Random network coding over composite fields

Olav Geil
Aalborg University
Denmark

Daniel E. Lucani
Aarhus University
Denmark

5th International Castle Meeting on Coding Theory and Applications, Vihula Manor, Estonia, August 28th–31st 2017

Overview of this talk

We recall

- The algorithm by Jaggi et al. 2005 to determine a solution for a solvable multicast problem.
- Random network coding as introduced by Ho et al. 2006.
- The modification by Balli et al. 2009 of Jaggi et al. 2005 to obtain different bounds on the success probability.
- The use of composite fields as suggested by Barbero et al. 2010 and Heide et al. 2015.

Our result:

- Using Balli et al.'s approach we then derive a bound on the success probability for random network coding when composite fields are used.

Throughout the talk all networks are assumed to be cycle free.

Overview of this talk

We recall

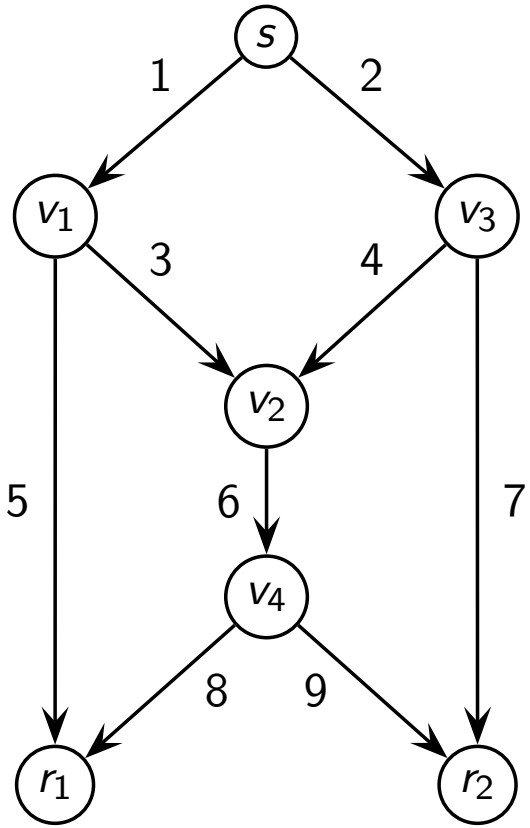
- The algorithm by Jaggi et al. 2005 to determine a solution for a solvable multicast problem.
- Random network coding as introduced by Ho et al. 2006.
- The modification by Balli et al. 2009 of Jaggi et al. 2005 to obtain different bounds on the success probability.
- The use of composite fields as suggested by Barbero et al. 2010 and Heide et al. 2015.

Our result:

- Using Balli et al.'s approach we then derive a bound on the success probability for random network coding when composite fields are used.

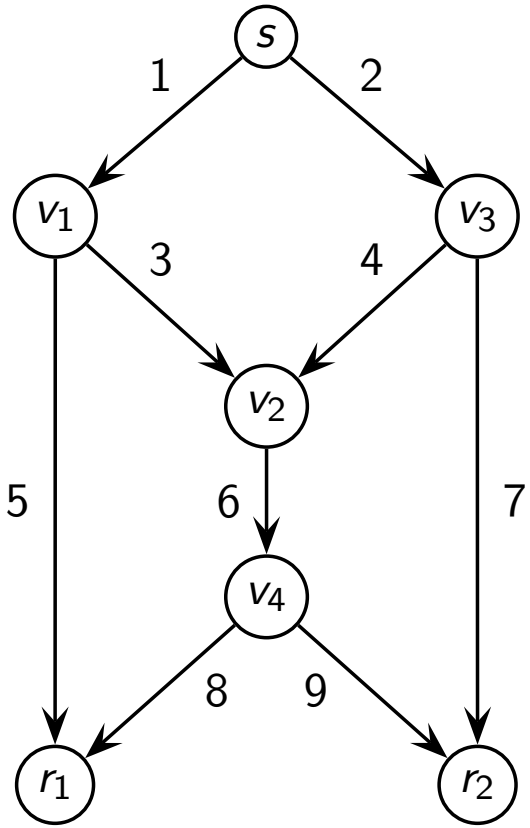
Throughout the talk all networks are assumed to be cycle free.

Simplest possible network coding problem

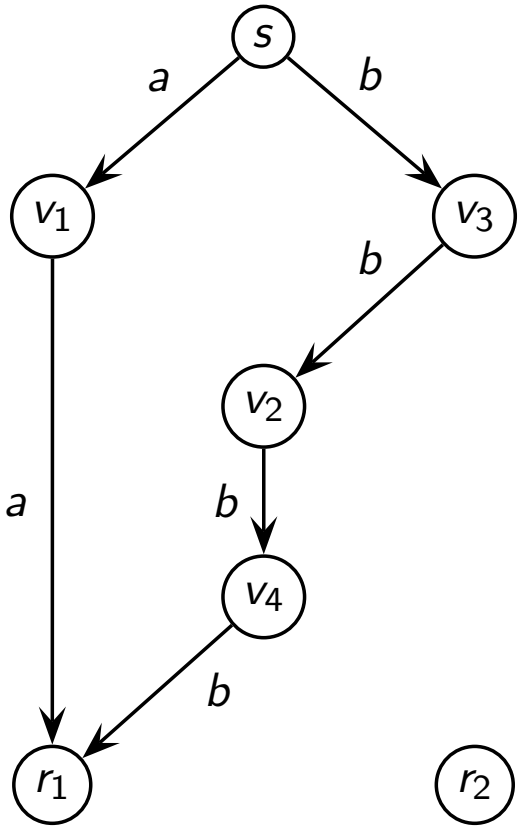


Sender s wants to send two messages $a, b \in \mathbb{F}_2$ to both receivers r_1 and r_2 simultaneously.

