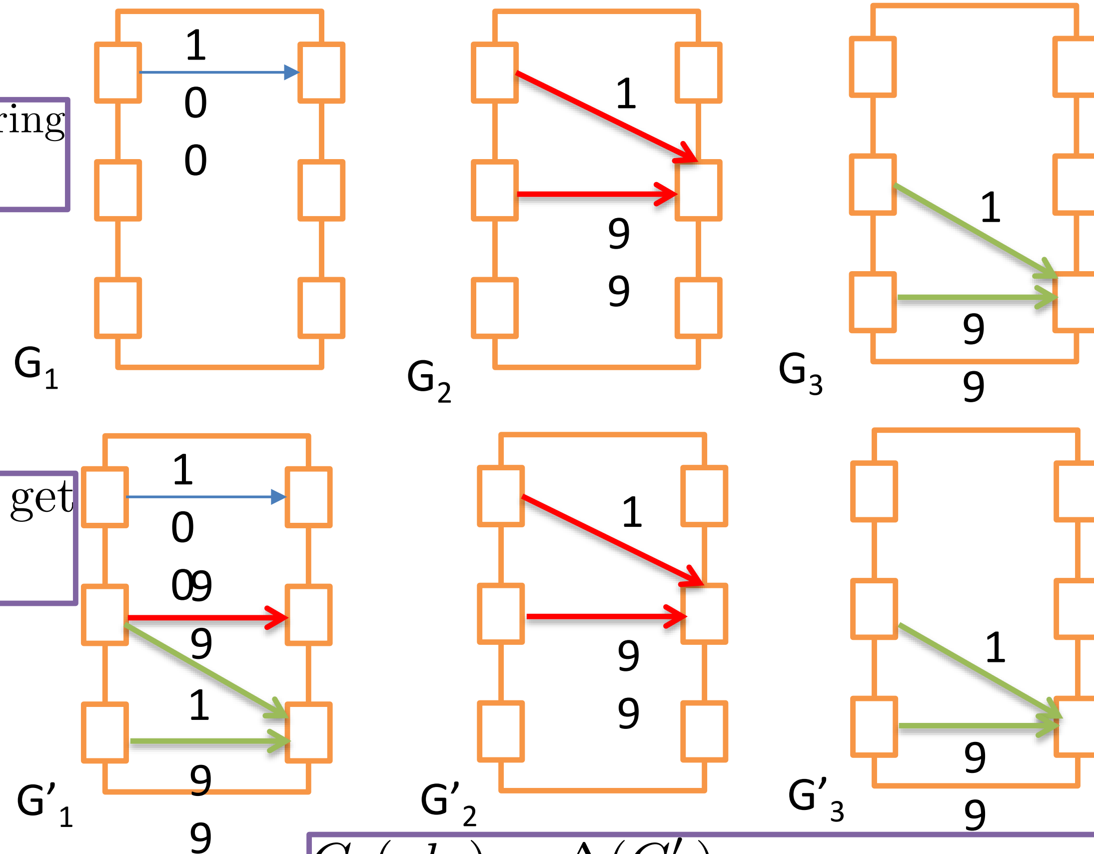


for $j = 1, 2, \cdots, n$:
$G'_j = G_j$
for $j = 1, 2, \cdots, n-1$:
for $k = j+1, j+2, \cdots, n$:
Move $e \in G'_k$ to G'_j if $\Delta(G'_j)$ does not increase.

Schedule new Coflows $G'_1, \cdots, G'_n$ using application of Hall's Theorem.

