

Discrete Inference and Learning

Lecture 2: Maximum Flow, Minimum Cut

Yuliya Tarabalka yuliya.tarabalka@inria.fr

Outline

- Preliminaries
 - Functions and Excess Functions
 - s-t Flow
 - s-t Cut
 - Flows vs. Cuts

- Maximum Flow
- Algorithms
- Energy minimization with max flow/min cut

Context

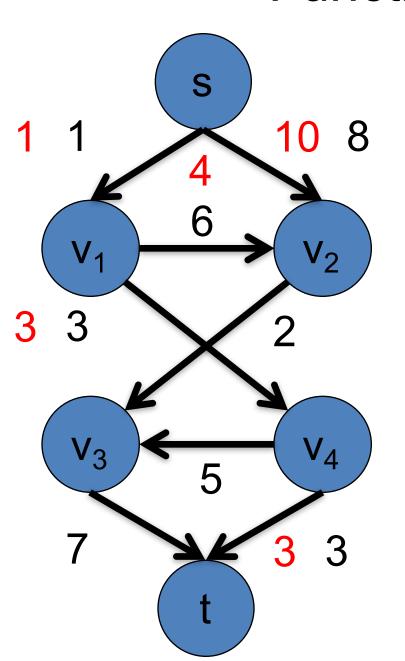
Example: network optimization problems

Nodes	Arcs	Flow
Intersections	Roads	Vehicles
Airports Switching points	Air lanes Wires, channels	Aircraft Messages
Pumping stations	Pipes	Fluids
Work centers	Materials-handling routes	Jobs

Maximum flow problem

Applications

- Maximize the flow through a company's distribution network from factories to customers
- Maximize the flow of oil through a system of pipelines
- Maximize the flow of vehicles through a transportation network



$$D = (V, A)$$

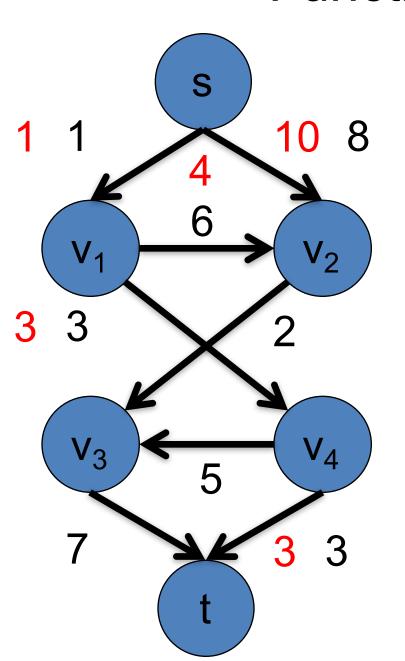
Arc capacities c(a)

Function f: A → Reals

Excess function $E_f(v)$

Incoming value

Outgoing value



$$D = (V, A)$$

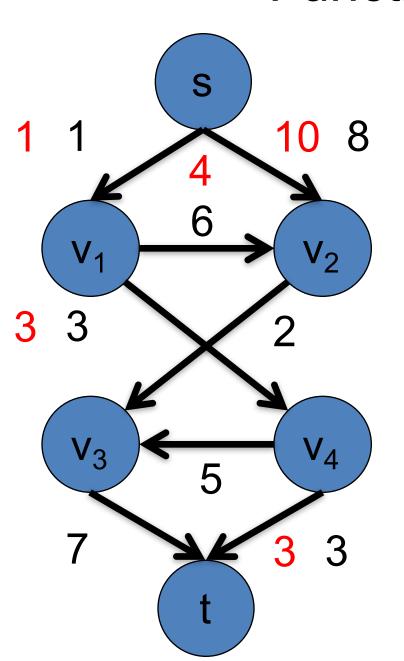
Arc capacities c(a)

Function f: A → Reals

Excess function $E_f(v)$

$$\Sigma_{a \in \text{in-arcs}(v)} f(a)$$

Outgoing value



$$D = (V, A)$$

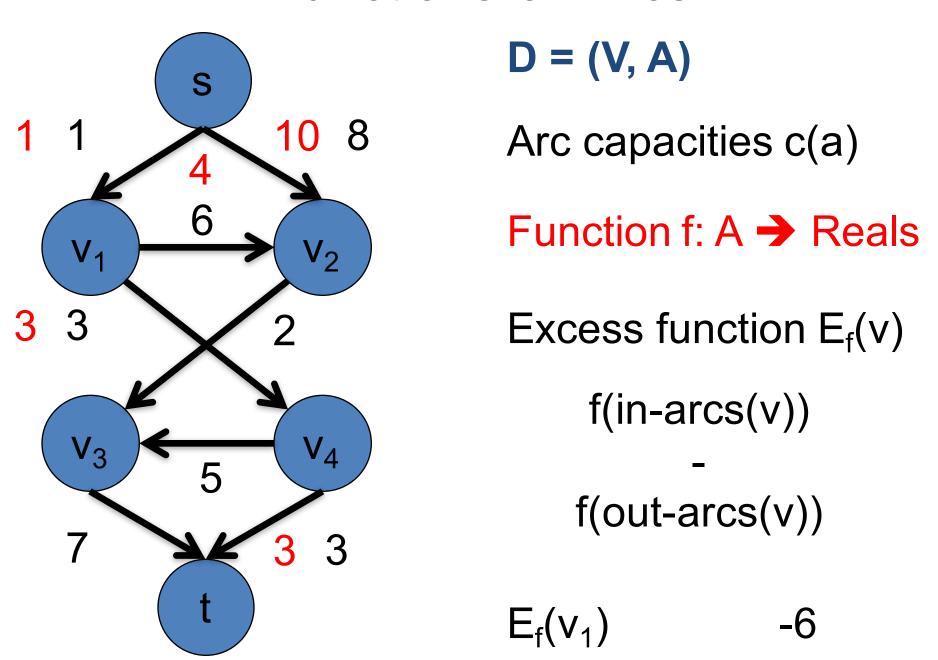
Arc capacities c(a)

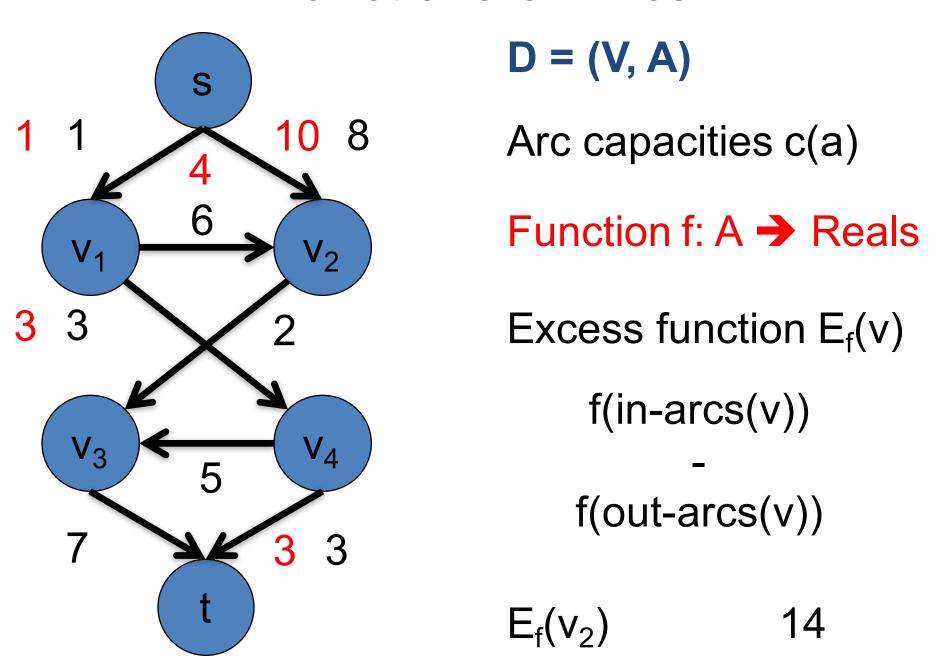
Function f: A → Reals

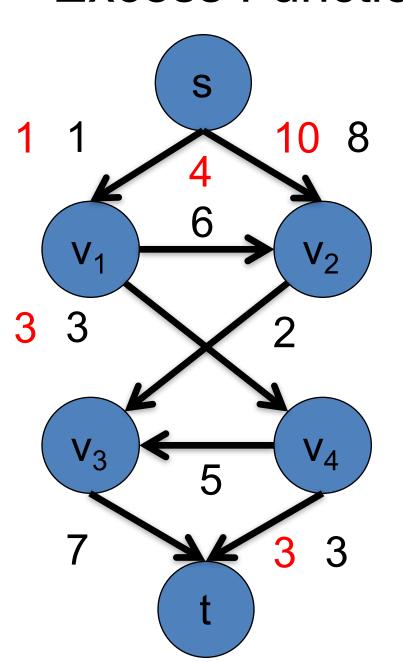
Excess function $E_f(v)$

$$\Sigma_{a \in in\text{-}arcs(v)} f(a)$$

$$\Sigma_{a \in out-arcs(v)} f(a)$$



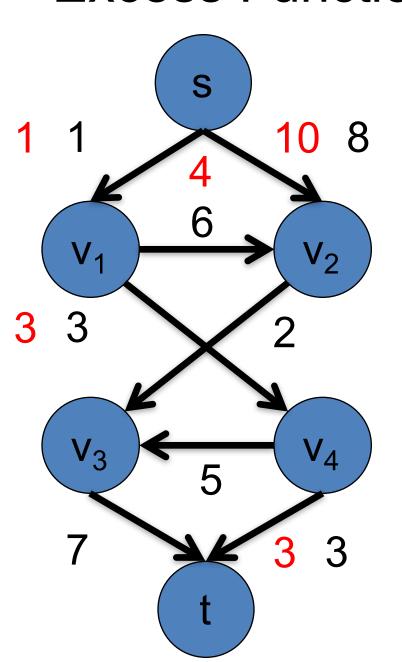




Excess function $E_f(U)$

Incoming Value

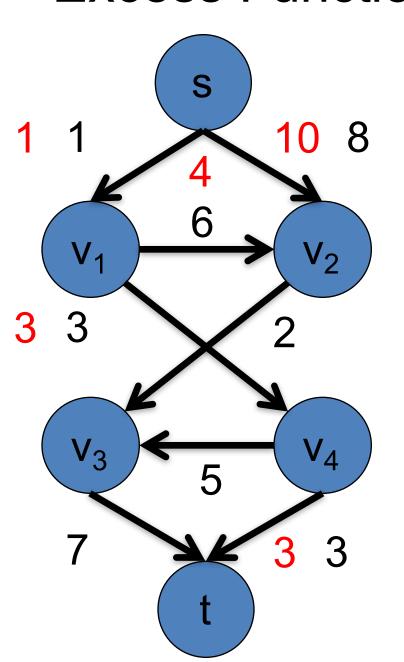
Outgoing Value



Excess function $E_f(U)$

$$\Sigma_{a \in in\text{-}arcs(U)} f(a)$$

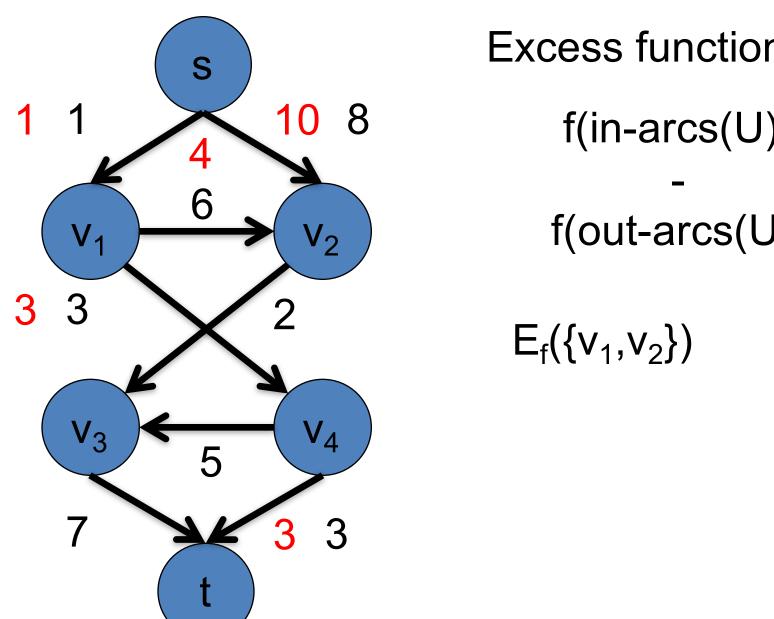
Outgoing Value



Excess function $E_f(U)$

$$\Sigma_{a \in in\text{-}arcs(U)} \, f(a)$$

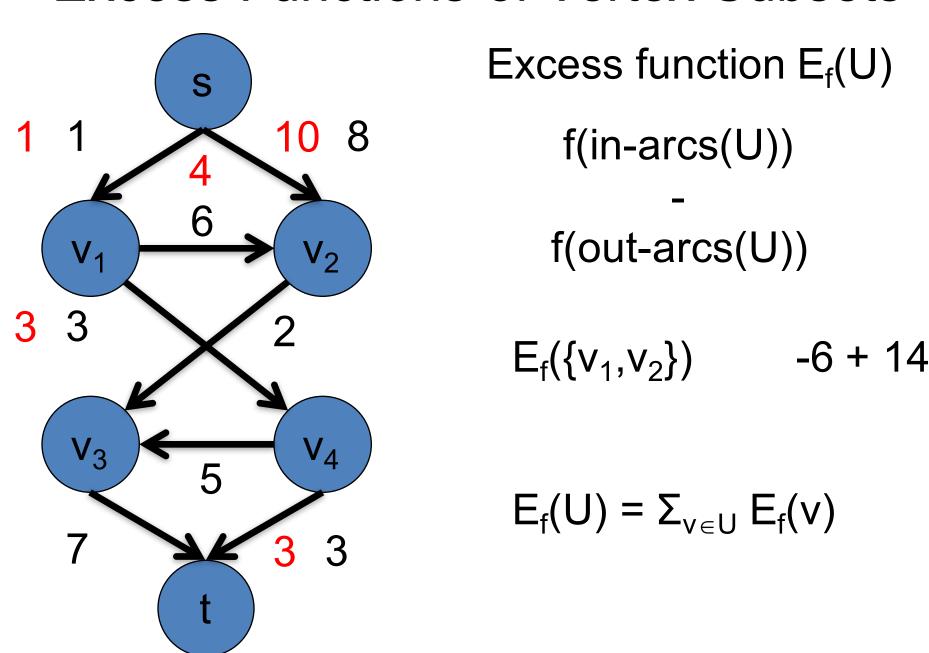
$$\Sigma_{a \in out\text{-}arcs(U)} f(a)$$