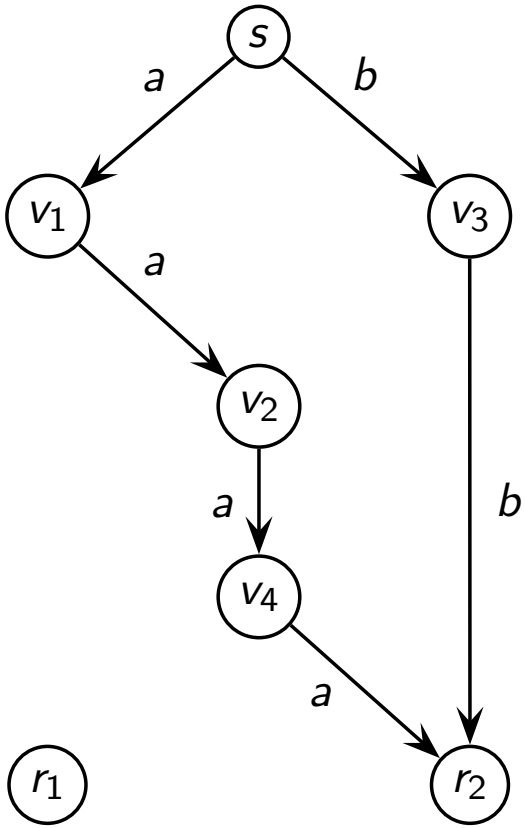
Two partial solutions



The network



Flow F_1



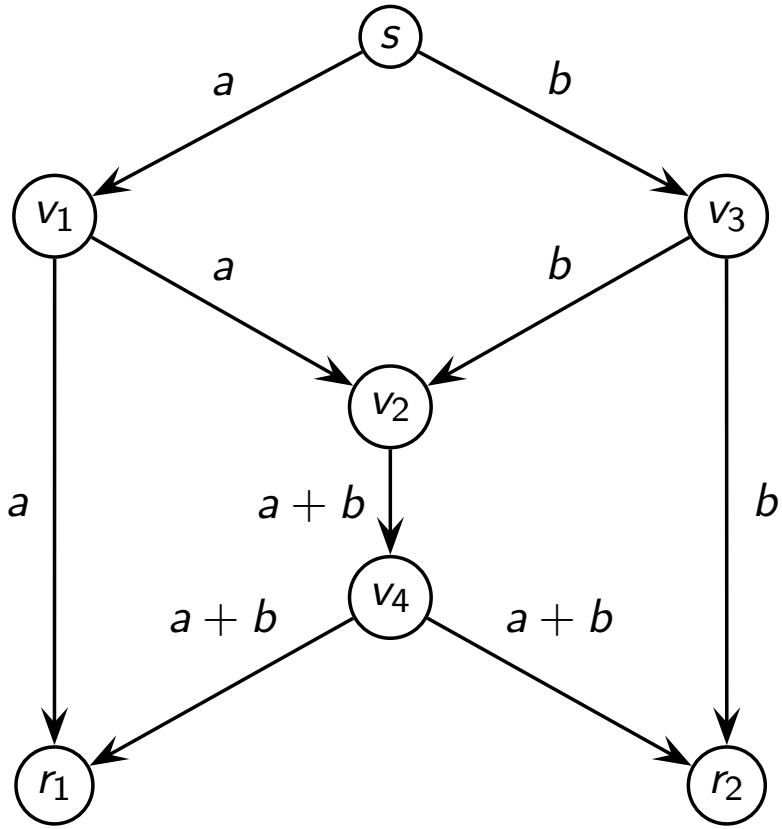
Flow F_2

The flow system is $\mathcal{F} = \{F_1, F_2\}$

$F_1 = \{(1,5),(2,4,6,8)\}, F_2 = \{(1,3,6,9),(2,7)\}$

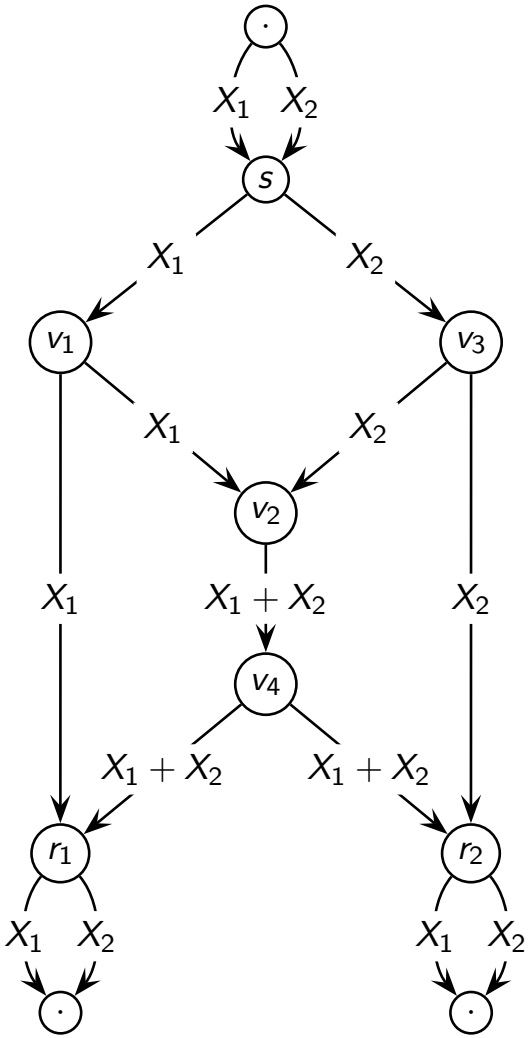
A solution

Routing is insufficient, but problem is solvable



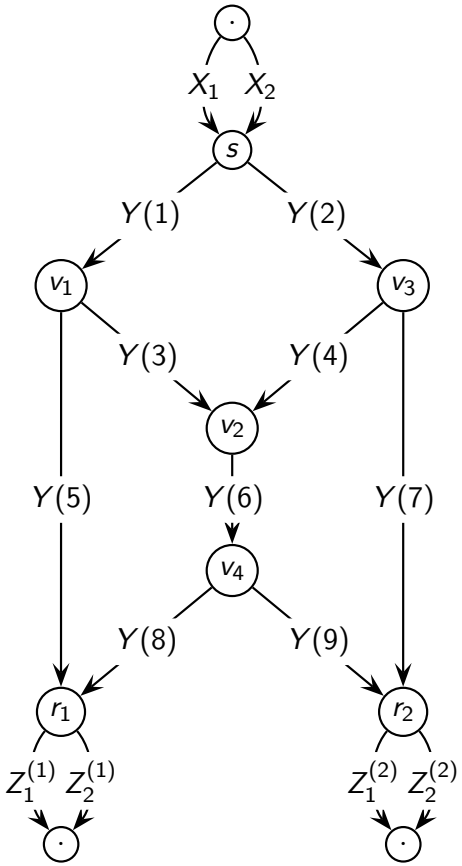
Receiver r_1 can reconstruct b as $a + (a + b)$
Receiver r_2 can reconstruct a as $(a + b) + b$

Linear network coding



Think of X_1 and X_2 as variables that take values in $\mathbb{F}_q = \mathbb{F}_2$.

Linear network coding



$$Y(1) = a_{11}X_1 + a_{21}X_2 = 1 \cdot X_1 + 0 \cdot X_2 = X_1$$
$$Y(2) = a_{12}X_1 + a_{22}X_2 = 0 \cdot X_1 + 1 \cdot X_2 = X_2$$

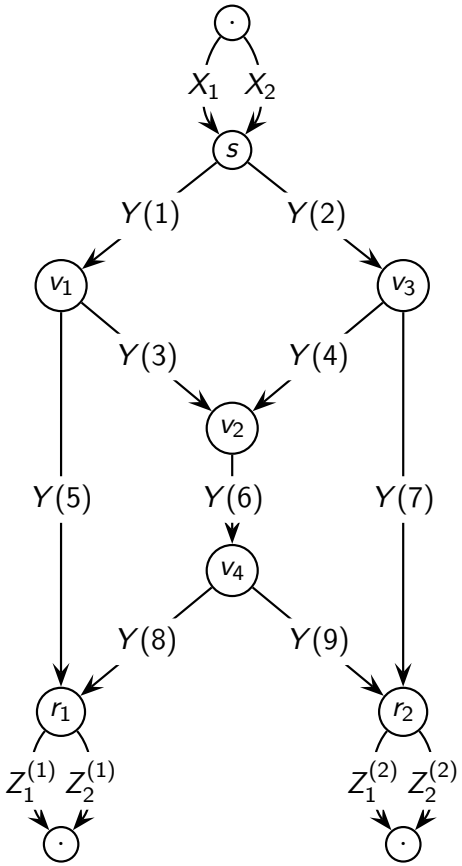
$$Y(3) = f_{13}Y(1) = 1 \cdot Y(1) = X_1$$
$$Y(4) = f_{24}Y(2) = 1 \cdot Y(2) = X_2$$
$$Y(5) = f_{15}Y(1) = 1 \cdot Y(1) = X_1$$
$$Y(6) = f_{36}Y(3) + f_{46}Y(4)$$
$$= 1 \cdot Y(3) + 1 \cdot Y(4) = X_1 + X_2$$

$$\vdots$$

$$Z_2^{(1)} = b_{52}^{(r_1)}Y(5) + b_{82}^{(r_1)}Y(8)$$
$$= 1 \cdot Y(5) + 1 \cdot Y(8) = X_1 + X_1 + X_2 = X_2$$

$$\vdots$$

Linear network coding



$$Y(1) = a_{11}X_1 + a_{21}X_2 = 1 \cdot X_1 + 0 \cdot X_2 = X_1$$
$$Y(2) = a_{12}X_1 + a_{22}X_2 = 0 \cdot X_1 + 1 \cdot X_2 = X_2$$