Step 3: Find a Feasible Coflow Schedule

Solve the LP to get the ordering $G_1, G_2, \cdots, G_n$



Move edges backward to get $G'_1, G'_2, \cdots, G'_n$

Schedule coflows $G'_1, G'_2, \cdots, G'_n$ using application of Hall's theorem.

$$C_1(alg) = \Delta(G'_1)$$

$$C_2(alg) \leq \Delta(G'_1) + \Delta(G'_2)$$

$$C_3(alg) \leq \Delta(G'_1) + \Delta(G'_2) + \Delta(G'_3)$$

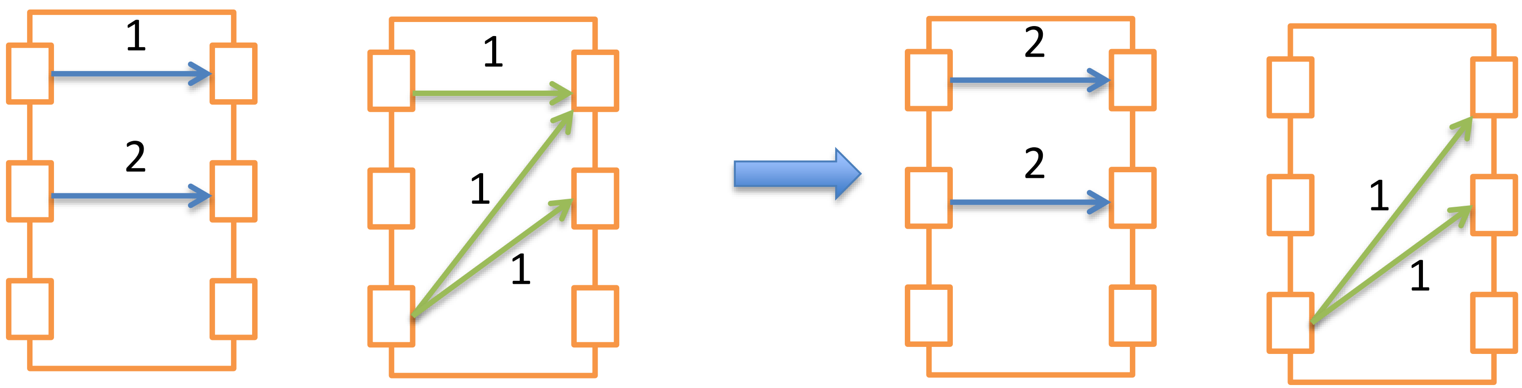
$$C_j(alg) \leq \sum_{k \leq j} \Delta(G'_k)$$

Cost of Algorithm

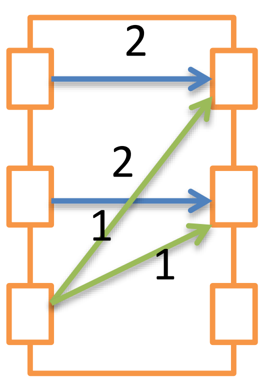
$G_1, \cdots, G_n$: Initial Coflows

$G'_1, \cdots, G'_n$: Coflows after moving edges backward.

$$\text{Lemma 1: For all } j \in \{1, 2, \cdots, n\}, \sum_{k \leq j} \Delta(G'_k) \leq 2\Delta(\bigcup_{k \leq j} G_k)$$



G_1 G_2 G'_1 G'_2



$G_1 \cup G_2$

$j = 2:$

$$\sum_{k \leq j} \Delta(G'_k) = \Delta(G'_1) + \Delta(G'_2) = 2 + 2 = 4$$

$$\Delta(\bigcup_{k \leq j} G_k) = \Delta(G_1 \cup G_2) = 3$$

$$\rightarrow \sum_{k \leq j} \Delta(G'_k) \leq 2\Delta(\bigcup_{k \leq j} G_k)$$

Cost of Algorithm:

$$\min \sum_{j \in J} w_j C_j$$

subject to, $C_j \geq r_j + L_{i,j}$ $\forall j \in J, \forall i \in M$

$$\sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2}\left(\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2\right) \quad \forall i \in M, \forall S \subseteq J$$

Lemma 1: For all $j \in \{1, 2, \cdots, n\}$, $\sum_{k \leq j} \Delta(G'_k) \leq 2\Delta(\bigcup_{k \leq j} G_k)$

Lemma 2: $C_j \geq \frac{1}{2} \max_i \sum_{k=1}^{j} L_{i,k} = \frac{1}{2}\Delta(\bigcup_{k \leq j} G_k)$

$L_{ij} = deg_{G_j}(i)$

Theorem 1:

$$C_j(alg) = \sum_{k \leq j} \Delta(G'_k) \leq 2\Delta(\bigcup_{k \leq j} G_k) \leq 4C_j$$

$$\sum_{j=1}^{n} w_j C_j(alg) \leq 4 \sum_{j=1}^{n} w_j C_j \leq 4OPT$$

Theorem 2: There exists a deterministic combinatorial 5 approximation algorithm for coflow scheduling with release times.