Excess function $E_f(U)$

f(in-arcs(U))

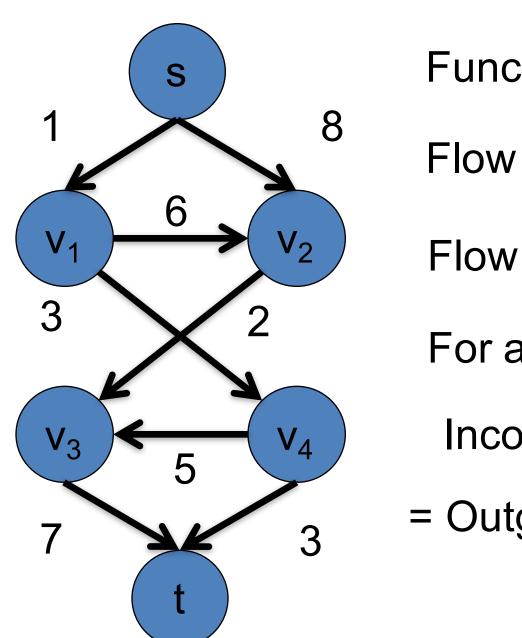
f(out-arcs(U))



Outline

- Preliminaries
 - Functions and Excess Functions
 - s-t Flow
 - s-t Cut
 - Flows vs. Cuts

- Maximum Flow
- Algorithms
- Energy minimization with max flow/min cut



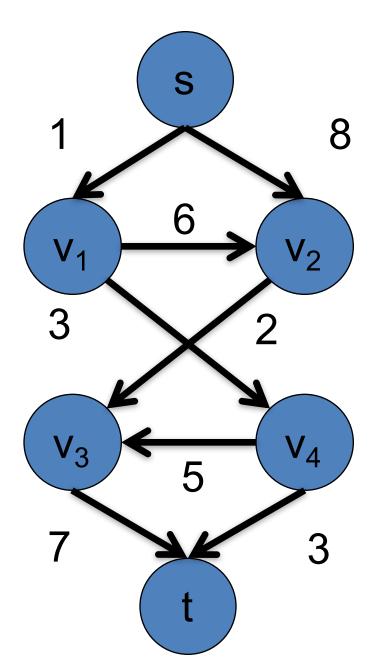
Function flow: A → R

Flow of arc ≤ arc capacity

Flow is non-negative

For all vertex except s,t

Incoming flow



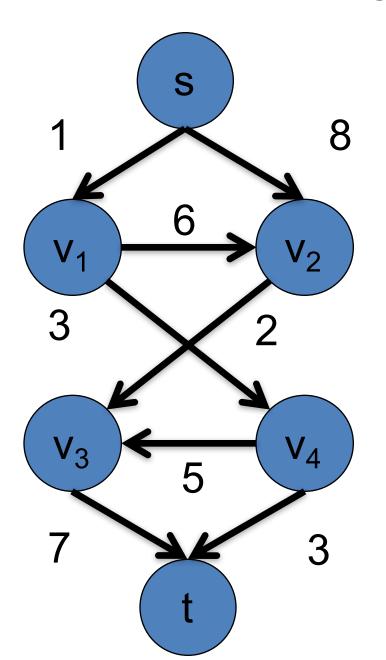
Function flow: A → R

 $flow(a) \le c(a)$

Flow is non-negative

For all vertex except s,t

Incoming flow



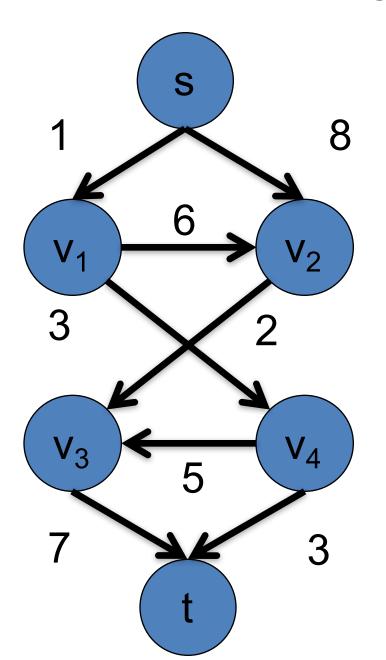
Function flow: A → R

 $flow(a) \le c(a)$

 $flow(a) \ge 0$

For all vertex except s,t

Incoming flow



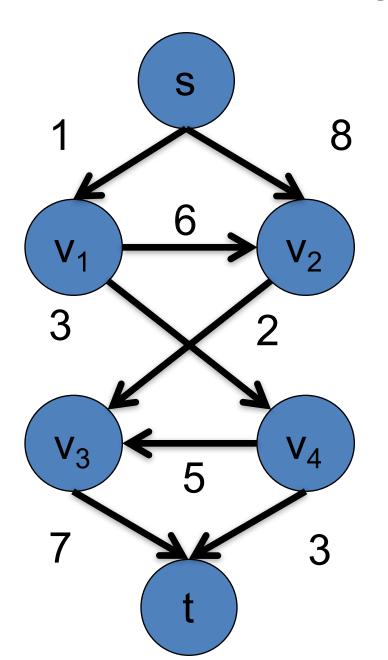
Function flow: A → R

 $flow(a) \le c(a)$

 $flow(a) \ge 0$

For all $v \in V \setminus \{s,t\}$

Incoming flow



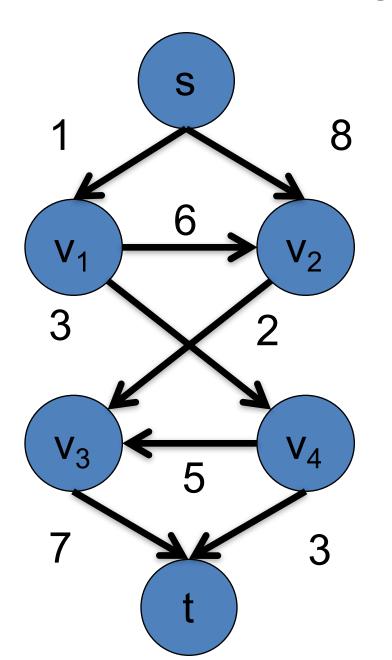
Function flow: A → R

 $flow(a) \le c(a)$

 $flow(a) \ge 0$

For all $v \in V \setminus \{s,t\}$

 $\Sigma_{(u,v)\in A}$ flow((u,v))



Function flow: A -> R

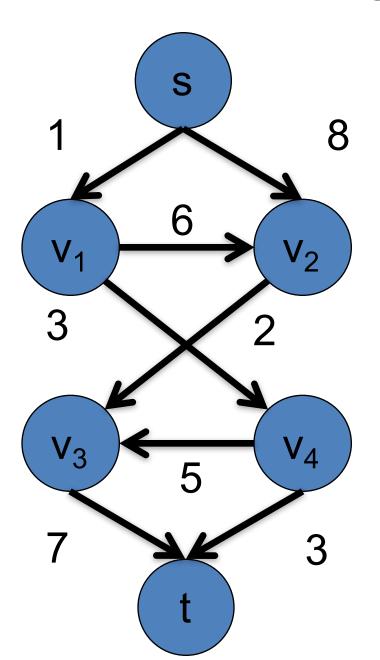
 $flow(a) \le c(a)$

 $flow(a) \ge 0$

For all $v \in V \setminus \{s,t\}$

$$\Sigma_{(u,v)\in A}$$
 flow((u,v))

= $\Sigma_{(v,u)\in A}$ flow((v,u))

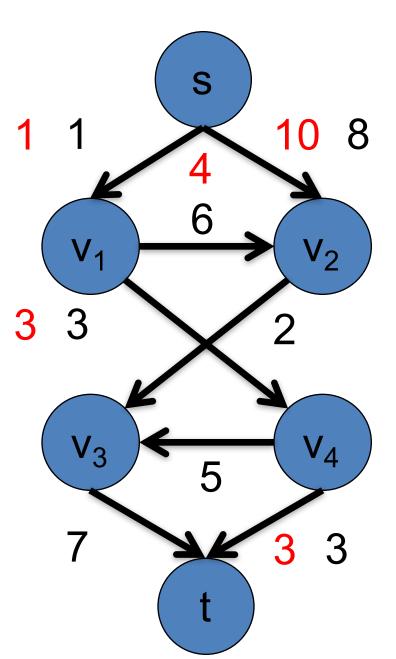


Function flow: A → R

 $flow(a) \le c(a)$

 $flow(a) \ge 0$

$$E_{flow}(v) = 0$$



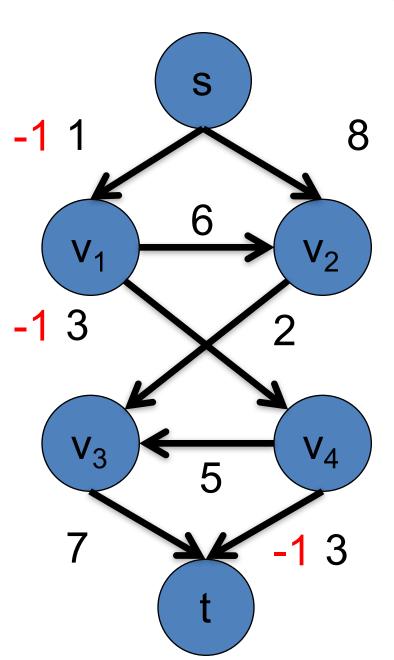
Function flow: A → R

 $flow(a) \le c(a)$

 $flow(a) \ge 0$

$$E_{flow}(v) = 0$$





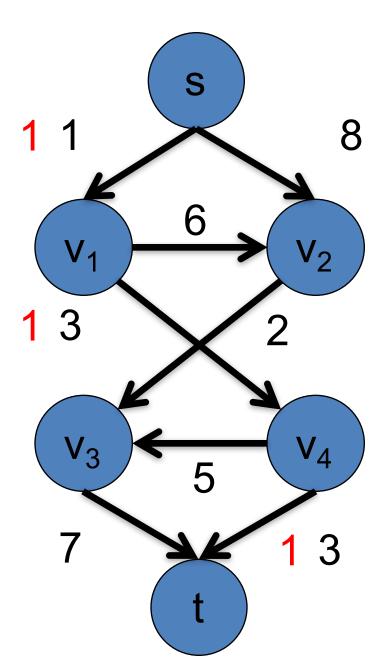
Function flow: A → R

 $flow(a) \le c(a)$

 $flow(a) \ge 0$

$$E_{flow}(v) = 0$$





Function flow: A → R

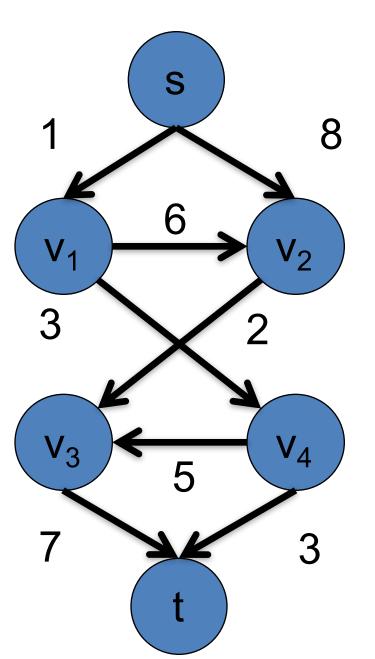
 $flow(a) \le c(a)$

 $flow(a) \ge 0$

$$E_{flow}(v) = 0$$



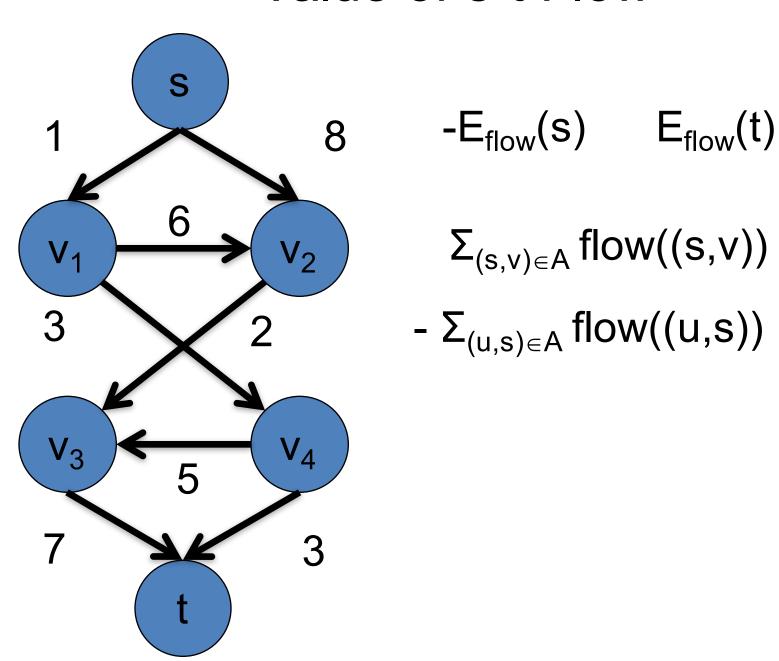
Value of s-t Flow



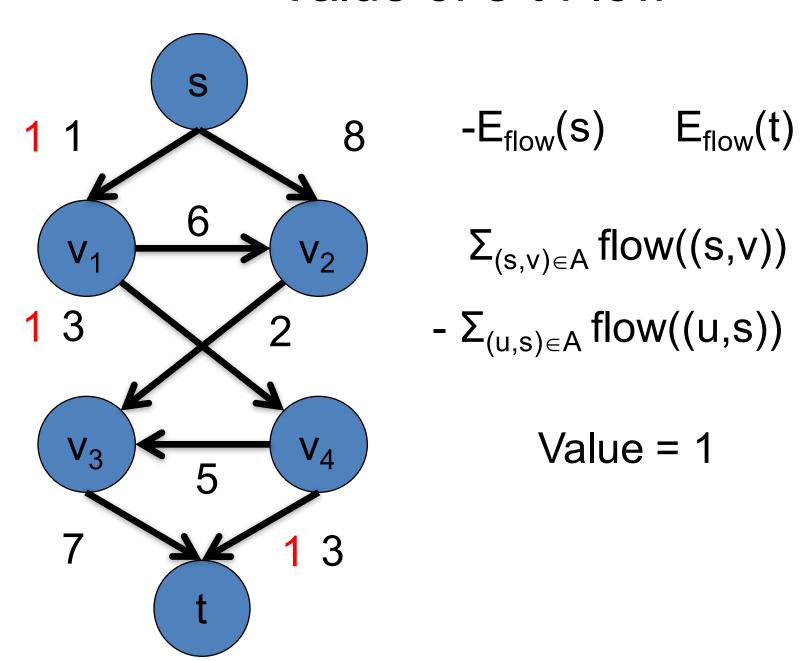
Outgoing flow of s

- Incoming flow of s

Value of s-t Flow



Value of s-t Flow



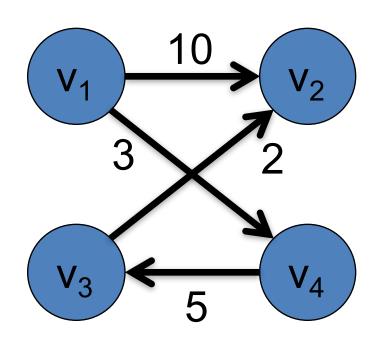
Outline

- Preliminaries
 - Functions and Excess Functions
 - s-t Flow
 - s-t Cut
 - Flows vs. Cuts