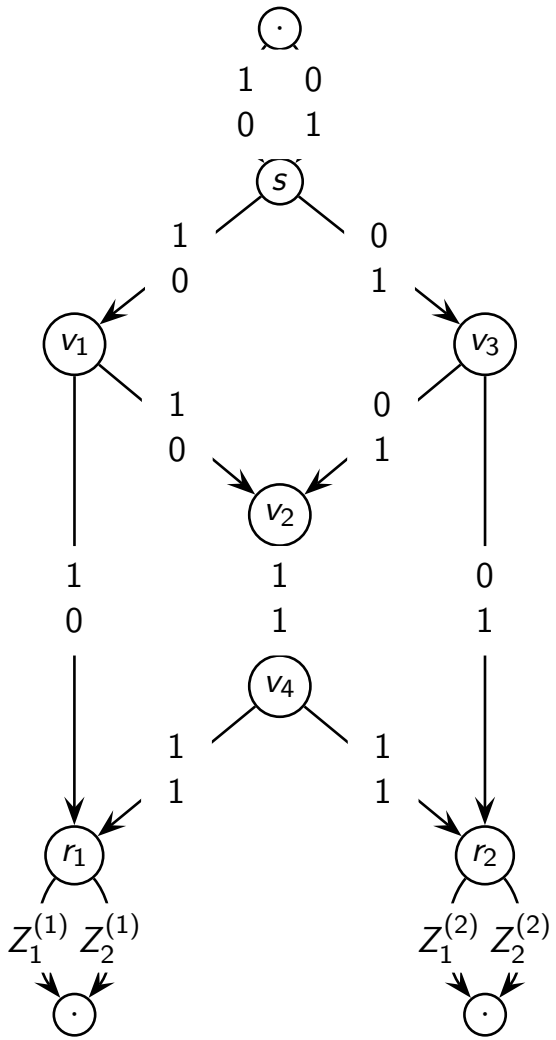
$$Y(3) = f_{13}Y(1) = 1 \cdot Y(1) = X_1$$
$$Y(4) = f_{24}Y(2) = 1 \cdot Y(2) = X_2$$
$$Y(5) = f_{15}Y(1) = 1 \cdot Y(1) = X_1$$
$$Y(6) = f_{36}Y(3) + f_{46}Y(4)$$
$$= 1 \cdot Y(3) + 1 \cdot Y(4) = X_1 + X_2$$

$$\vdots$$

$$Z_2^{(1)} = b_{52}^{(r_1)}Y(5) + b_{82}^{(r_1)}Y(8)$$
$$= 1 \cdot Y(5) + 1 \cdot Y(8) = X_1 + X_1 + X_2 = X_2$$

$$\vdots$$

Global coding vectors

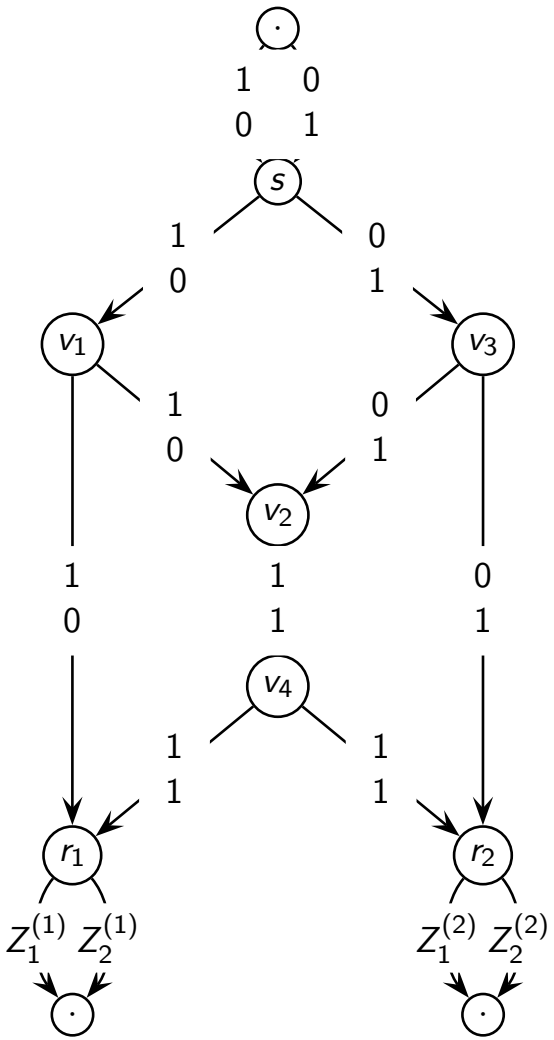


$Y(i) = c_1(i)X_1 + \cdots + c_h(i)X_h$ for some $c_1, \ldots, c_h \in \mathbb{F}_q$.

We shall call $(c_1(i), \ldots, c_h(i))$ the global coding vector for edge i.

If and only if the global coding vectors on in-comming edges of r_ℓ span $\mathbb{F}_q^h$ then the entire message can be recovered at r_ℓ.

Global coding vectors

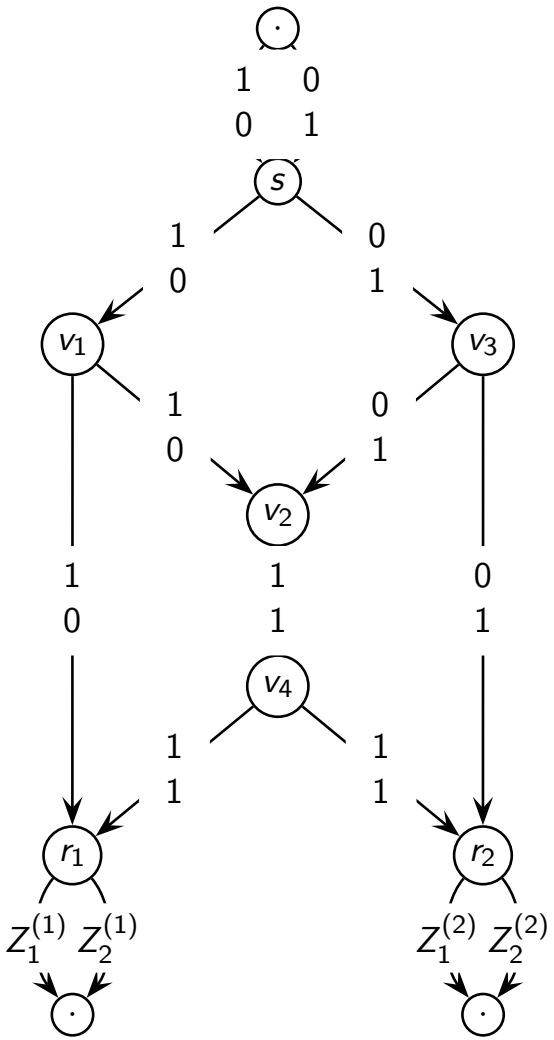


$Y(i) = c_1(i)X_1 + \cdots + c_h(i)X_h$ for some $c_1, \ldots, c_h \in \mathbb{F}_q$.

We shall call $(c_1(i), \ldots, c_h(i))$ the global coding vector for edge i.

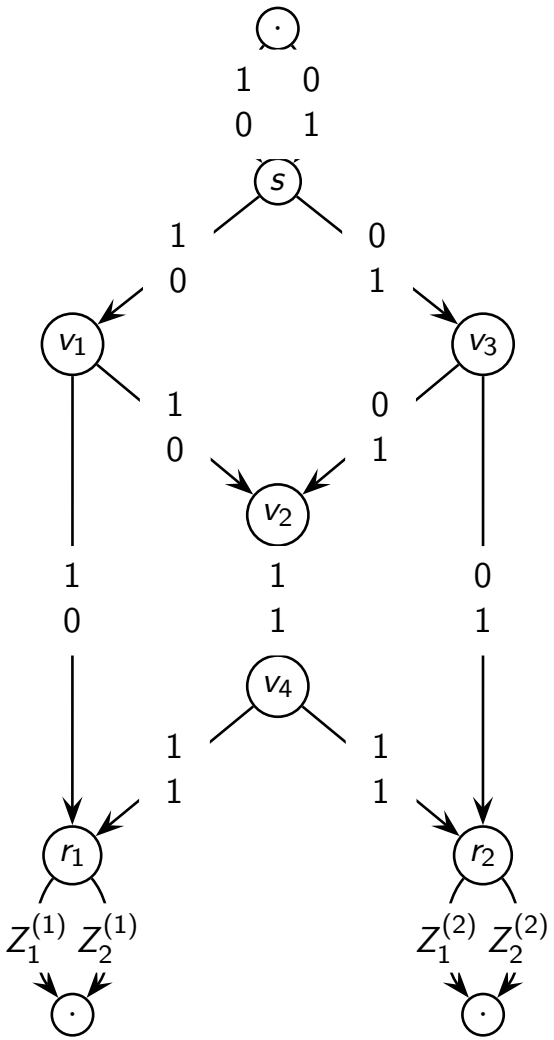
If and only if the global coding vectors on in-comming edges of r_ℓ span $\mathbb{F}_q^h$ then the entire message can be recovered at r_ℓ.