Proof of Lemma 1:

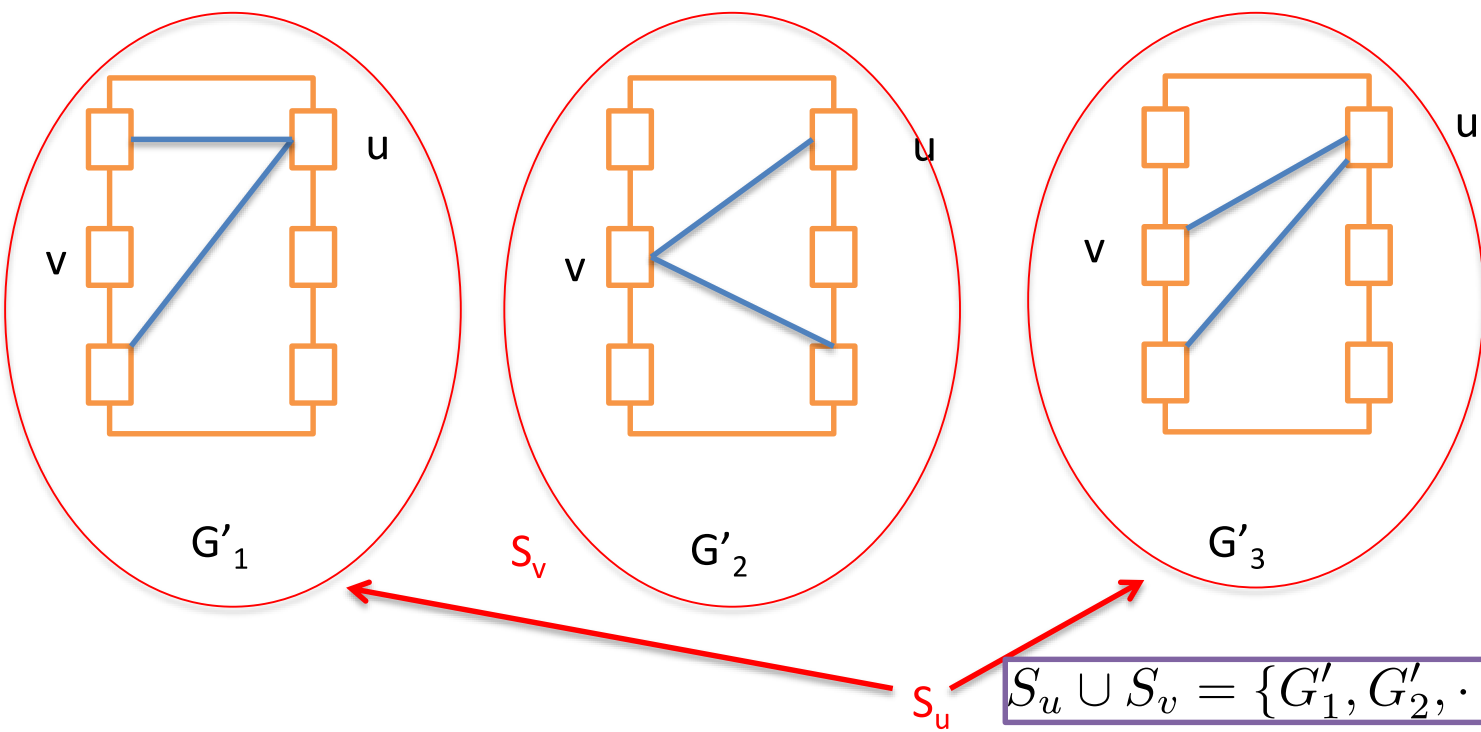
$G_1, \cdots, G_n$: Initial Coflows
$G'_1, \cdots, G'_n$: Coflows after moving edges backward.