- Maximum Flow
- Algorithms
- Energy minimization with max flow/min cut

$$D = (V, A)$$

Let U be a subset of V



C is a set of arcs such that

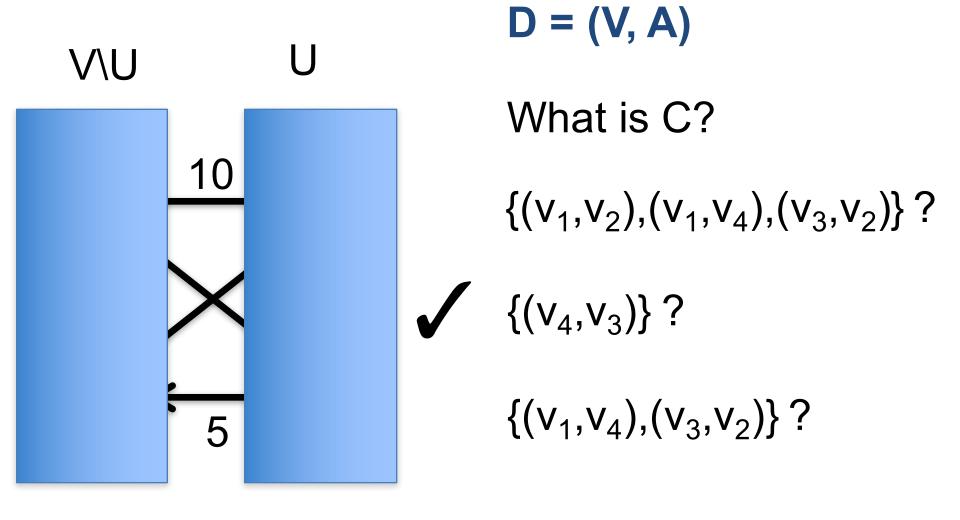
- (u,v) ∈ A
- u ∈ U
- ∨ ∈ V\U

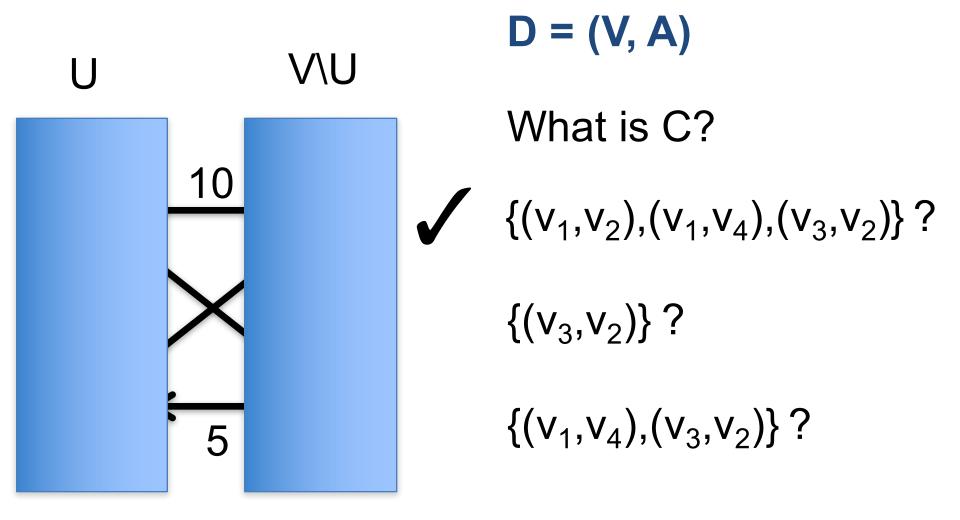
C is a cut in the digraph D

U
What is C?
$$\{(v_1,v_2),(v_1,v_4)\}?$$

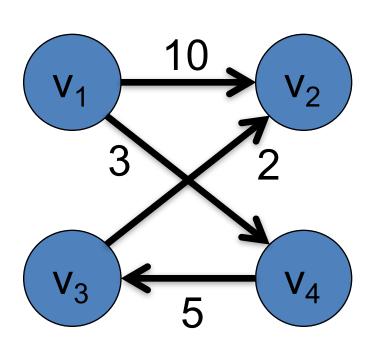
$$\{(v_1,v_4),(v_3,v_2)\}?$$

$$\{(v_1,v_4)\}?$$
V/U



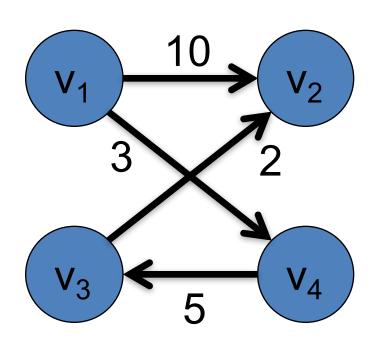


$$D = (V, A)$$



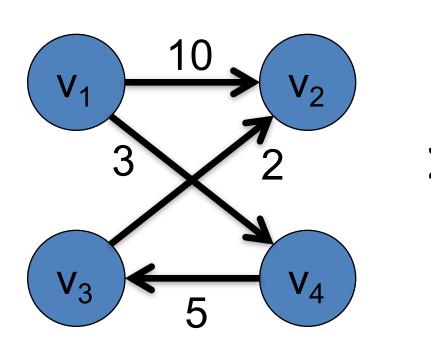
C = out-arcs(U)

Capacity of Cut



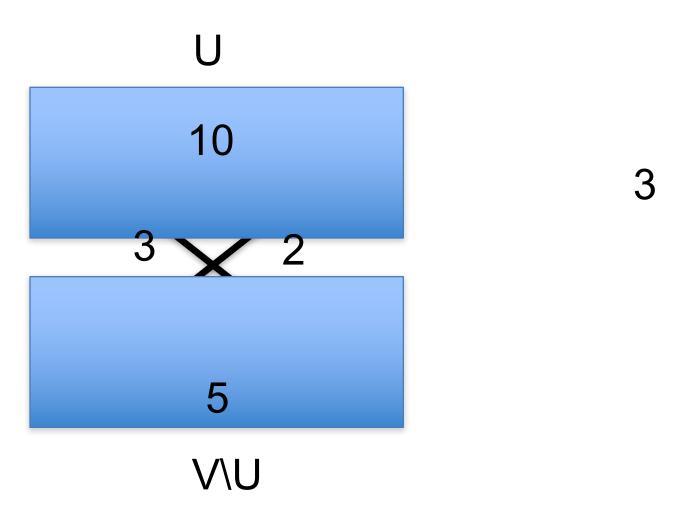
Sum of capacity of all arcs in C

Capacity of Cut

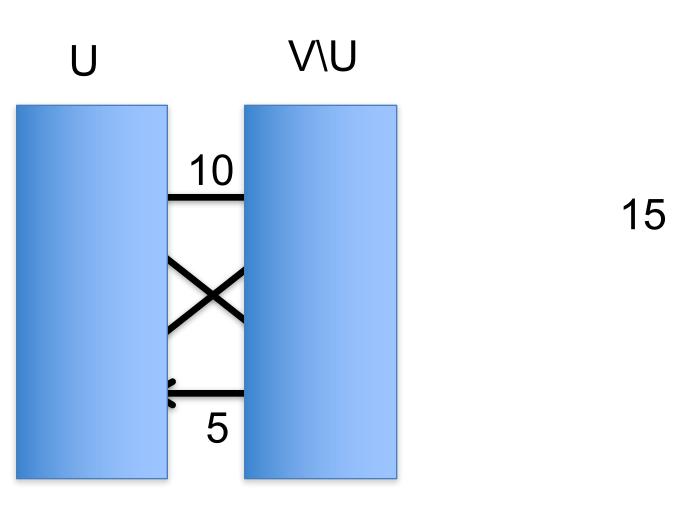


$$\Sigma_{a \in C} c(a)$$

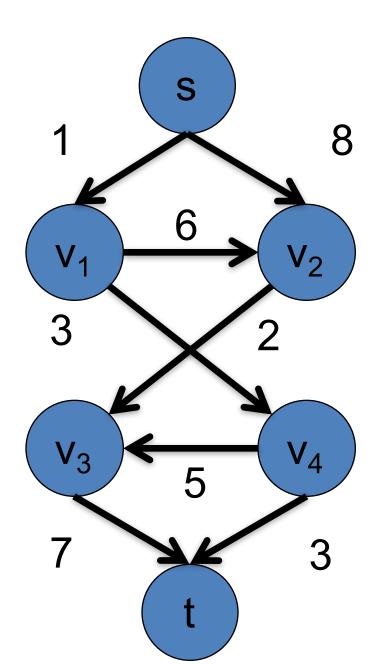
Capacity of Cut



Capacity of Cut



s-t Cut



$$D = (V, A)$$

A source vertex "s"

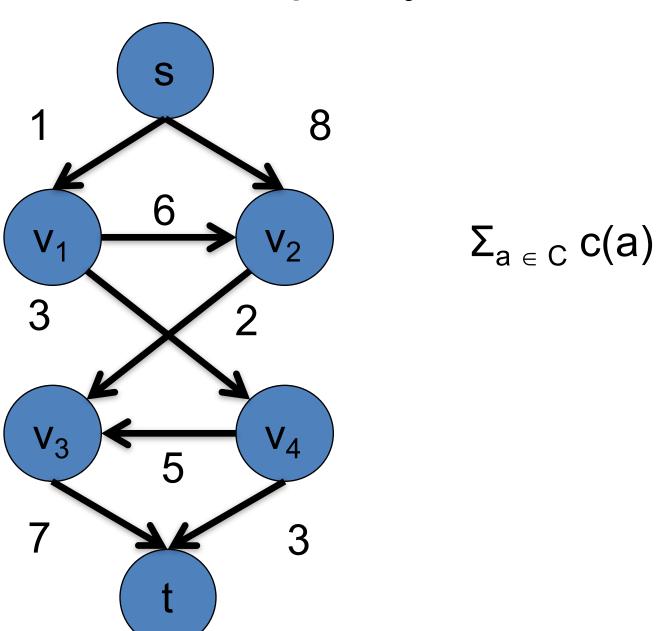
A sink vertex "t"

C is a cut such that

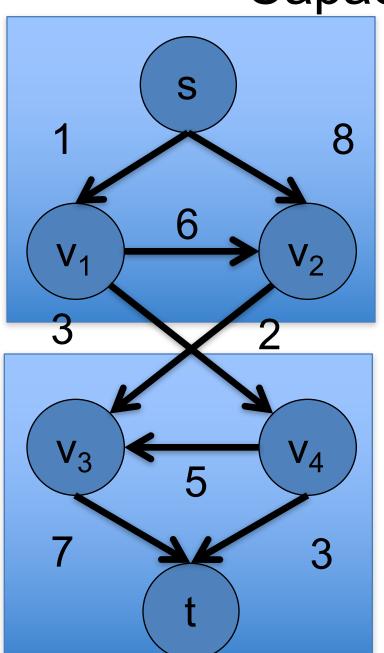
- s ∈ U
- t ∈ V\U

C is an s-t cut

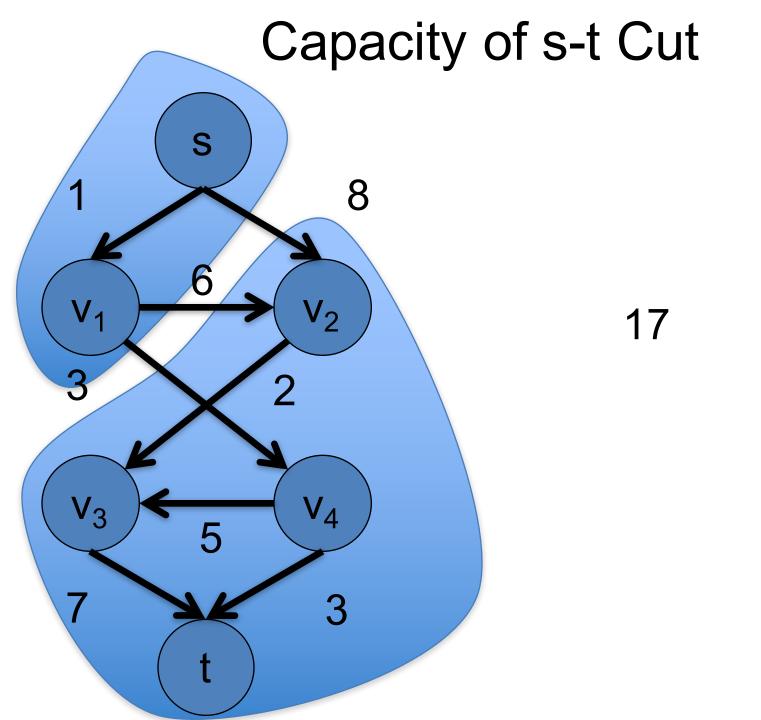
Capacity of s-t Cut



Capacity of s-t Cut



5



Outline

- Preliminaries
 - Functions and Excess Functions
 - s-t Flow
 - s-t Cut
 - Flows vs. Cuts

- Maximum Flow
- Algorithms
- Energy minimization with max flow/min cut

An s-t flow function : A → Reals

An s-t cut C such that $s \in U$, $t \in V \setminus U$

Value of flow ≤ Capacity of C

Value of flow =
$$-E_{flow}(s)$$

= $-E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$
= $-E_{flow}(U)$
= flow(out-arcs(U))
- flow(in-arcs(U))
 \leq Capacity of C
- flow(in-arcs(U))

Value of flow
$$= -E_{flow}(s)$$

 $= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$
 $= -E_{flow}(U)$
 $= flow(out-arcs(U))$
 $- flow(in-arcs(U))$
 $\leq Capacity of C$

When does equality hold?