Algorithm by Jaggi et al., 2005



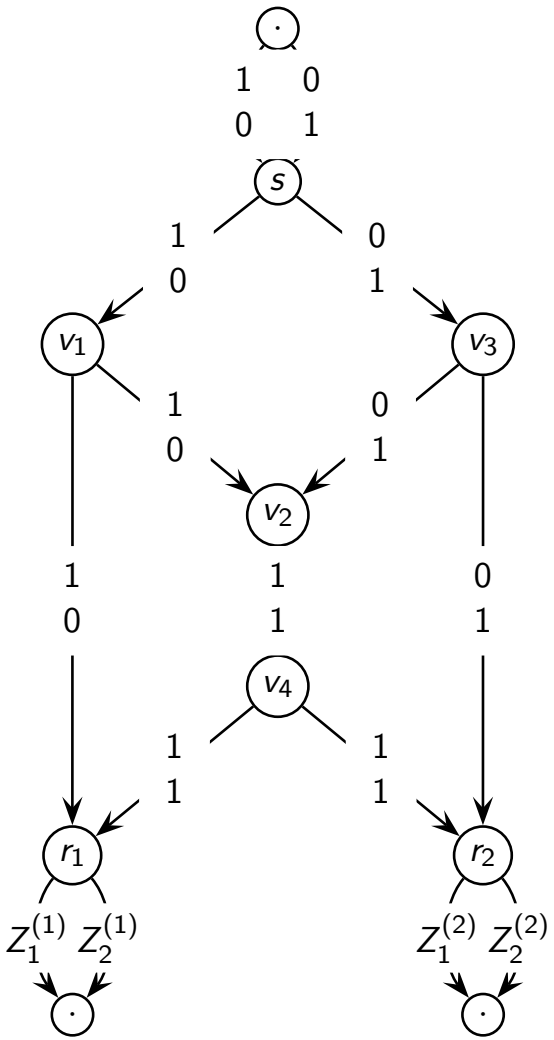
1. Start by localizing a flow system.
2. At the top of the flow system choose a basis.
3. Update local coding coefficients edge by edge moving down the flow system, in a way such that basis is kept in the cut.

Algorithm by Jaggi et al., 2005

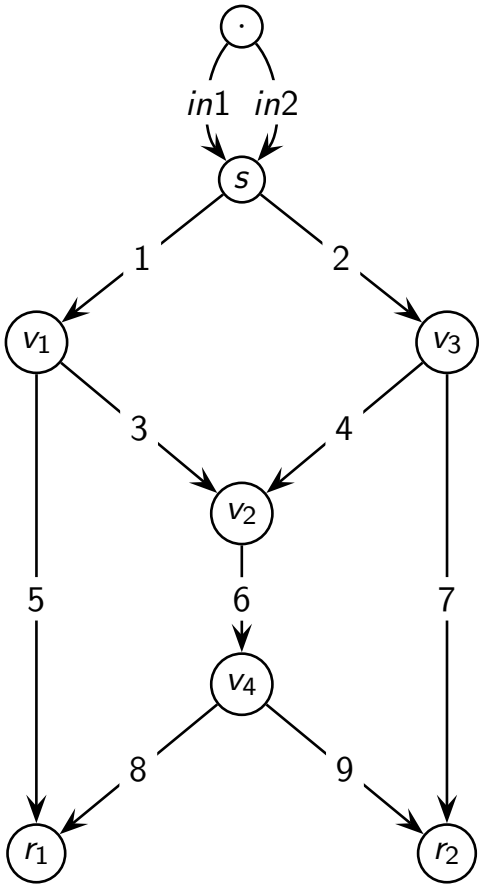


1. Start by localizing a flow system.
2. At the top of the flow system choose a basis.
3. Update local coding coefficients edge by edge moving down the flow system, in a way such that basis is kept in the cut.

Algorithm by Jaggi et al., 2005



1. Start by localizing a flow system.
2. At the top of the flow system choose a basis.
3. Update local coding coefficients edge by edge moving down the flow system, in a way such that basis is kept in the cut.

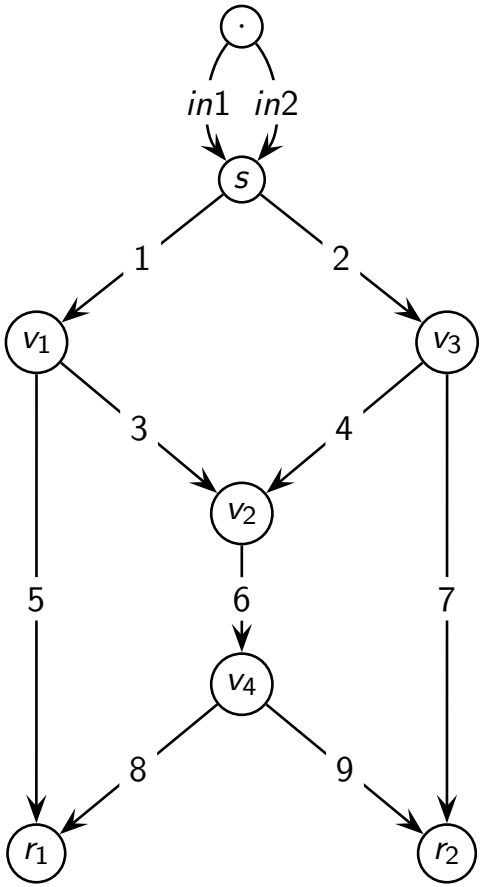


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

initialization:
$C_1 = (in1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (in1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 1:
$C_1 = (1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

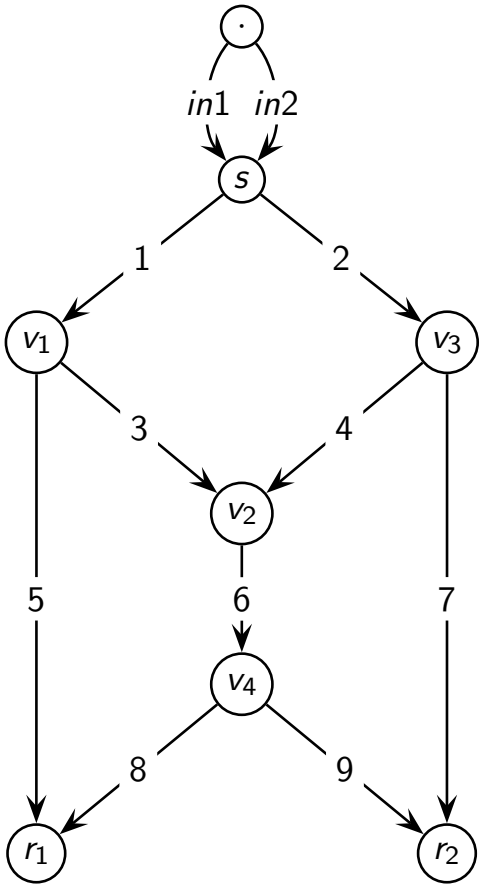


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

initialization:
$C_1 = (in1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (in1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 1:
$C_1 = (1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