Lemma 1: For all $j \in \{1, 2, \cdots, n\}$, $\sum_{k \le j} \Delta(G'_k) \le 2\Delta(\bigcup_{k \le j} G_k)$

Proof: For the simple case when $j = n$ we'll show:

$$\sum_{k \le n} \Delta(G'_k) \le 2\Delta(\bigcup_{k \le n} G_k)$$

After all edges are moved backward, consider a vertex of maximum degree in G'_n. Assume it's u.



$S_u \cup S_v = \{G'_1, G'_2, \cdots, G'_n\}$

$$\bar{G_n} = \bigcup_{k \le n} G_k$$

$G_1, \cdots, G_n$: Initial Coflows
$G'_1, \cdots, G'_n$: Coflows after moving edges backward.

$S_u \subseteq \{G'_1, G'_2, \cdots, G'_n\}$ is the set of modified coflows (after moving edges backward) where u has max degree.
$S_v \subseteq \{G'_1, G'_2, \cdots, G'_n\}$ is the set of modified coflows where v has max degree.
$S_u \cup S_v = \{G'_1, G'_2, \cdots, G'_n\}$

$$\Delta(\bar{G_n}) \ge \max\{deg_{\bar{G_n}}(u), deg_{\bar{G_n}}(v)\}$$

$$deg_{\bar{G_n}}(u) \ge \sum_{G \in S_u} deg_G(u)$$

$$deg_{\bar{G_n}}(v) \ge \sum_{G \in S_v} deg_G(v)$$

$$\Delta(\bar{G_n}) \ge \max\{\sum_{G \in S_u} deg_G(u), \sum_{G \in S_v} deg_G(v)\}$$

$$= \max\{\sum_{G \in S_u} \Delta(G), \sum_{G \in S_v} \Delta(G)\}$$

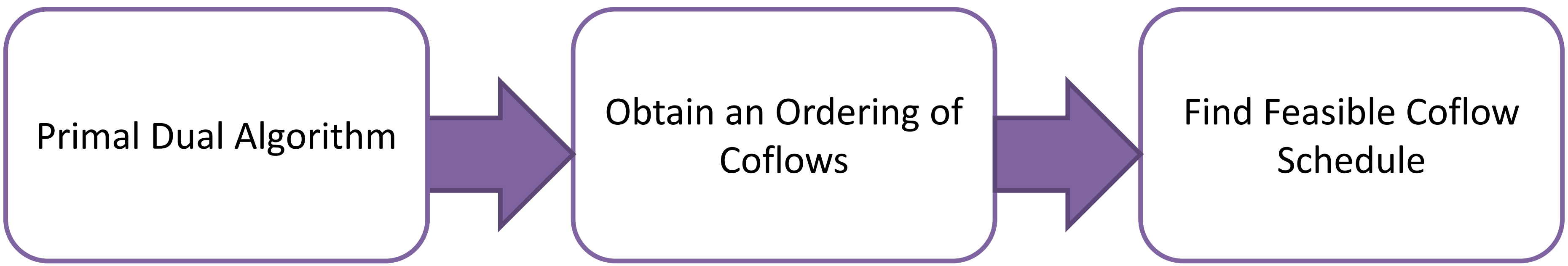
$$\sum_{k \le n} \Delta(G'_k) = \sum_{G \in S_u} \Delta(G) + \sum_{G \in S_v} \Delta(G)$$

$$\le 2(\max\{\sum_{G \in S_u} \Delta(G), \sum_{G \in S_v} \Delta(G)\})$$

$$\sum_{k \le n} \Delta(G'_k) \le 2(\Delta(\bar{G_n})) = 2(\Delta(\bigcup_{k \le n} G_k))$$

Can we make it combinatorial?