Value of flow
$$= -E_{flow}(s)$$

 $= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$
 $= -E_{flow}(U)$
 $= flow(out-arcs(U))$
 $- flow(in-arcs(U))$
 $\leq Capacity of C$

 $flow(a) = c(a), a \in out-arcs(U)$ $flow(a) = 0, a \in in-arcs(U)$

Value of flow
$$= -E_{flow}(s)$$

 $= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$
 $= -E_{flow}(U)$
 $= flow(out-arcs(U))$
 $- flow(in-arcs(U))$
 $= Capacity of C$

Outline

Preliminaries

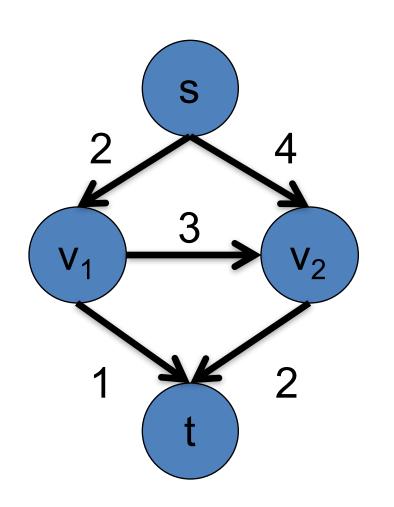
Maximum Flow

- Residual Graph
- Max-Flow Min-Cut Theorem

Algorithms

Energy minimization with max flow/min cut

Maximum Flow Problem

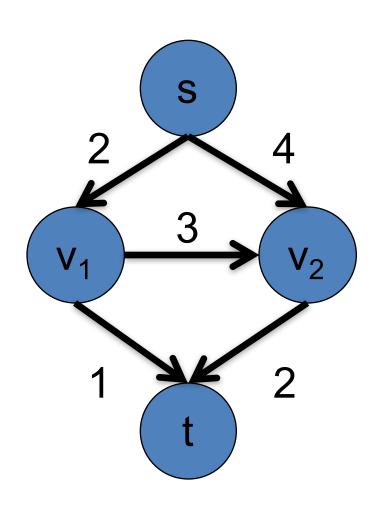


Find the flow with the maximum value !!

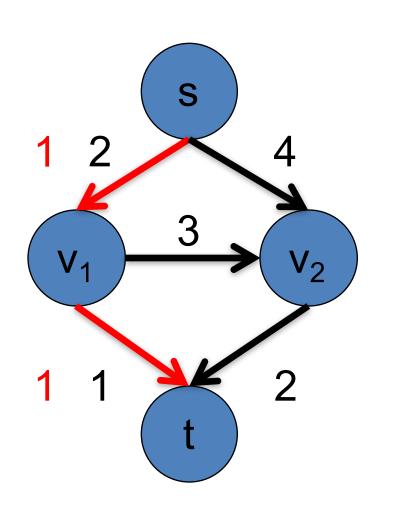
$$\Sigma_{(s,v)\in A}$$
 flow((s,v))

-
$$\Sigma_{(u,s)\in A}$$
 flow((u,s))

First suggestion to solve this problem !!

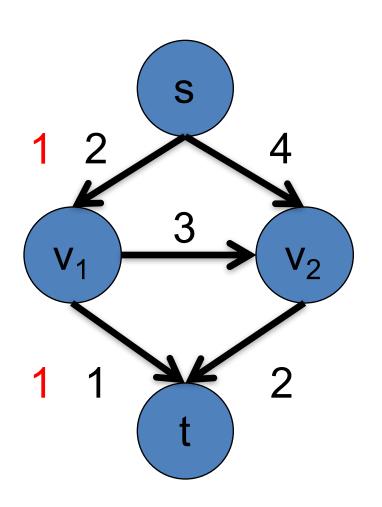


Find an s-t path where flow(a) < c(a) for all arcs

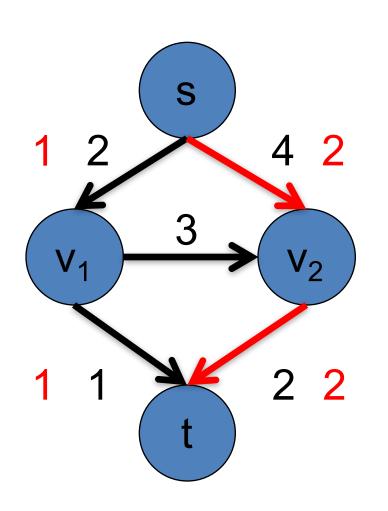


Find an s-t path where flow(a) < c(a) for all arcs

Pass maximum allowable flow through the arcs

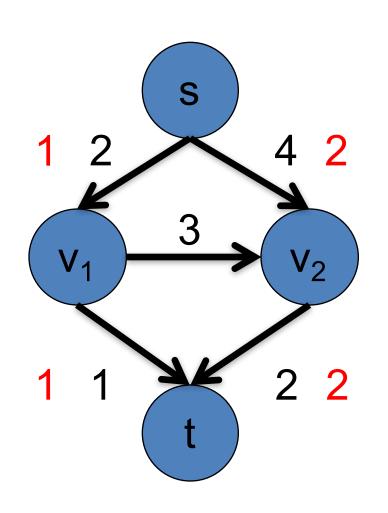


Find an s-t path where flow(a) < c(a) for all arcs



Find an s-t path where flow(a) < c(a) for all arcs

Pass maximum allowable flow through the arcs

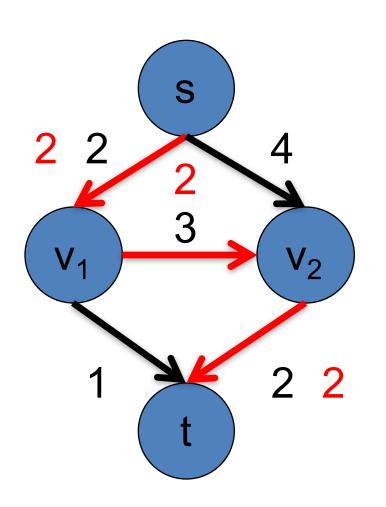


Find an s-t path where flow(a) < c(a) for all arcs

No more paths. Stop.

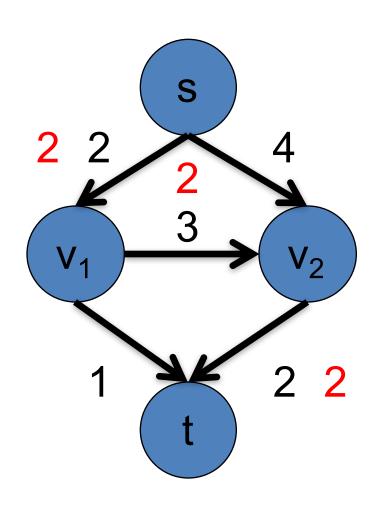
Will this give us maximum flow?

NO !!!



Find an s-t path where flow(a) < c(a) for all arcs

Pass maximum allowable flow through the arcs



Find an s-t path where flow(a) < c(a) for all arcs

No more paths. Stop.

Another method?

Incorrect Answer!!

Outline

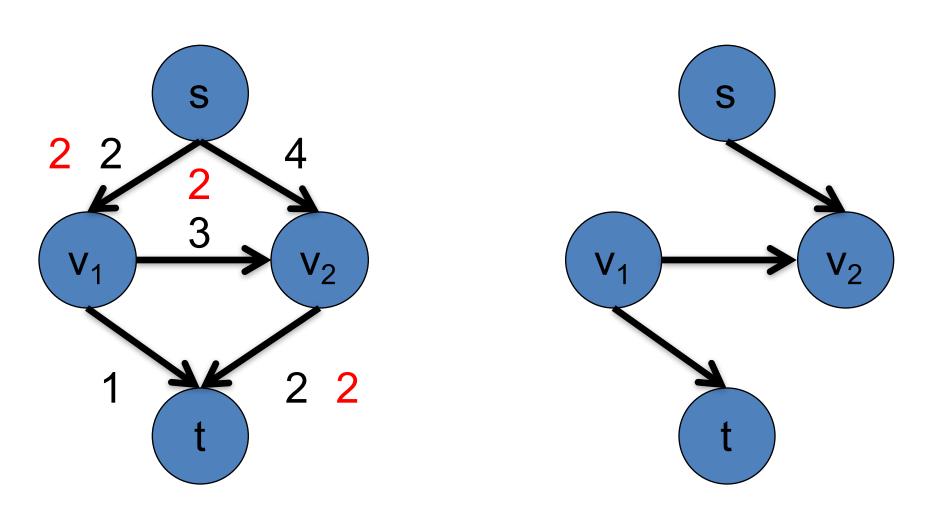
Preliminaries

- Maximum Flow
 - Residual Graph
 - Max-Flow Min-Cut Theorem

Algorithms

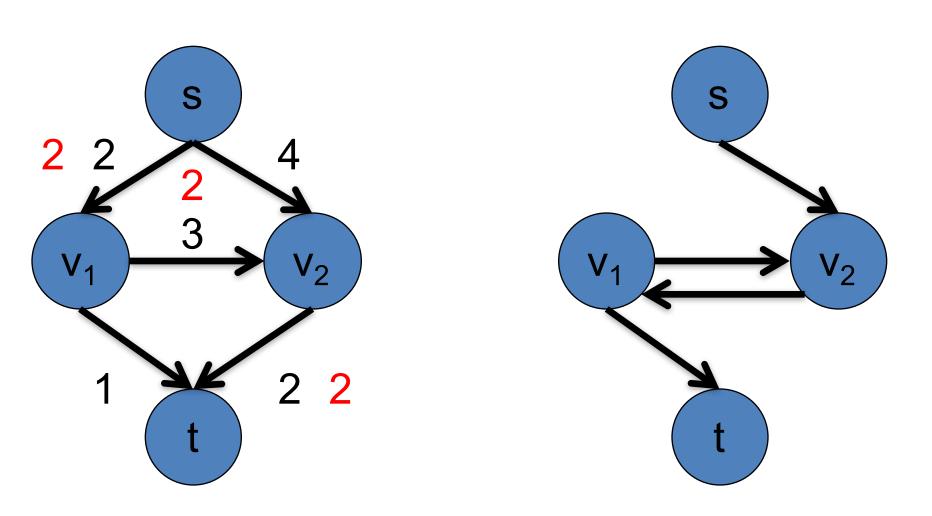
Energy minimization with max flow/min cut

Residual Graph

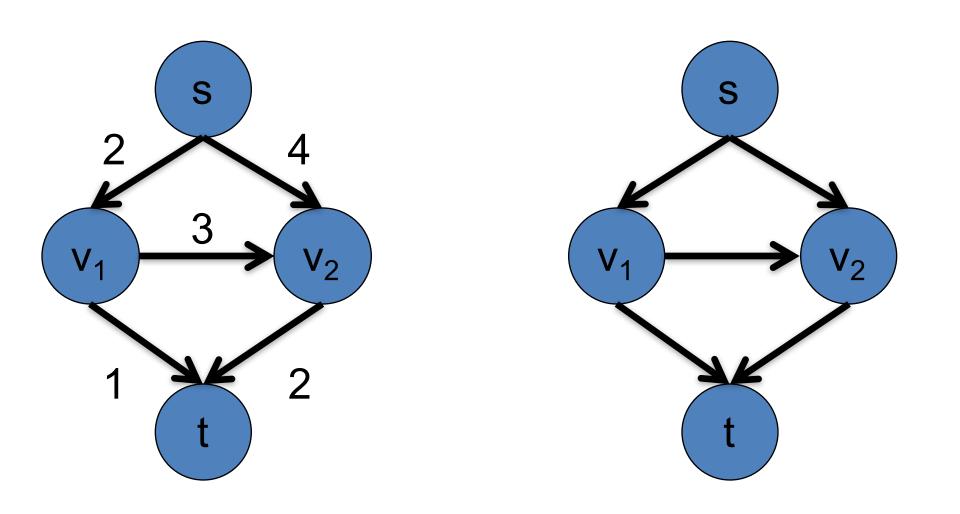


Arcs where flow(a) < c(a)