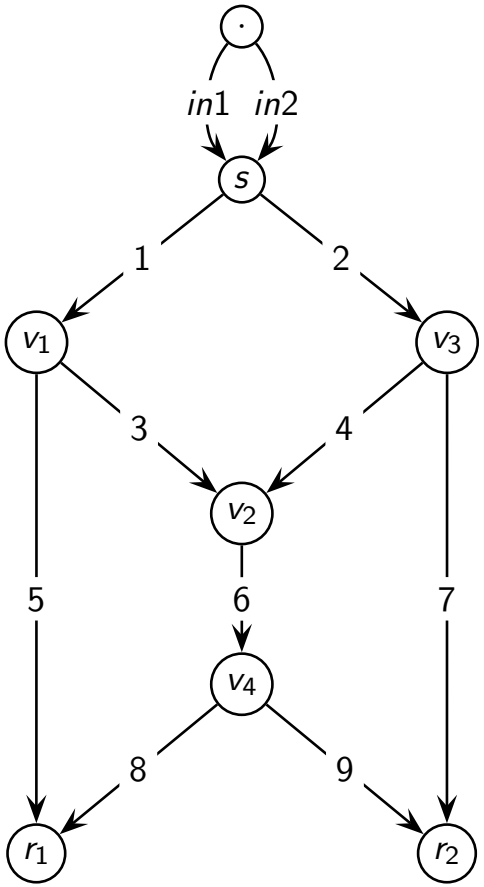


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

initialization:
$C_1 = (in1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (in1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 1:
$C_1 = (1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

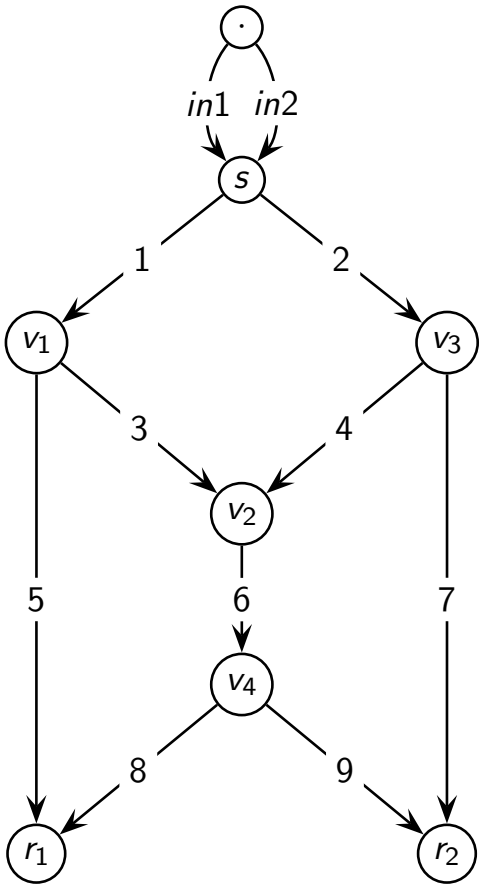


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

initialization:
$C_1 = (in1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (in1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 1:
$C_1 = (1, in2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, in2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

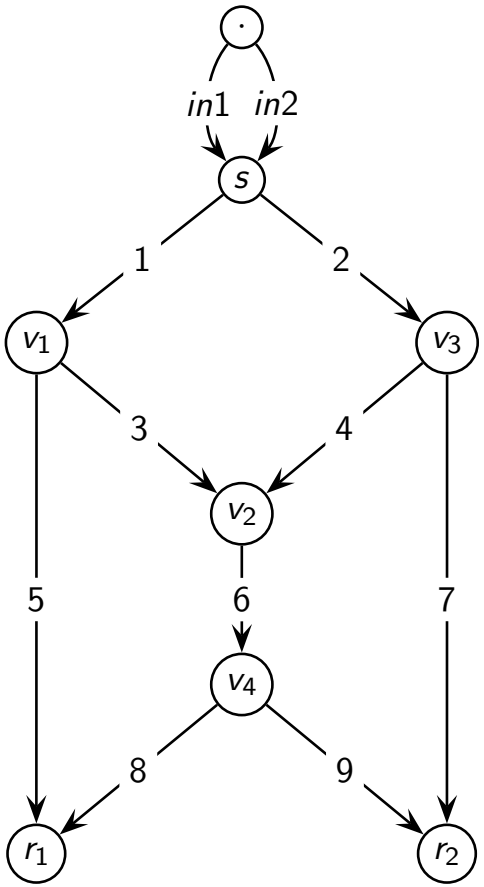


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 3:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 4:
$C_1 = (1, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

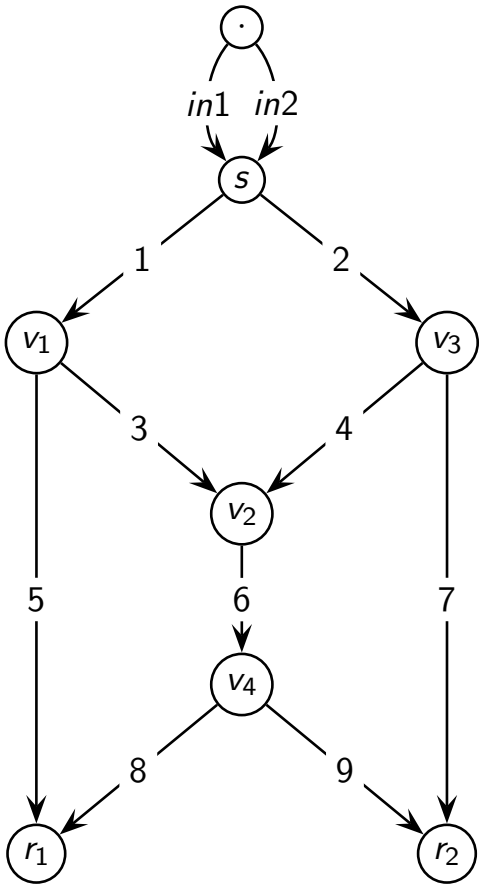


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 3:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 4:
$C_1 = (1, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

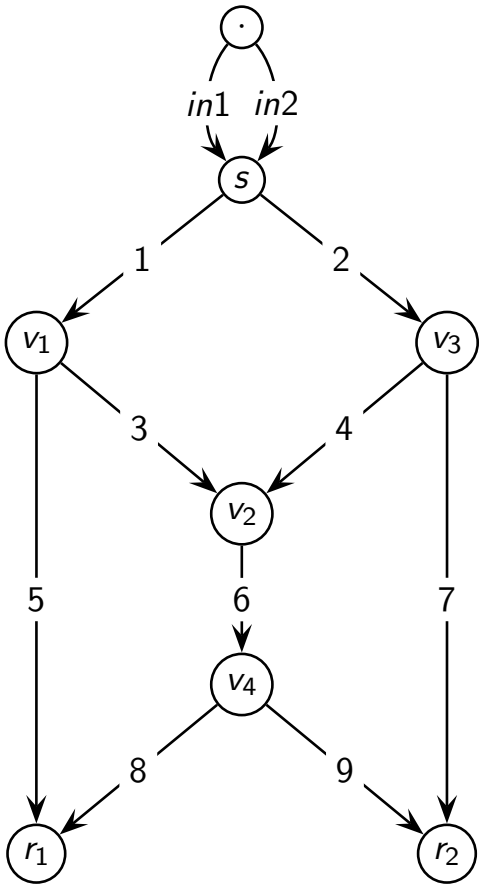


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 2:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (1, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 3:
$C_1 = (1, 2),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 4:
$C_1 = (1, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

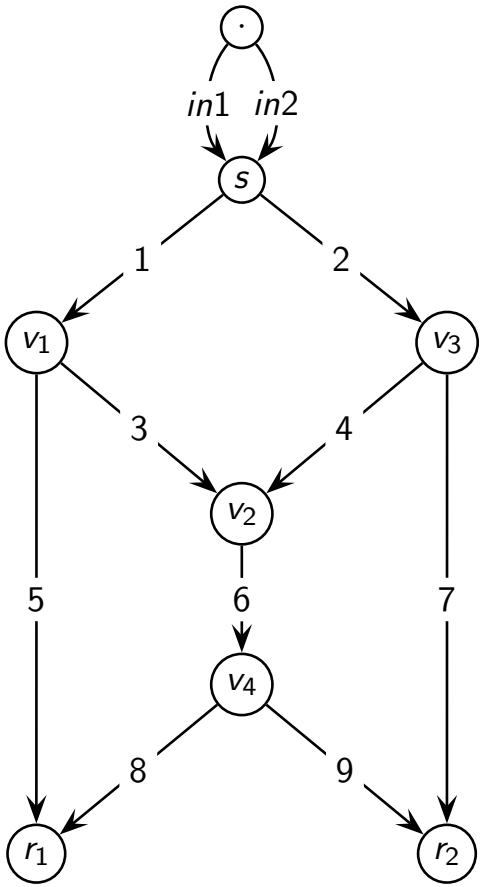


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 4:
$C_1 = (1, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 5:
$C_1 = (5, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 6:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 2),\ B_2 = \big((1, 1), (0, 1)\big)$