Overview of Our Approach



Inspired by
Davis, Gandhi and Kothari [2013]
Mastrolilli, Queyranne, Schulz, Svensson and Uhan [2010]

Primal Dual Algorithm

Primal LP:

$$\min \sum_{j \in J} w_j C_j$$

$$\text{subject to,} \quad C_j \geq r_j + L_{i,j} \qquad \forall j \in J, \forall i \in M$$

$$\sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2}\left(\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2\right) \qquad \forall i \in M, \forall S \subseteq J$$

$$\min \sum_{j \in J} w_j C_j$$

$$\alpha_{i,j} C_j \geq \alpha_{i,j}(r_j + L_{i,j}) \qquad \forall j \in J, \forall i \in M$$

$$\beta_{i,S} \sum_{j \in S} L_{i,j} C_j \geq \beta_{i,S} \frac{1}{2}\left(\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2\right) \qquad \forall i \in M, \forall S \subseteq J$$

Dual LP:

$$\max \sum_{j \in J} \sum_{i \in M} \alpha_{i,j}(r_j + L_{i,j}) + \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} \frac{1}{2}\left(\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2\right)$$

$$\text{subject to,} \quad \sum_{i \in M} \alpha_{i,j} + \sum_{i \in M} \sum_{S/j \in S} L_{i,j} \beta_{i,S} \leq w_j \qquad \forall j \in J$$

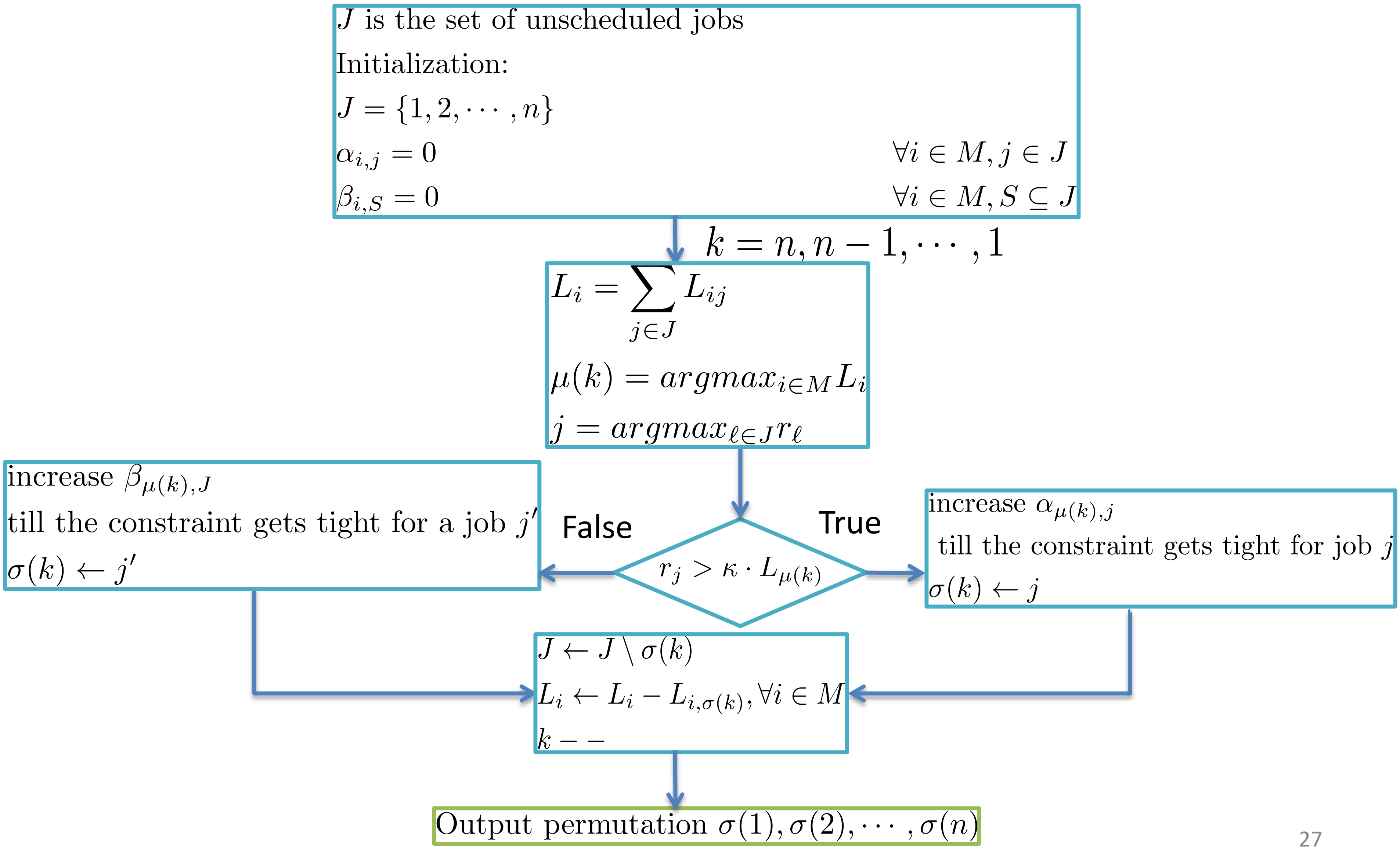
$$\alpha_{i,j} \geq 0 \qquad \forall j \in J, i \in M$$