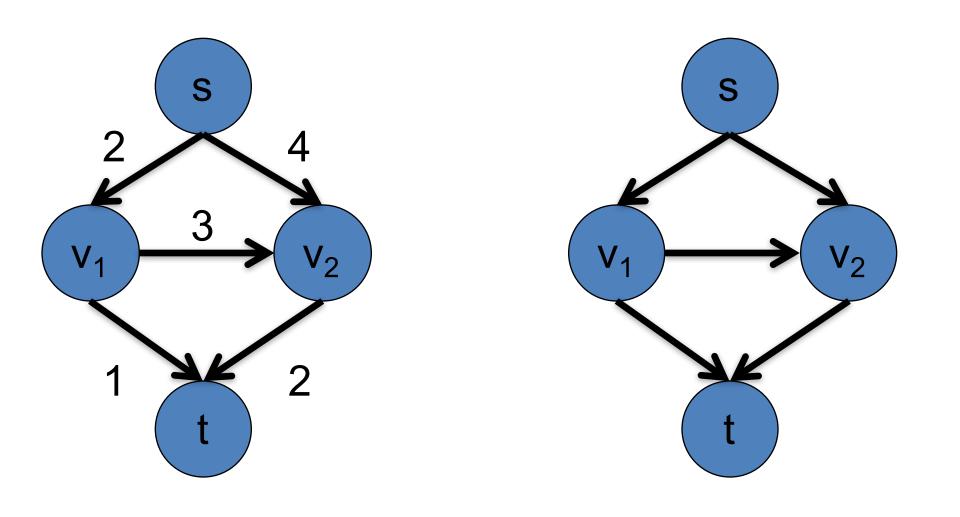
Residual Graph



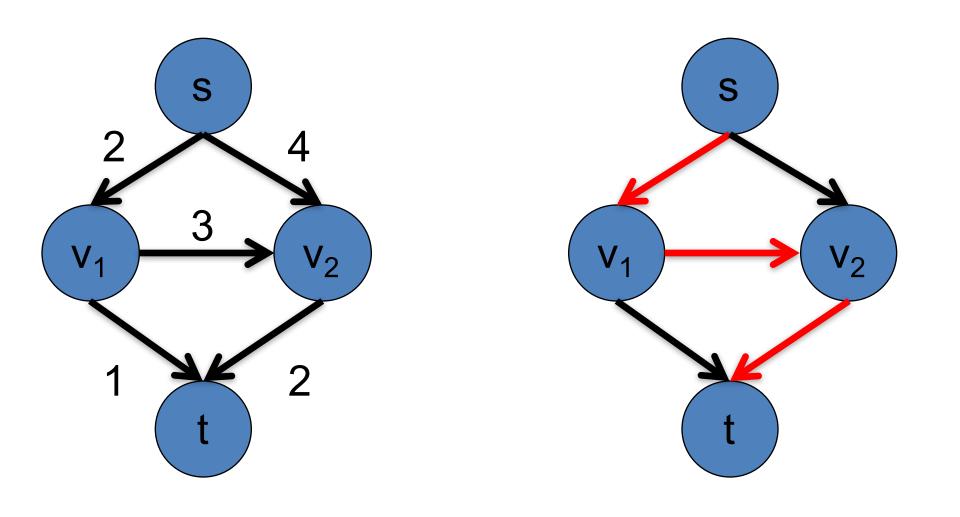
Including arcs to s and from t is not necessary Inverse of arcs where flow(a) > 0



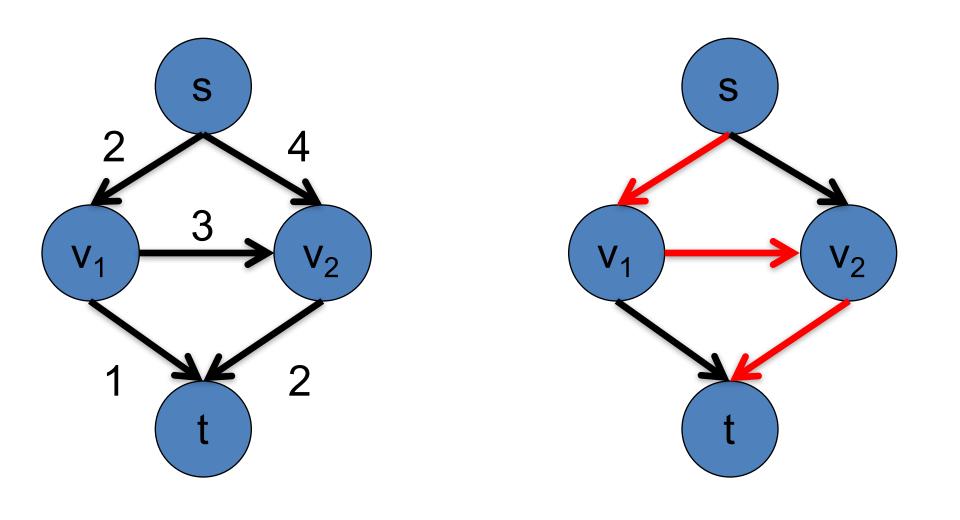
Start with zero flow.



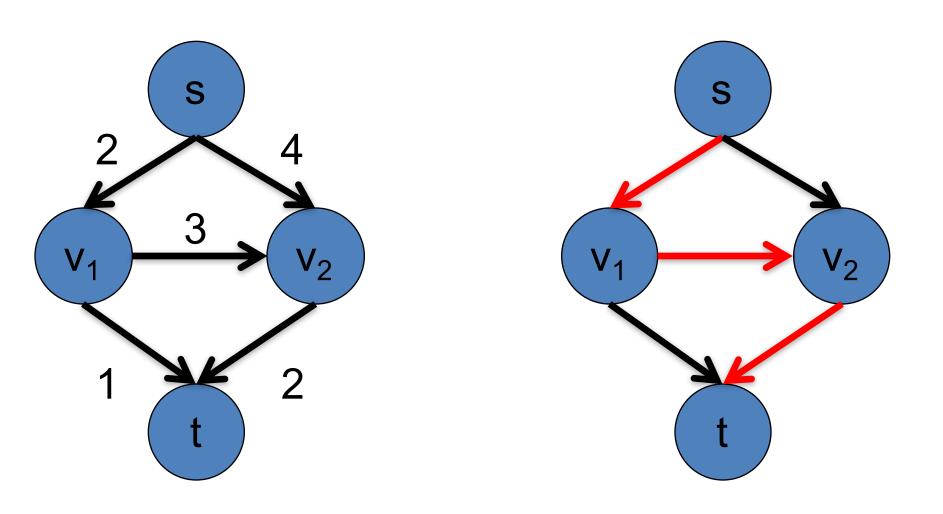
Find an s-t path in the residual graph.



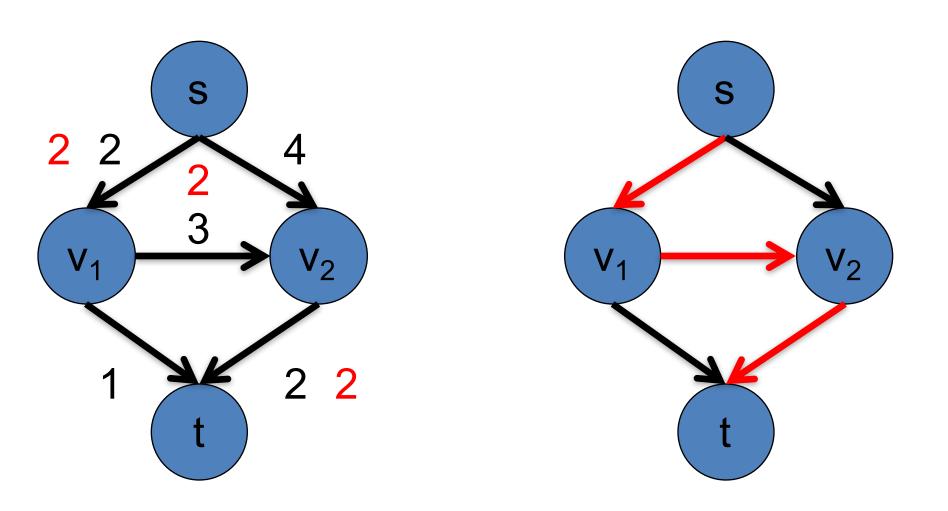
Find an s-t path in the residual graph.



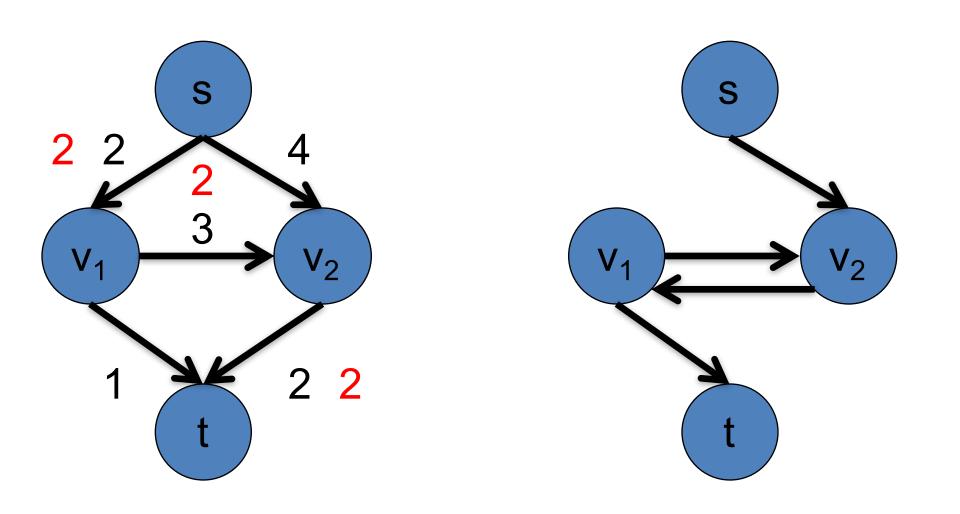
For inverse arcs in path, subtract flow K.



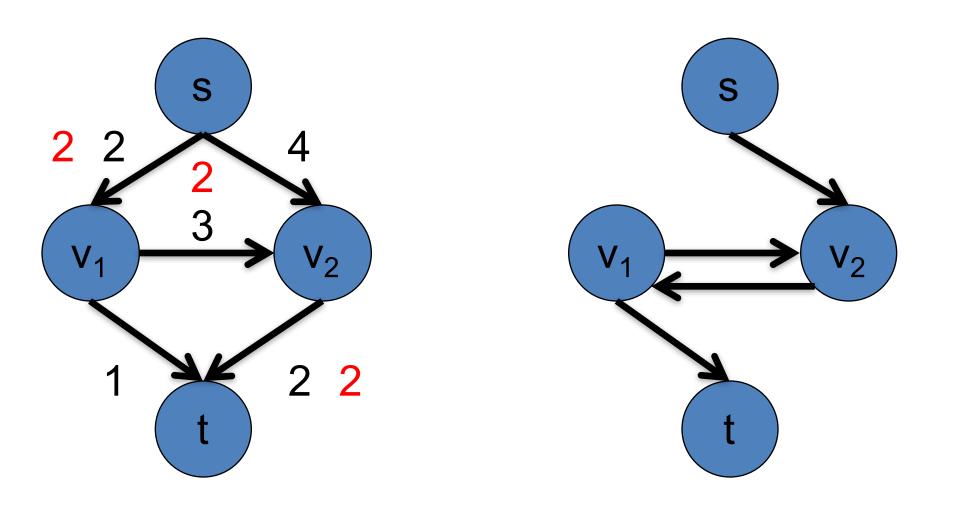
Choose maximum allowable value of K. For forward arcs in path, add flow K.



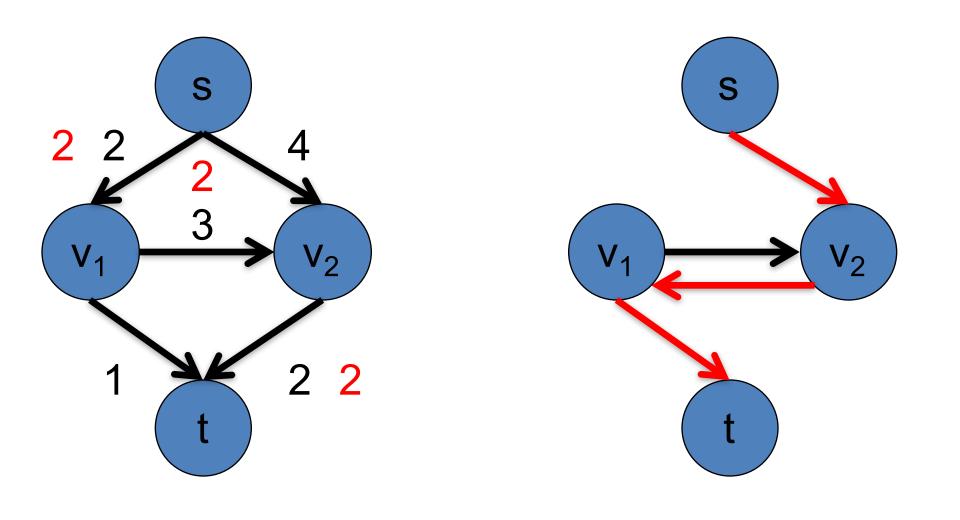
Choose maximum allowable value of K. For forward arcs in path, add flow K.



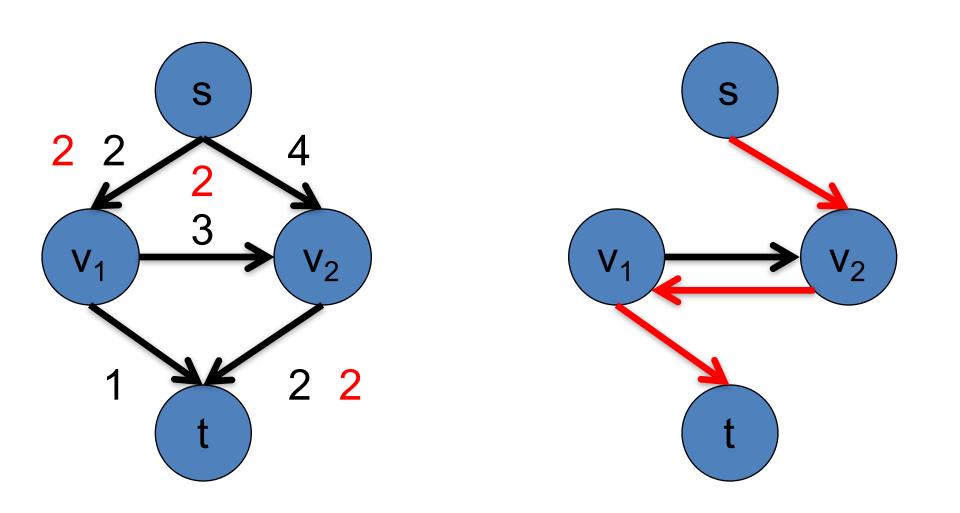
Update the residual graph.