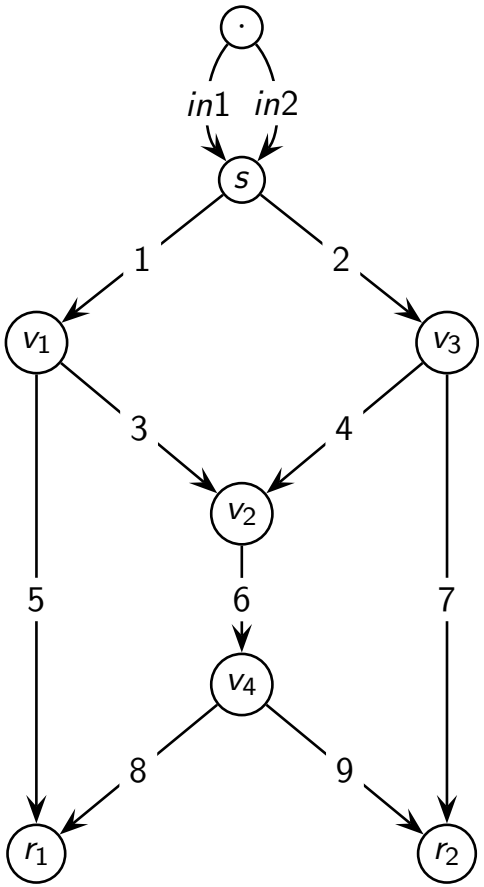


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 4:
$C_1 = (1, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 5:
$C_1 = (5, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 6:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 2),\ B_2 = \big((1, 1), (0, 1)\big)$

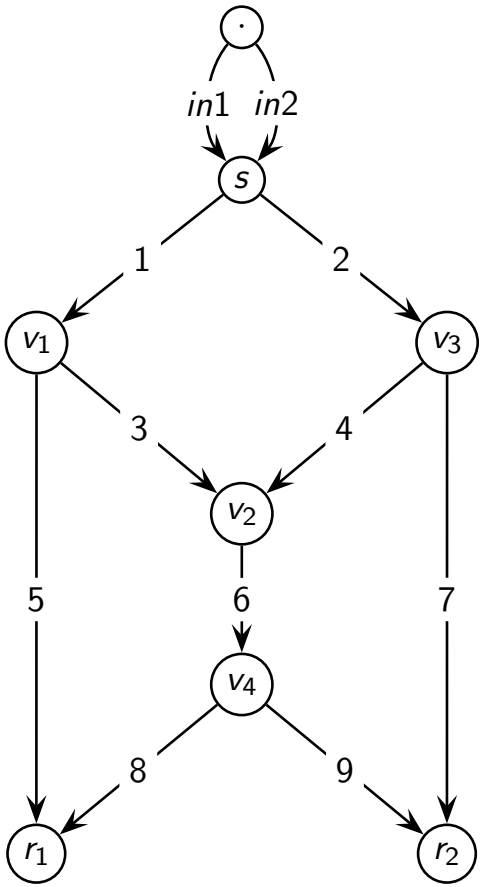


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 4:
$C_1 = (1, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 5:
$C_1 = (5, 4),\ B_1 = \big((1, 0), (0, 1)\big)$
$C_2 = (3, 2),\ B_2 = \big((1, 0), (0, 1)\big)$

update 6:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 2),\ B_2 = \big((1, 1), (0, 1)\big)$

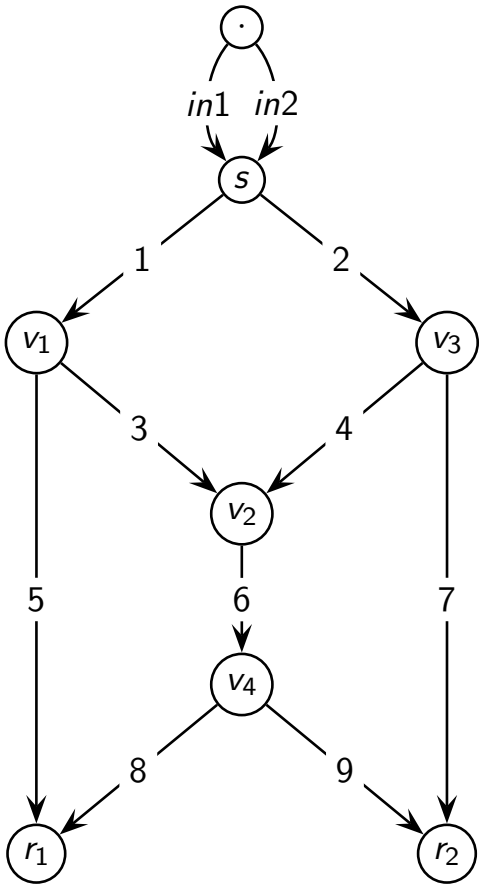


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 6:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 2),\ B_2 = \big((1, 1), (0, 1)\big)$

update 7:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

update 8:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

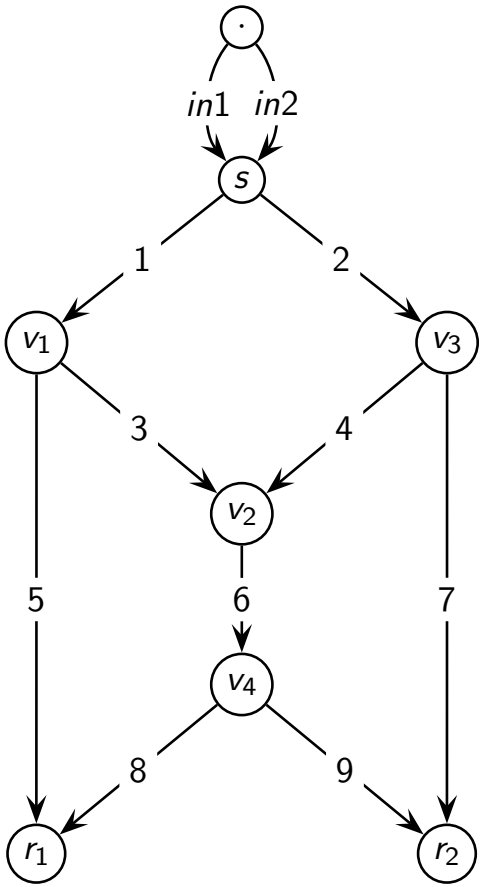


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 6:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 2),\ B_2 = \big((1, 1), (0, 1)\big)$

update 7:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

update 8:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

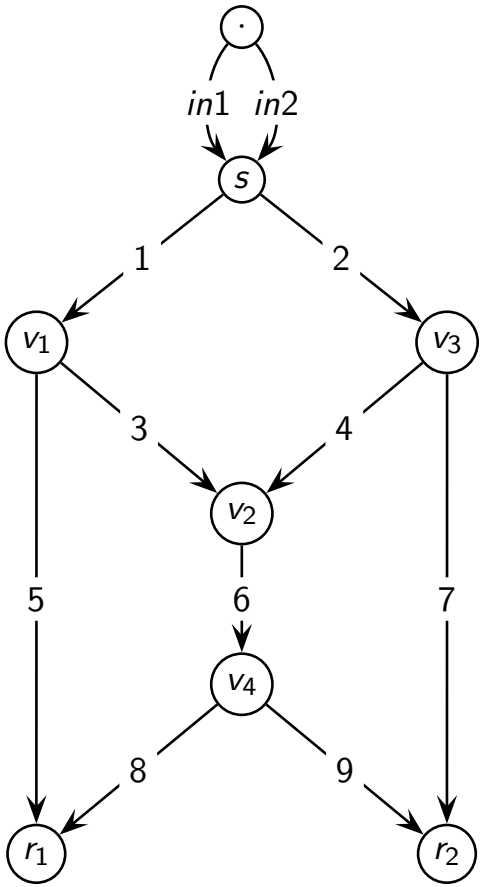


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 6:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 2),\ B_2 = \big((1, 1), (0, 1)\big)$