$$\beta_{i,S} \geq 0 \qquad \forall i \in M, \forall S \subseteq J$$

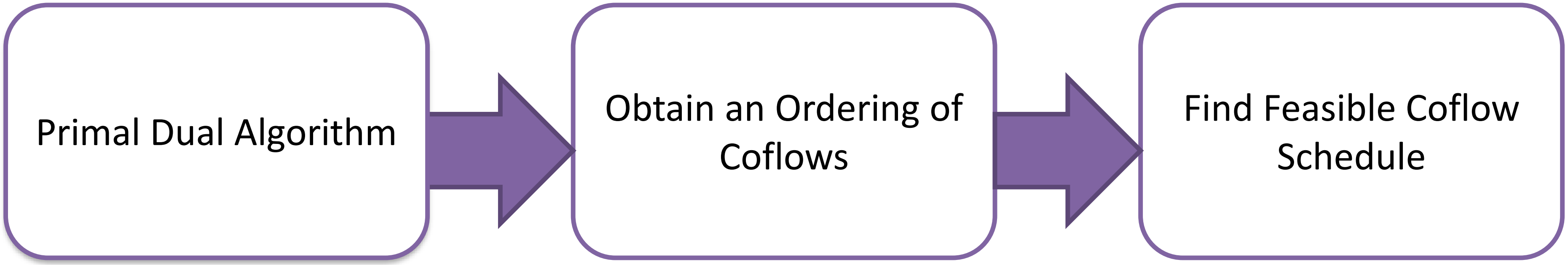
Exponential number of variables
Polynomial number of constraints

Primal Dual Algorithm

$$\max \sum_{j\in J}\sum_{i\in M}\alpha_{i,j}(r_j+L_{i,j})+\sum_{i\in M}\sum_{S\subseteq J}\beta_{i,S}\frac{1}{2}\left(\sum_{j\in S}L_{i,j}^2+(\sum_{j\in S}L_{i,j})^2\right)$$