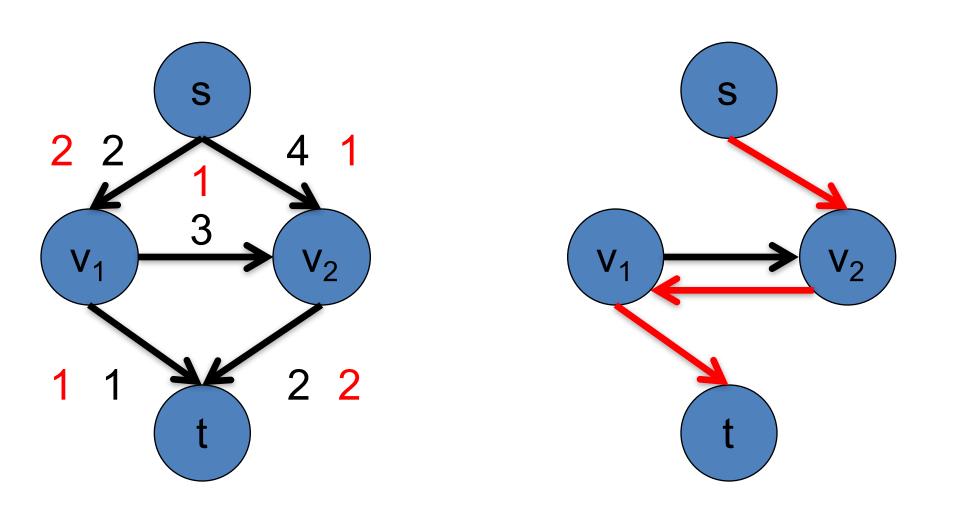
Find an s-t path in the residual graph.



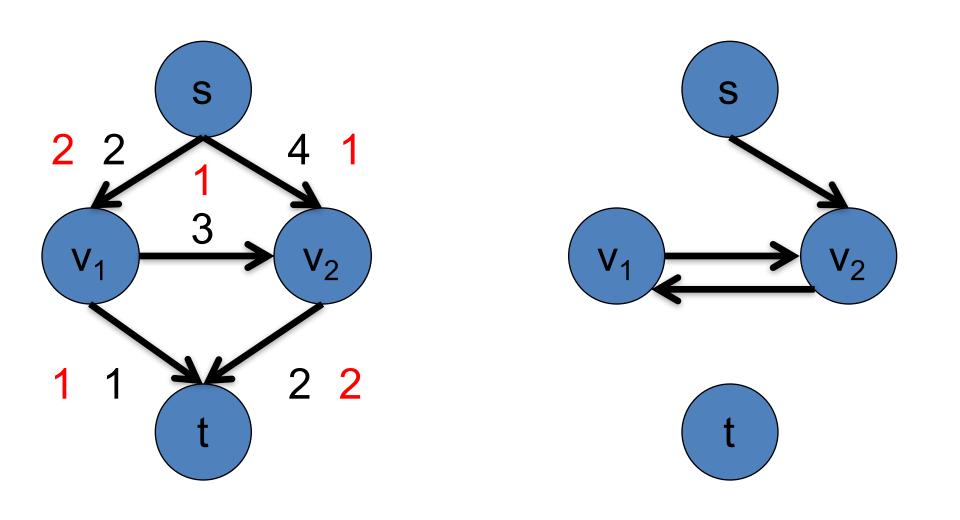
Find an s-t path in the residual graph.



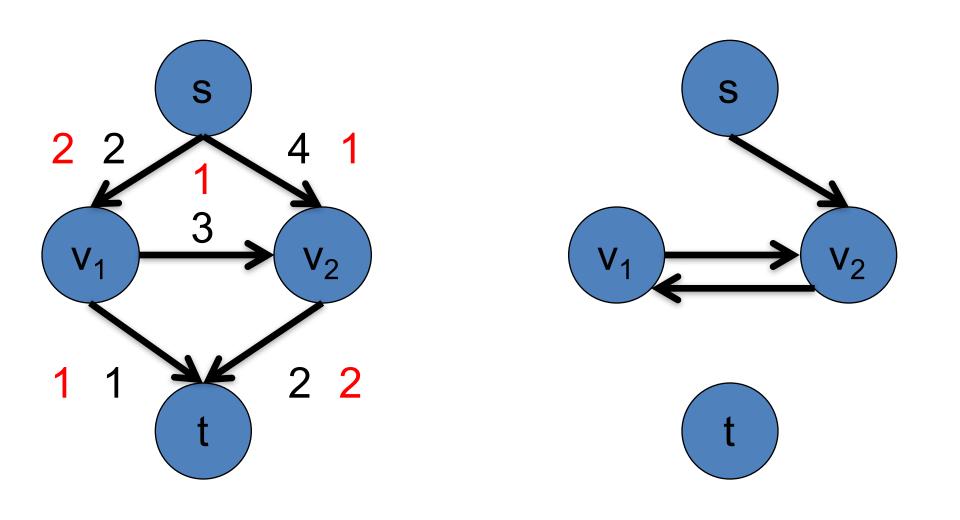
Choose maximum allowable value of K. Add K to (s,v_2) and (v_1,t) . Subtract K from (v_1,v_2) .

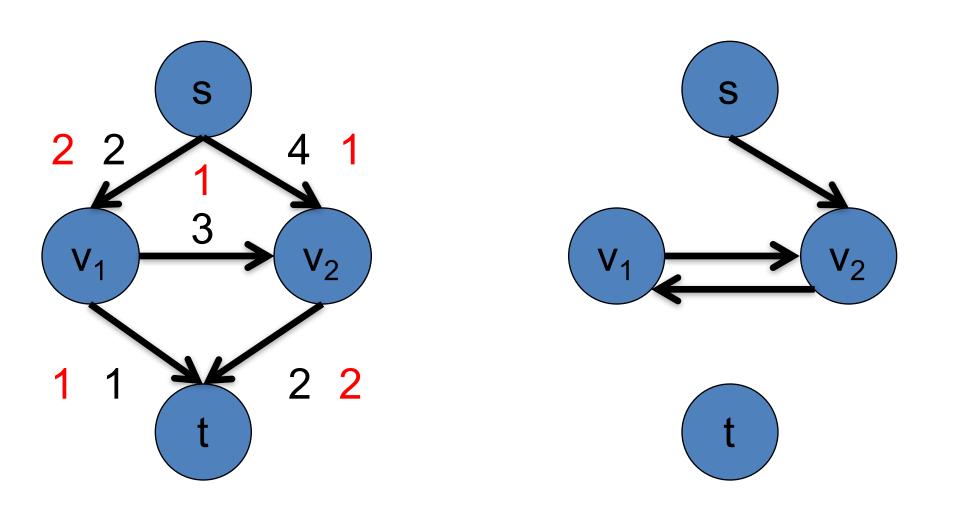


Choose maximum allowable value of K. Add K to (s,v_2) and (v_1,t) . Subtract K from (v_1,v_2) .

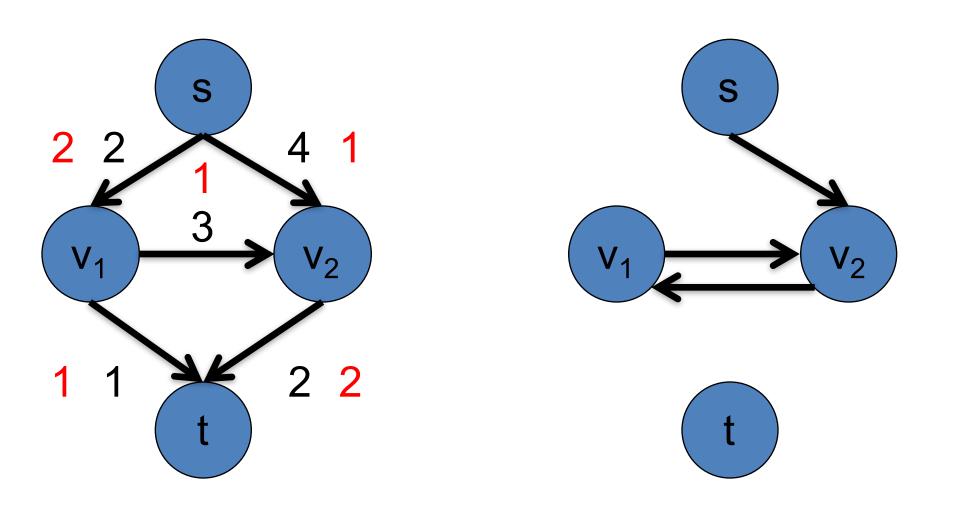


Update the residual graph.

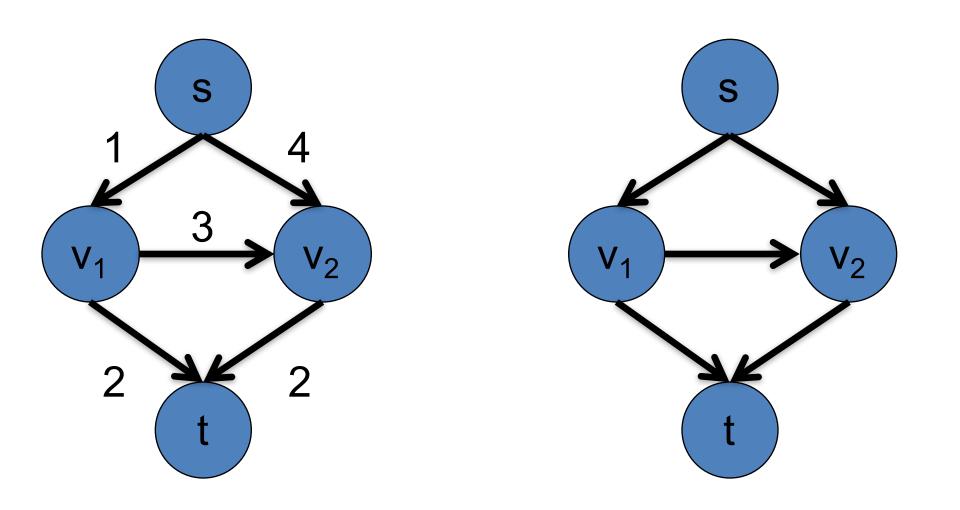




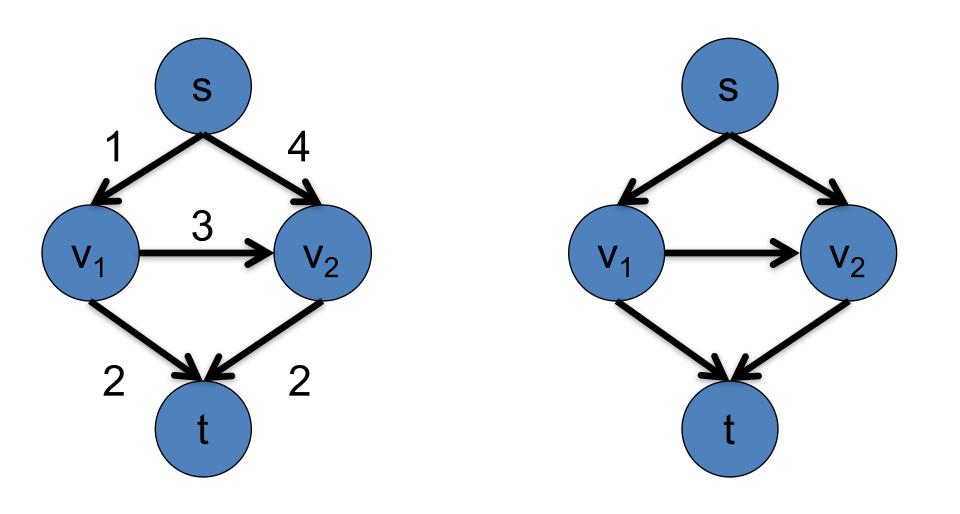
No more s-t paths. Stop.

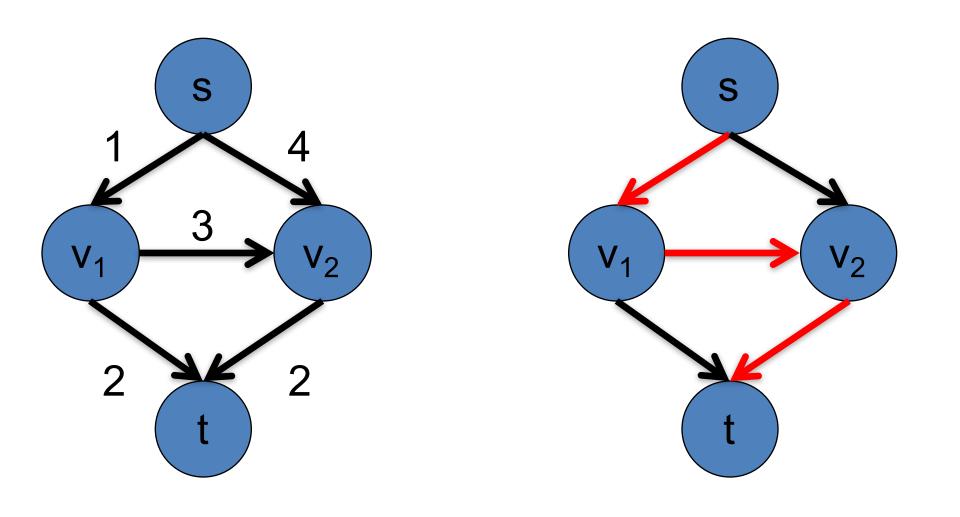


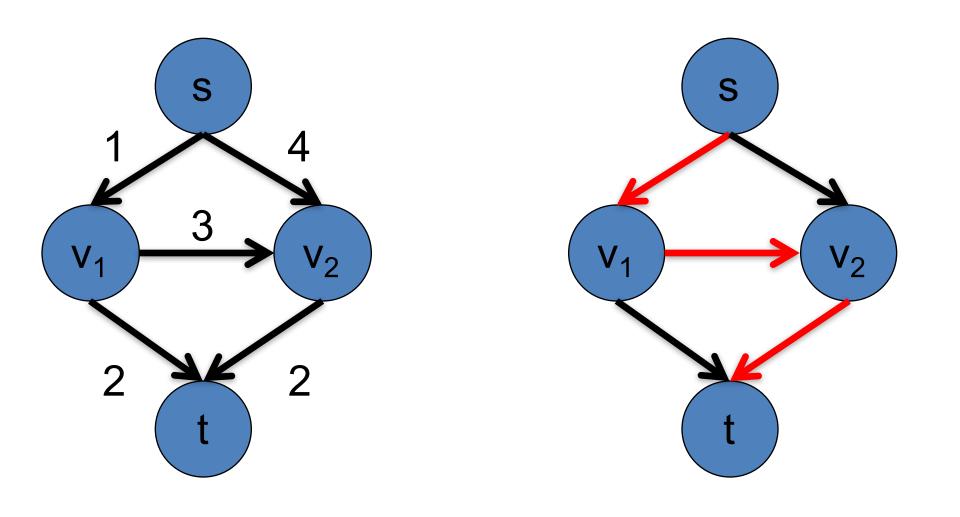
Correct Answer.