update 7:
$C_1 = (5, 6),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

update 8:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

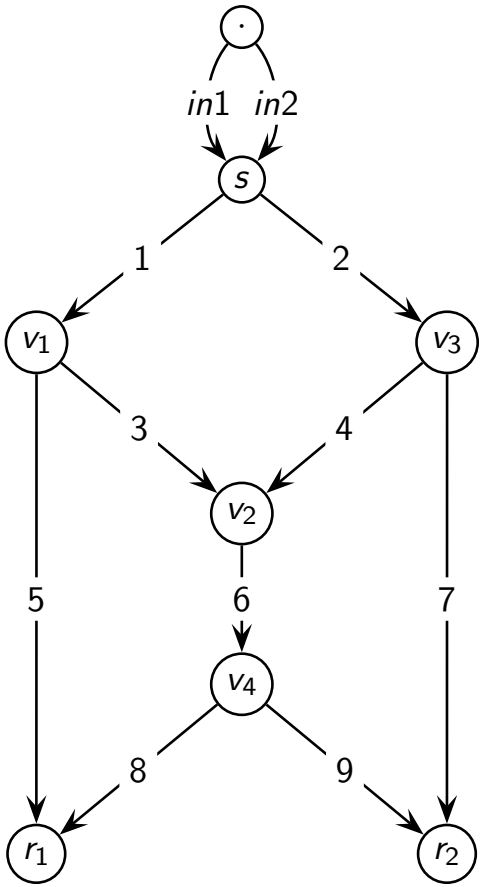


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 8:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

update 9:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (9, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

Success!!!

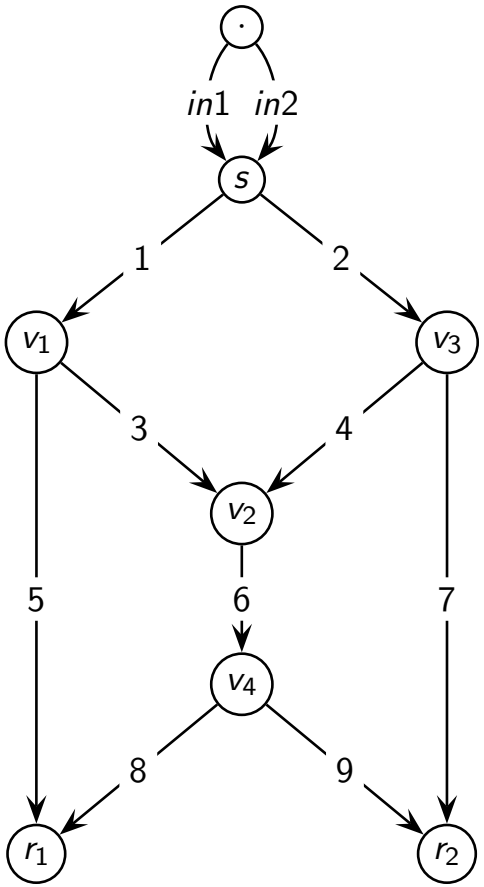


$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 8:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

update 9:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (9, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

Success!!!



$F_1 = \big((in1, 1, 5), (in2, 2, 4, 6, 8)\big)$
$F_2 = \big((in1, 1, 3, 6, 9), (in2, 2, 7)\big)$
Ordering: $(in1, in2, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

update 8:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (6, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

update 9:
$C_1 = (5, 8),\ B_1 = \big((1, 0), (1, 1)\big)$
$C_2 = (9, 7),\ B_2 = \big((1, 1), (0, 1)\big)$

Success!!!

The algorithm by Jaggi et al. - cont.

Lemma 1: Given a basis $\{\vec{b}_1, \ldots, \vec{b}_h\}$ for $\mathbb{F}_q^h$ and $\vec{c} \in \mathbb{F}_q^h$, there is exactly one choice of $a \in \mathbb{F}_q$ such that

$$\vec{c} + a\vec{b}_h \in \text{span}_{\mathbb{F}_q}\{\vec{b}_1, \ldots, \vec{b}_{h-1}\}.$$

From the Jaggi et al. algorithm we get $q \geq |R|$ is enough!!!
(the zero-solution does not work for any receiver)

Modification of Jaggi et al.'s algorithm to random network coding: Choose all local coding coefficients a_{ij} and f_{ij} at random (uniformly and independently). Use Jaggi et al.'s algorithm to check for success.

$$P_{succ} \geq \prod_{j \in \mathcal{F}} \frac{q - R_{\mathcal{F}}(j)}{q},$$

where $R_{\mathcal{F}}(j)$ is the number of receivers with a flow in $\mathcal{F}$ passing j.

The algorithm by Jaggi et al. - cont.

Lemma 1: Given a basis $\{\vec{b}_1, \ldots, \vec{b}_h\}$ for $\mathbb{F}_q^h$ and $\vec{c} \in \mathbb{F}_q^h$, there is exactly one choice of $a \in \mathbb{F}_q$ such that

$$\vec{c} + a\vec{b}_h \in \text{span}_{\mathbb{F}_q}\{\vec{b}_1, \ldots, \vec{b}_{h-1}\}.$$

From the Jaggi et al. algorithm we get $q \geq |R|$ is enough!!!
(the zero-solution does not work for any receiver)

Modification of Jaggi et al.'s algorithm to random network coding: Choose all local coding coefficients a_{ij} and f_{ij} at random (uniformly and independently). Use Jaggi et al.'s algorithm to check for success.

$$P_{succ} \geq \prod_{j \in \mathcal{F}} \frac{q - R_{\mathcal{F}}(j)}{q},$$

where $R_{\mathcal{F}}(j)$ is the number of receivers with a flow in $\mathcal{F}$ passing j.

The algorithm by Jaggi et al. - cont.

Lemma 1: Given a basis $\{\vec{b}_1, \ldots, \vec{b}_h\}$ for $\mathbb{F}_q^h$ and $\vec{c} \in \mathbb{F}_q^h$, there is exactly one choice of $a \in \mathbb{F}_q$ such that

$$\vec{c} + a\vec{b}_h \in \text{span}_{\mathbb{F}_q}\{\vec{b}_1, \ldots, \vec{b}_{h-1}\}.$$

From the Jaggi et al. algorithm we get $q \geq |R|$ is enough!!!
(the zero-solution does not work for any receiver)

Modification of Jaggi et al.'s algorithm to random network coding: Choose all local coding coefficients a_{ij} and f_{ij} at random (uniformly and independently). Use Jaggi et al.'s algorithm to check for success.

$$P_{succ} \geq \prod_{j \in \mathcal{F}} \frac{q - R_{\mathcal{F}}(j)}{q},$$

where $R_{\mathcal{F}}(j)$ is the number of receivers with a flow in $\mathcal{F}$ passing j.

The birth of random network coding

Building on the algebraic approach of Kötter-Médard 2003, Ho et al. in 2006 introduced random network coding and derived results such as

$$P_{succ} \geq \left(\frac{q - |R|}{q} \right)^{|E|}.$$

Note: Algebraic methods are not needed. Reformulating the clever lemmas in Ho et al. 2006 and using the combinatorial approach of Jaggi et al. 2005 is enough.