Overview of our Approach

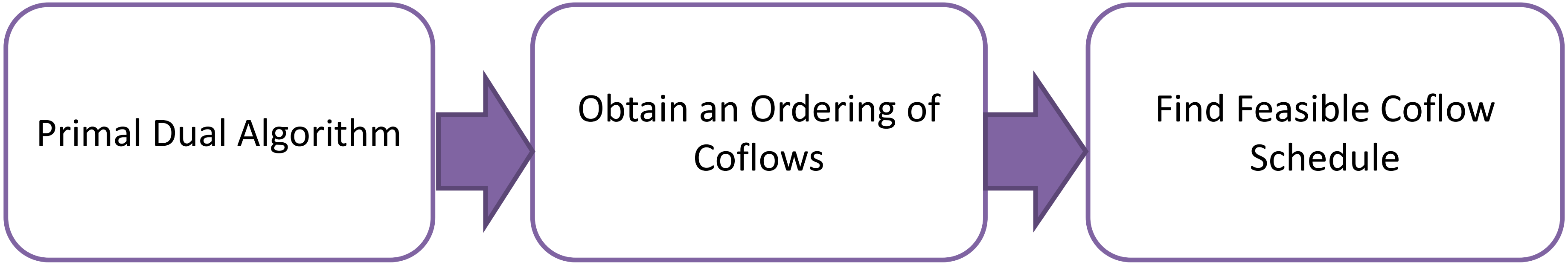


Cost of Combinatorial Algorithm

$$f_i(S) = \frac{\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2}{2}$$

Lemma 3: If there is an algorithm that generates a feasible coflow schedule such that for any coflow j, $C_j(alg) \leq a \max_{k \leq j} r_k + b\Delta(\bigcup_{k \leq j} G_k)$ for some constants a and b, then the total cost of the schedule is bounded as follows.

$$\sum_j w_j C_j(alg) \leq (a + \frac{b}{\kappa}) \sum_{j=1}^{n} \sum_{i \in M} \alpha_{i,j} r_j + 2(a\kappa + b) \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} f_i(S)$$



$G_1, G_2, \cdots, G_n$

$C_j(alg) \leq a \max_{k \leq j} r_k + b\Delta(\bigcup_{k \leq j} G_k)$

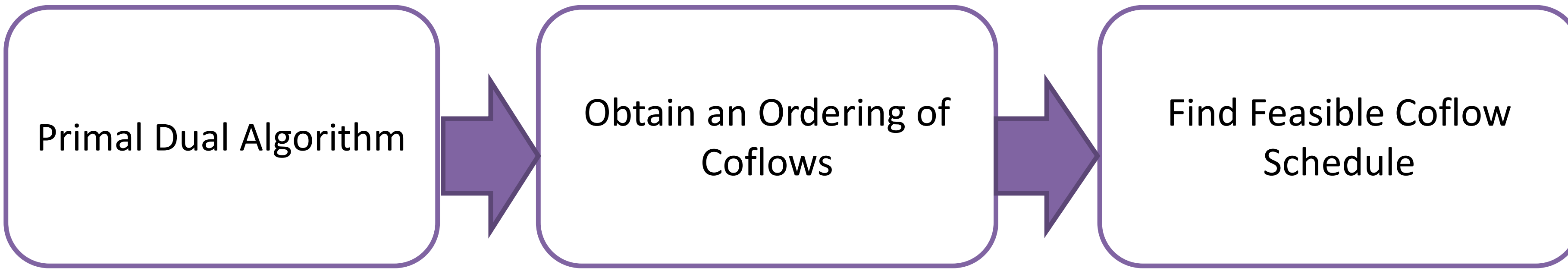
$$\sum_{j=1}^{n} w_j C_j(alg) \leq (a + \frac{b}{\kappa}) \sum_{j=1}^{n} \sum_{i \in M} \alpha_{i,j} r_j + 2(a\kappa + b) \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} f_i(S)$$

Cost of Combinatorial Algorithm

$$f_i(S) = \frac{\sum_{j\in S} L_{i,j}^2 + (\sum_{j\in S} L_{i,j})^2}{2}$$

Lemma 3: If there is an algorithm that generates a feasible coflow schedule such that for any coflow j, $C_j(alg) \le a \max_{k\le j} r_k + b\Delta(\bigcup_{k\le j} G_k)$ for some constants a and b, then the total cost of the schedule is bounded as follows.

$$\sum_j w_j C_j(alg) \le (a + \frac{b}{\kappa}) \sum_{j=1}^{n} \sum_{i\in M} \alpha_{i,j} r_j + 2(a\kappa + b) \sum_{i\in M} \sum_{S\subseteq J} \beta_{i,S} f_i(S)$$