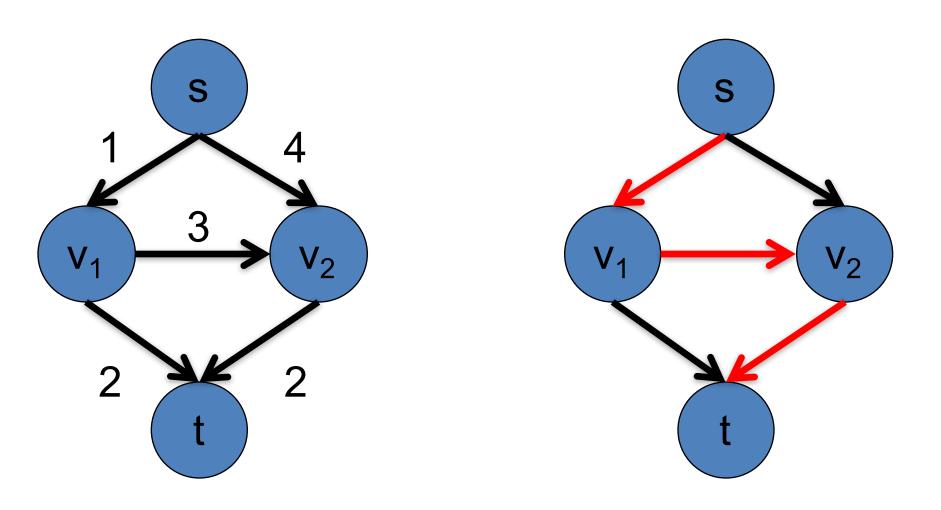
Start with zero flow



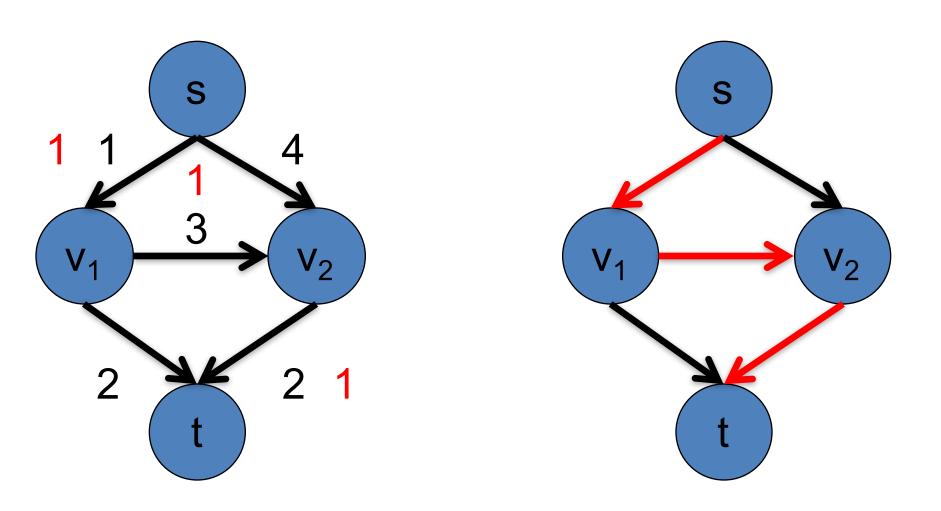




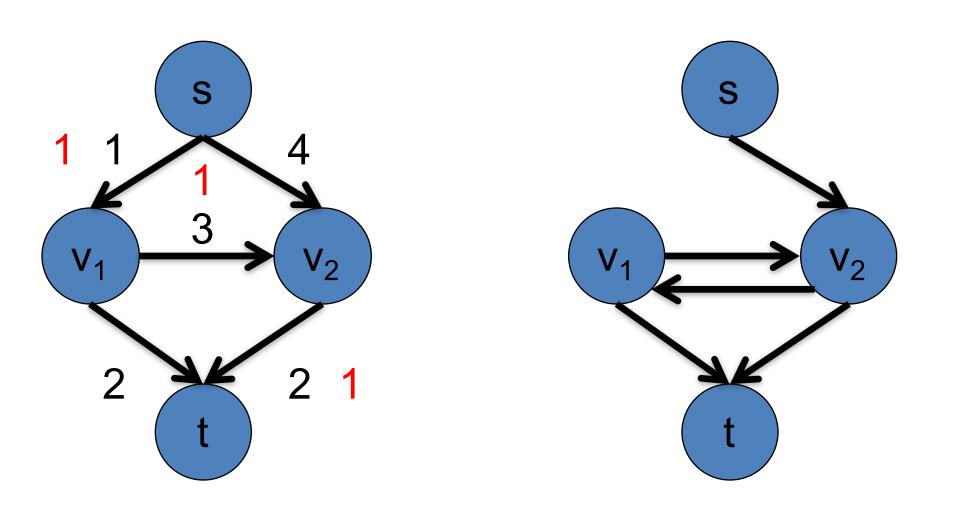
For inverse arcs in path, subtract flow K.



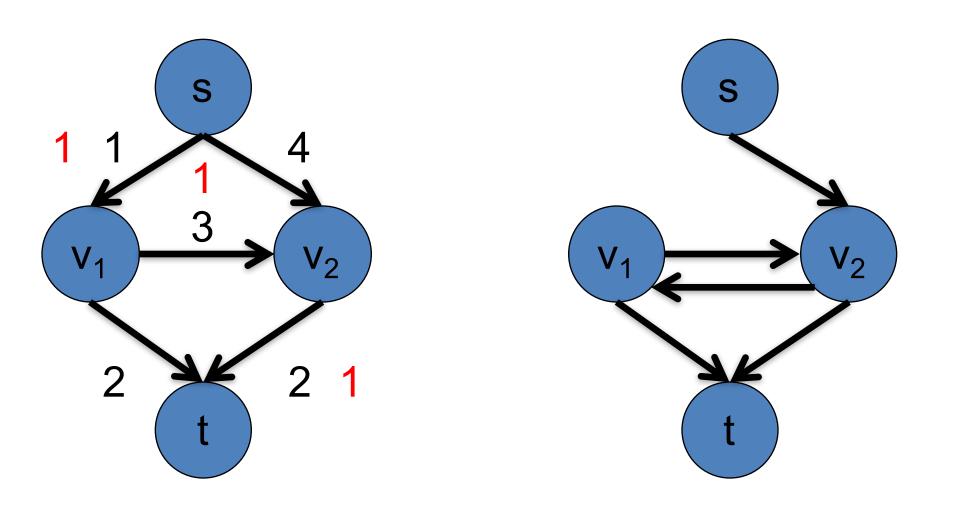
Choose maximum allowable value of K. For forward arcs in path, add flow K.

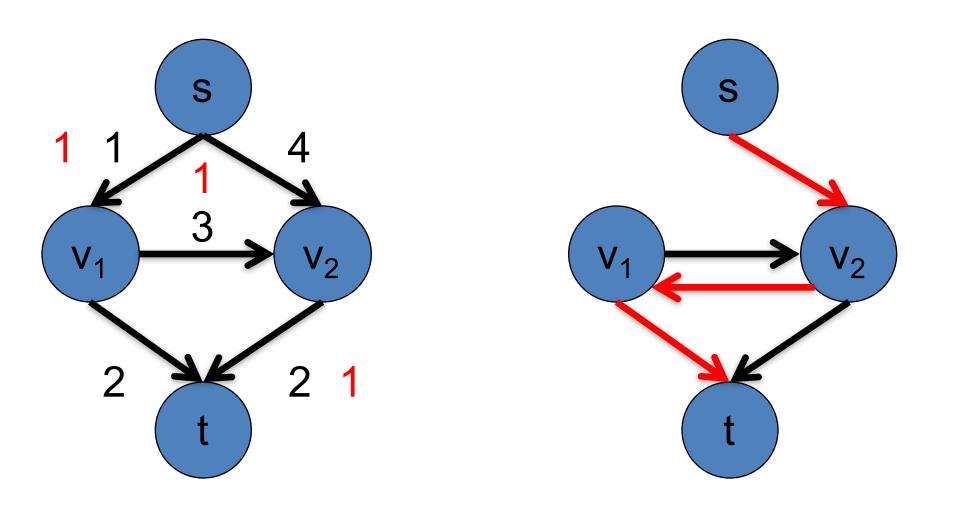


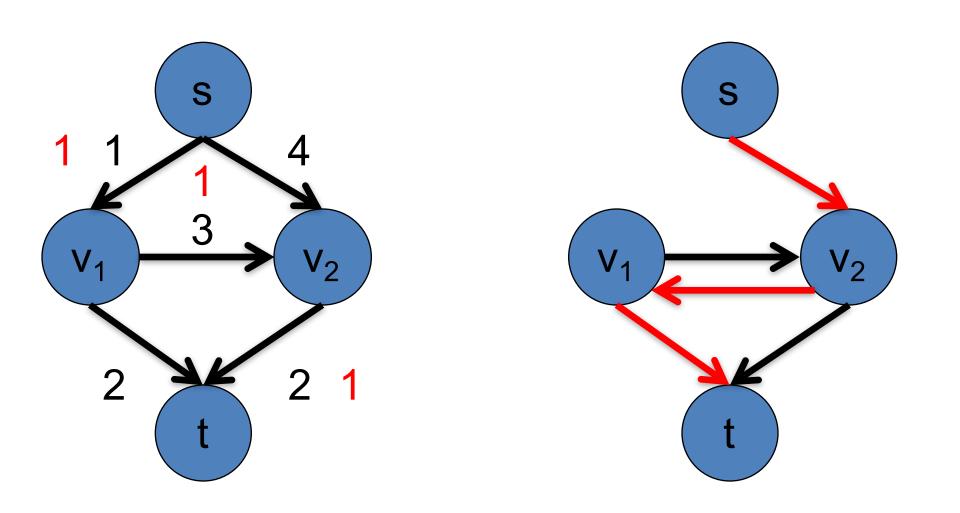
Choose maximum allowable value of K. For forward arcs in path, add flow K.



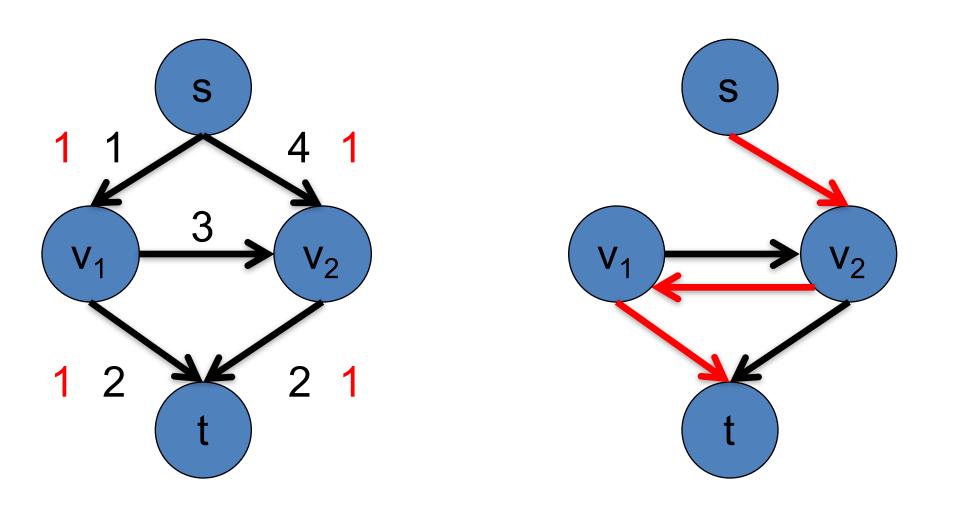
Update the residual graph.



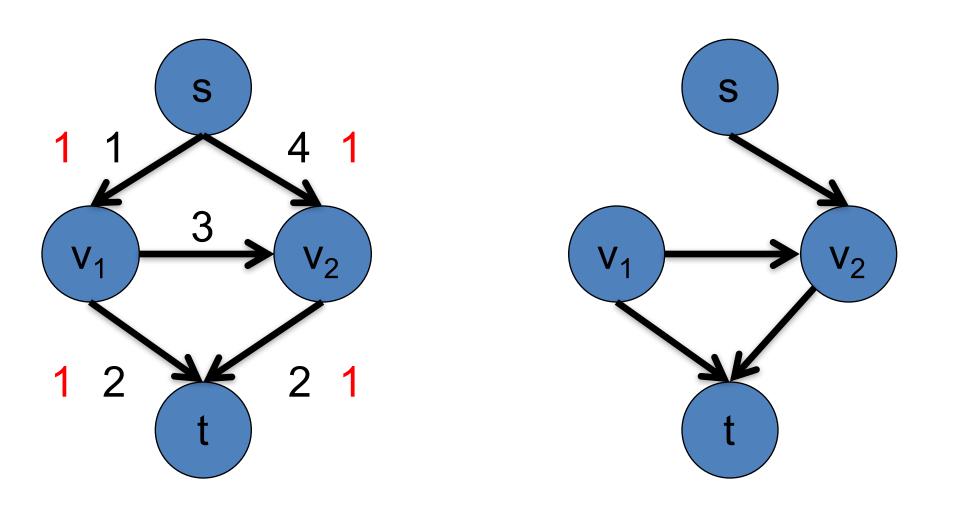




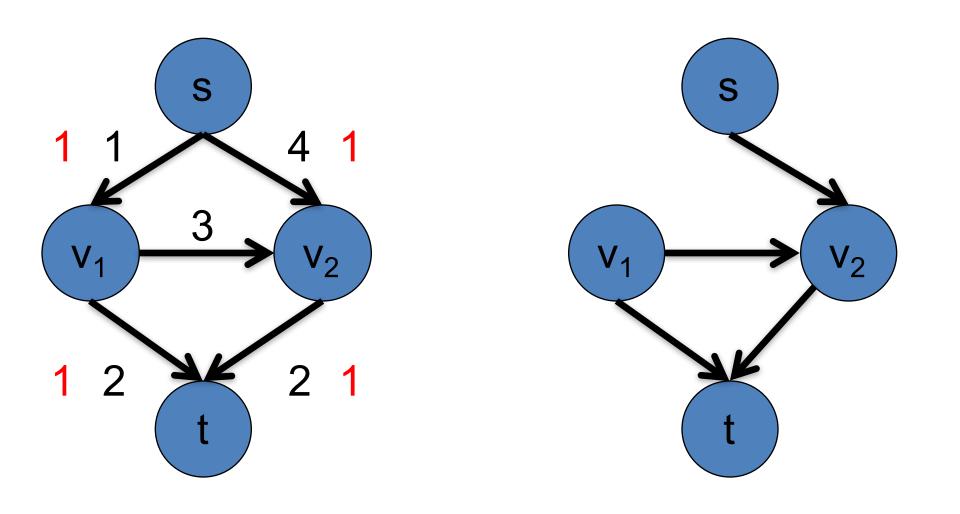
Choose maximum allowable value of K. Add K to (s,v_2) and (v_1,t) . Subtract K from (v_1,v_2) .