Balli et al. 2009 modified Jaggi et al.'s algorithm to conduct updates "vertex by vertex" rather than "edge by edge". This often results in better bounds and in particular shows:

$$P_{succ} \geq \left(\frac{q - |R|}{q - 1} \right)^{|V|}.$$

The birth of random network coding

Building on the algebraic approach of Kötter-Médard 2003, Ho et al. in 2006 introduced random network coding and derived results such as

$$P_{succ} \geq \left(\frac{q - |R|}{q} \right)^{|E|}.$$

Note: Algebraic methods are not needed. Reformulating the clever lemmas in Ho et al. 2006 and using the combinatorial approach of Jaggi et al. 2005 is enough.

Balli et al. 2009 modified Jaggi et al.'s algorithm to conduct updates "vertex by vertex" rather than "edge by edge". This often results in better bounds and in particular shows:

$$P_{succ} \geq \left(\frac{q - |R|}{q - 1} \right)^{|V|}.$$

Composite fields

Example: $\mathbb{F}_2 \subset \mathbb{F}_4 \subset \mathbb{F}_{16}$.

$$\begin{aligned} \mathbb{F}_{16} &= \{a\alpha_4 + b \mid a, b \in \mathbb{F}_4\} \\ \mathbb{F}_4 &= \{c\alpha_2 + d \mid c, d \in \mathbb{F}_2\} \end{aligned}$$

Identify an element $\beta \in \mathbb{F}_{16}$ with the following tuples (the latter being the one traveling through the network):

$$\begin{pmatrix} \beta_1^{(4)} \\ \beta_2^{(4)} \end{pmatrix} \in \mathbb{F}_4^2, \qquad \begin{pmatrix} \beta_1^{(2)} \\ \beta_2^{(2)} \\ \beta_3^{(2)} \\ \beta_4^{(2)} \end{pmatrix} \in \mathbb{F}_2^4.$$

Here,

$$\beta = \beta_1^{(4)}\alpha_4 + \beta_2^{(4)}, \quad \beta_1^{(4)} = \beta_1^{(2)}\alpha_2 + \beta_2^{(2)}, \quad \text{and} \quad \beta_2^{(4)} = \beta_3^{(2)}\alpha_2 + \beta_4^{(2)}.$$

Composite fields – cont.

$$\begin{pmatrix} \beta_1^{(4)} \\ \beta_2^{(4)} \end{pmatrix} \in \mathbb{F}_4^2, \qquad \begin{pmatrix} \beta_1^{(2)} \\ \beta_2^{(2)} \\ \beta_3^{(2)} \\ \beta_4^{(2)} \end{pmatrix} \in \mathbb{F}_2^4. \tag{1}$$

Multiplication

- with an element from $\mathbb{F}_2$ component wise on r.h.s. of expression in (1)
- with an element from $\mathbb{F}_4$ component wise on l.h.s. of expression in (1).

Random network coding over composite fields

Lemma 2: Let k, μ be non-negative integers, and let h be a positive integer such that $k + \mu < h$. Given a field $\mathbb{F}_q$ consider a basis $\{\vec{b}_1, \ldots, \vec{b}_h\}$ for $\mathbb{F}_q^h$ as a vector space over $\mathbb{F}_q$ and let $\vec{b}'_{k+1}, \ldots, \vec{b}'_{k+\mu}$ be such that

$$V = \mathrm{Span}_{\mathbb{F}_q}\{\vec{b}_1, \ldots \vec{b}_k, \vec{b}'_{k+1}, \ldots, \vec{b}'_{k+\mu}\}$$

is of dimension $k + \mu$ as a vector space over $\mathbb{F}_q$. Given $\vec{c} \in \mathbb{F}_q^h$ the number of choices of $(a_{k+1}, \ldots, a_h) \in \mathbb{F}_q^{h-k}$ such that $\vec{c} + a_{k+1}\vec{b}_{k+1} + \cdots + a_h\vec{b}_h \in V$ equals q^μ. If $\mathbb{F}_t$ is a proper subfield of $\mathbb{F}_q$ then number of choices of $(a_{k+1}, \ldots, a_h) \in \mathbb{F}_t^{h-k}$ such that $\vec{c} + a_{k+1}\vec{b}_{k+1} + \cdots + a_h\vec{b}_h \in V$ is at most t^μ.

Lemma 2: Let k, μ be non-negative integers, and let h be a positive integer such that $k + \mu < h$. Given a field $\mathbb{F}_q$ consider a basis $\{\vec{b}_1, \ldots, \vec{b}_h\}$ for $\mathbb{F}_q^h$ as a vector space over $\mathbb{F}_q$ and let $\vec{b}'_{k+1}, \ldots, \vec{b}'_{k+\mu}$ be such that

$$V = \mathrm{Span}_{\mathbb{F}_q}\{\vec{b}_1, \ldots \vec{b}_k, \vec{b}'_{k+1}, \ldots, \vec{b}'_{k+\mu}\}$$

is of dimension $k + \mu$ as a vector space over $\mathbb{F}_q$. Given $\vec{c} \in \mathbb{F}_q^h$ the number of choices of $(a_{k+1}, \ldots, a_h) \in \mathbb{F}_q^{h-k}$ such that $\vec{c} + a_{k+1}\vec{b}_{k+1} + \cdots + a_h\vec{b}_h \in V$ equals q^μ. If $\mathbb{F}_t$ is a proper subfield of $\mathbb{F}_q$ then number of choices of $(a_{k+1}, \ldots, a_h) \in \mathbb{F}_t^{h-k}$ such that $\vec{c} + a_{k+1}\vec{b}_{k+1} + \cdots + a_h\vec{b}_h \in V$ is at most t^μ.

Random network coding over composite fields – cont.

Assume h messages. For a vertex u let $h-k$ be the number of edges traveling on $\mathcal{F}_\ell$ via u. Assume u uses $\mathbb{F}_{q_u}$. Applying the lemma with $\mu = 0, \ldots, h-k-1$ we obtain that the probability for partial success is at least

$$\geq \prod_{i=0}^{h-k-1} \frac{q_u^{h-k} - q_u^i}{q_u^{h-k}} \geq 1 - \frac{1}{q_u - 1}.$$

The probability of success for all receivers is then at least

$$\geq 1 - \frac{\rho_u}{q_u - 1}$$

where ρ_u is the number of receivers with a flow in the flow system passing through u.

Random network coding over composite fields – cont.

Assume h messages. For a vertex u let $h-k$ be the number of edges traveling on $\mathcal{F}_\ell$ via u. Assume u uses $\mathbb{F}_{q_u}$. Applying the lemma with $\mu = 0, \ldots, h-k-1$ we obtain that the probability for partial success is at least

$$\geq \prod_{i=0}^{h-k-1} \frac{q_u^{h-k} - q_u^i}{q_u^{h-k}} \geq 1 - \frac{1}{q_u - 1}.$$

The probability of success for all receivers is then at least

$$\geq 1 - \frac{\rho_u}{q_u - 1}$$

where ρ_u is the number of receivers with a flow in the flow system passing through u.

Conclusion

Taking the product over all (inner) vertices in the flow system gives a bound for the overall success probability:

$$P_{succ} \geq \prod_{\substack{u \in \mathcal{F},\ \text{u has} \\ \text{outgoing edges in } \mathcal{F}}} (1 - \frac{\rho_u}{q_u - 1}) \geq \prod_{u \in \mathcal{F}} (1 - \frac{\rho_u}{q_u - 1}).$$

PS: As shown by Barbero et al., from Jaggi et al.'s algorithm one can similarly conclude that $q_u \geq \rho_u$ is enough for a solution to exists.

Conclusion

Taking the product over all (inner) vertices in the flow system gives a bound for the overall success probability:

$$P_{succ} \geq \prod_{\substack{u \in \mathcal{F},\ \text{u has} \\ \text{outgoing edges in } \mathcal{F}}} \left(1 - \frac{\rho_u}{q_u - 1}\right) \geq \prod_{u \in \mathcal{F}} \left(1 - \frac{\rho_u}{q_u - 1}\right).$$

PS: As shown by Barbero et al., from Jaggi et al.'s algorithm one can similarly conclude that $q_u \geq \rho_u$ is enough for a solution to exists.

Thanks