$G_1, G_2, \cdots, G_n$

Lemma 1: $C_j(alg) \le 2\Delta(\bigcup_{k\le j} G_k)$

$$a = 0, b = 2 : \sum_{j=1}^{n} w_j C_j(alg) \le (\frac{2}{\kappa}) \sum_{j=1}^{n} \sum_{i\in M} \alpha_{i,j} r_j + 2(2) \sum_{i\in M} \sum_{S\subseteq J} \beta_{i,S} f_i(S)$$

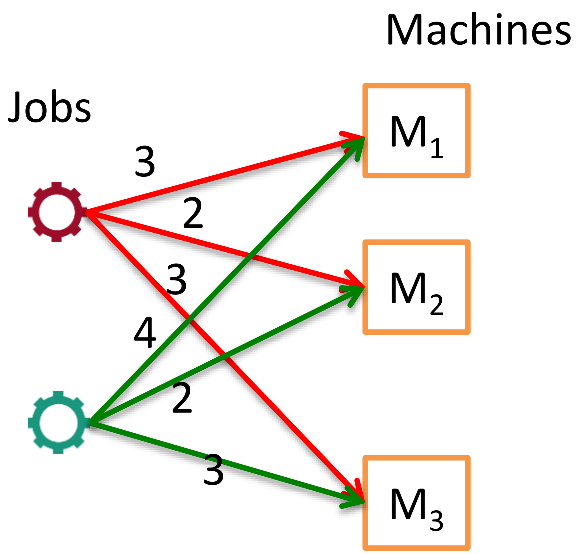
Cost of Combinatorial Algorithm

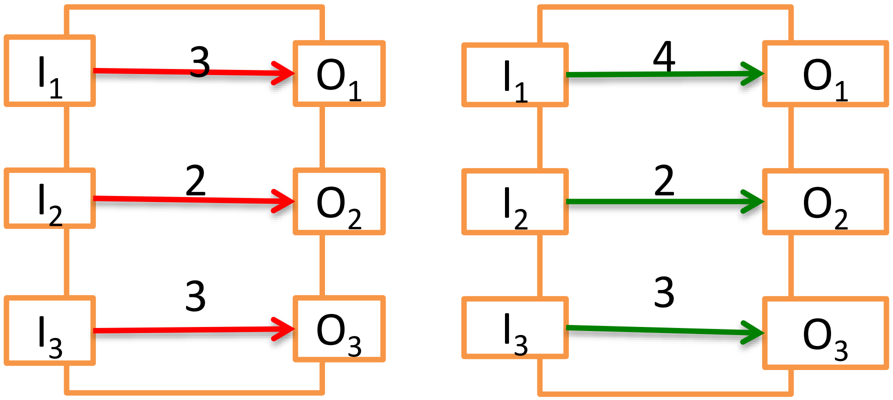
$$\frac{2}{\kappa} = 2(2) \rightarrow \kappa = \frac{1}{2}$$

$$\sum_{j=1}^{n} w_j C_j(alg) \leq 4\Big(\sum_{j=1}^{n}\sum_{i\in M} \alpha_{i,j} r_j + \sum_{i\in M}\sum_{S\subseteq J} \beta_{i,S} f_i(S)\Big) \leq 4OPT$$

5-approximation algorithm for coflow scheduling with release times!
4-approximation algorithm for coflow scheduling without release times!
3-approximation combinatorial algorithm for concurrent open shop with release times!

Concurrent Open Shop





Coflow scheduling generalizes concurrent open shop.

Open Problems

- Considering flow time instead of completion time.
- Since coflow scheduling generalizes concurrent open shop, it is NP-hard to approximate it within a factor better than (2-ε)[1]. We have an approximation factor of 4!

S. Sachdeva and R. Saket. Optimal inapproximability for scheduling problems via structural hardness for hypergraph vertex cover, 2013.

Thank You!