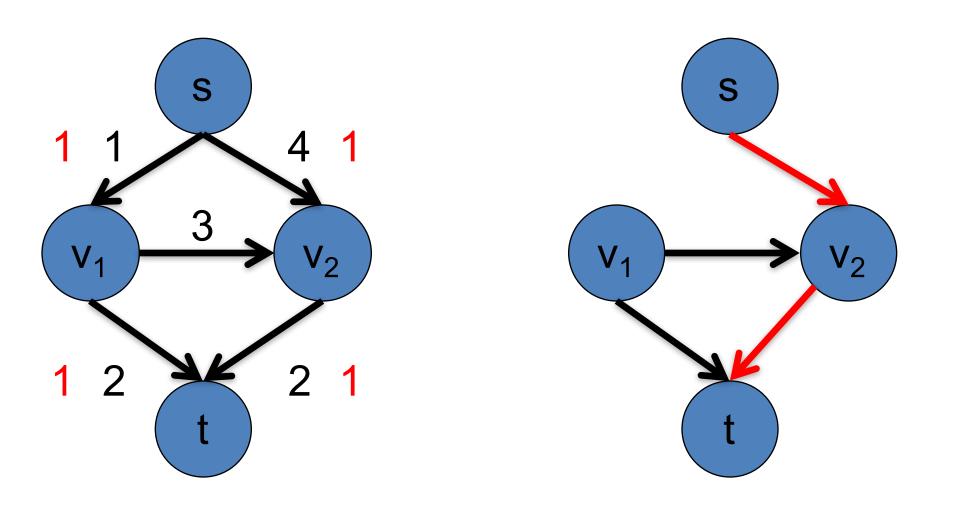


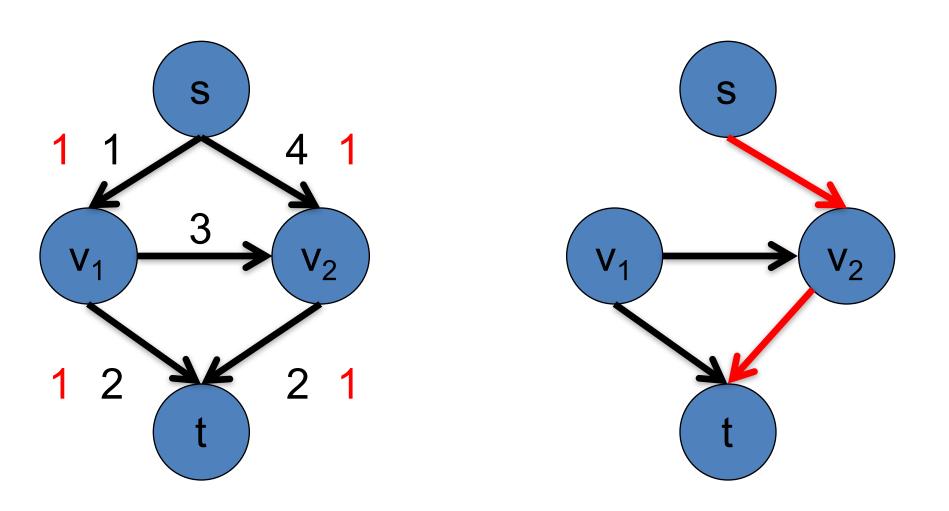
Choose maximum allowable value of K. Add K to (s,v_2) and (v_1,t) . Subtract K from (v_1,v_2) .



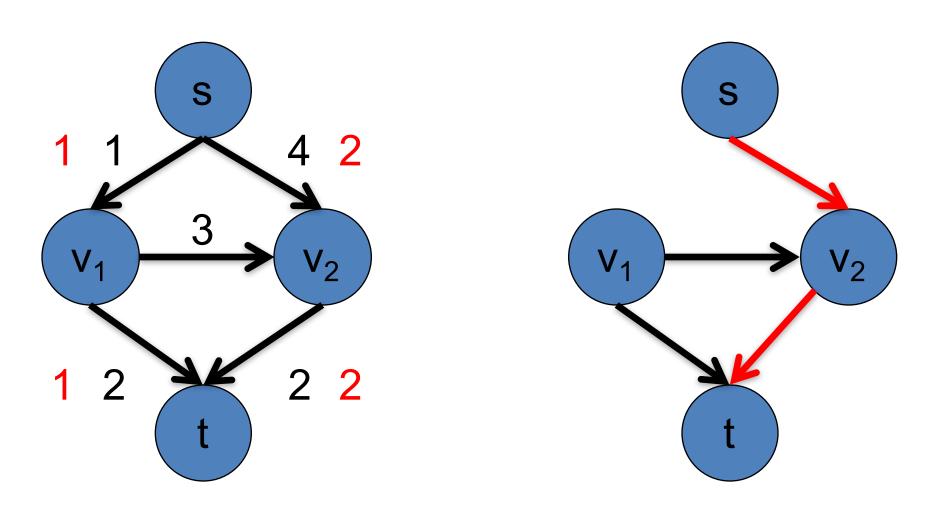
Update the residual graph.



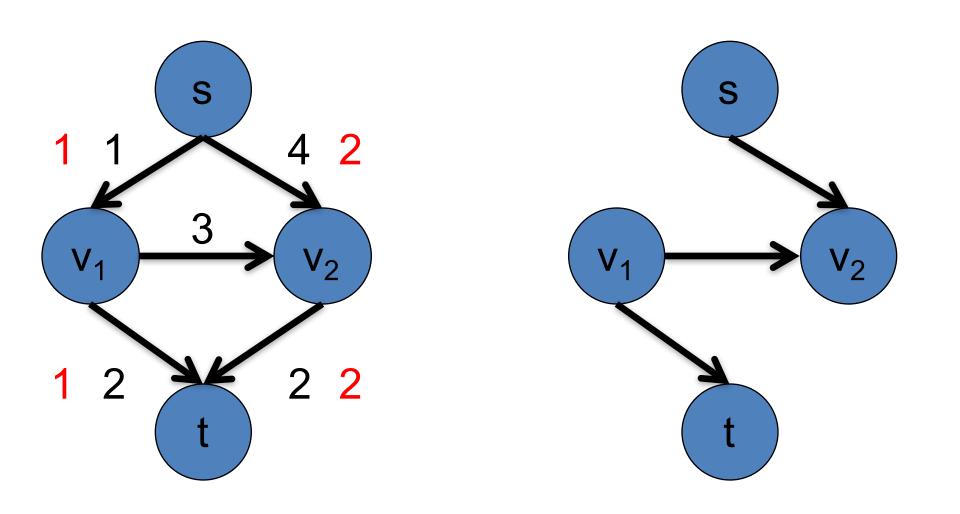




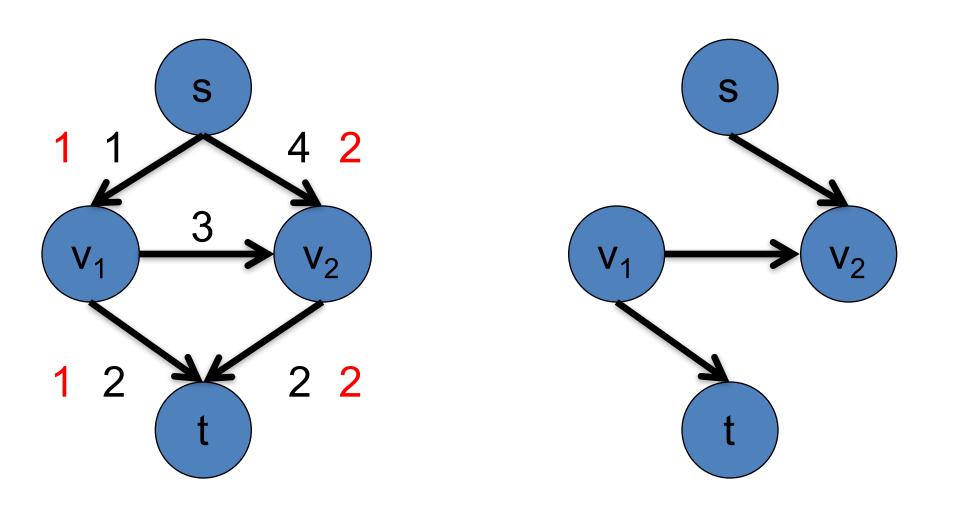
Choose maximum allowable value of K. Add K to (s,v_2) and (v_2,t) .



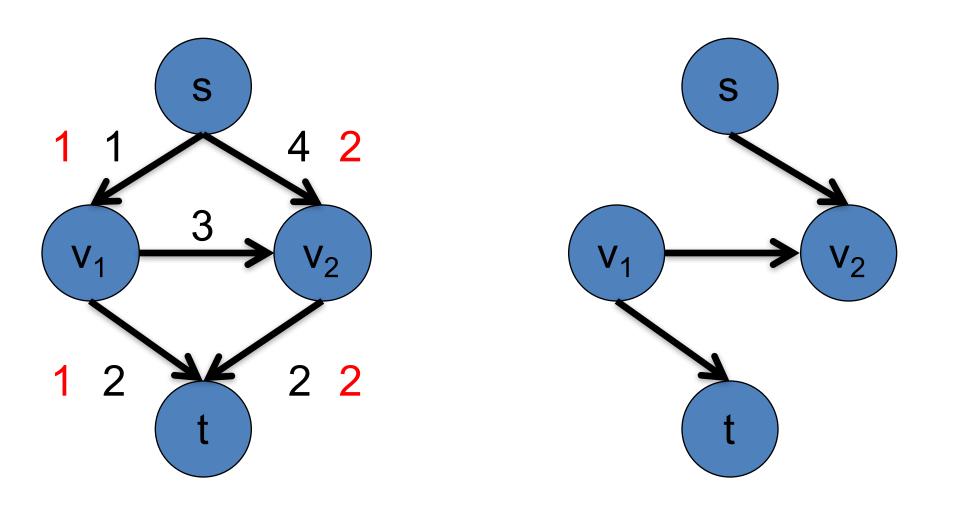
Choose maximum allowable value of K. Add K to (s,v_2) and (v_2,t) .



Update residual graph.



No more s-t paths. Stop.



How can I be sure this will always work?

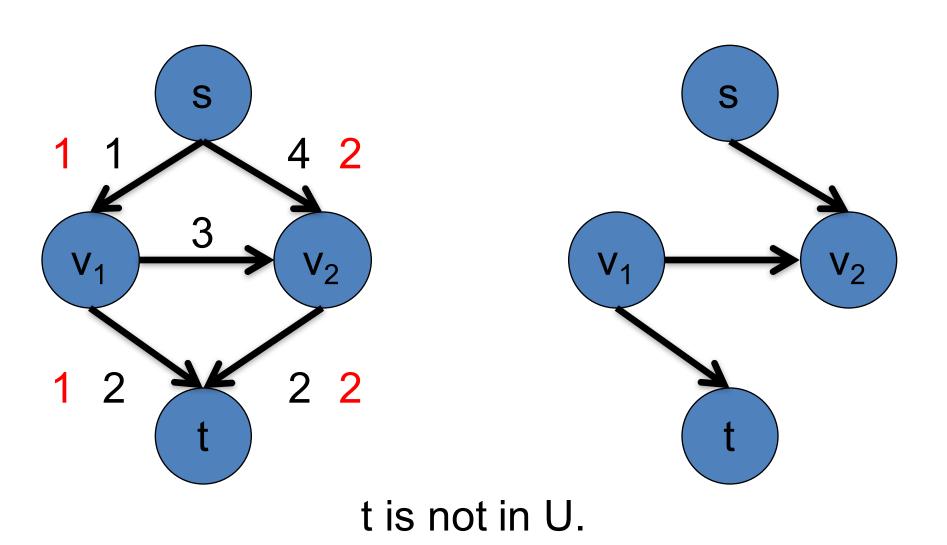
Outline

Preliminaries

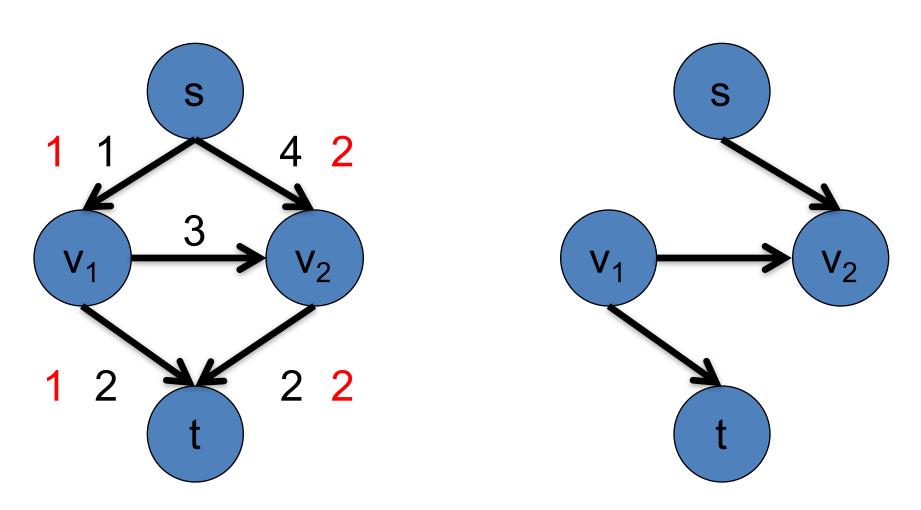
- Maximum Flow
 - Residual Graph
 - Max-Flow Min-Cut Theorem

Algorithms

Energy minimization with max flow/min cut

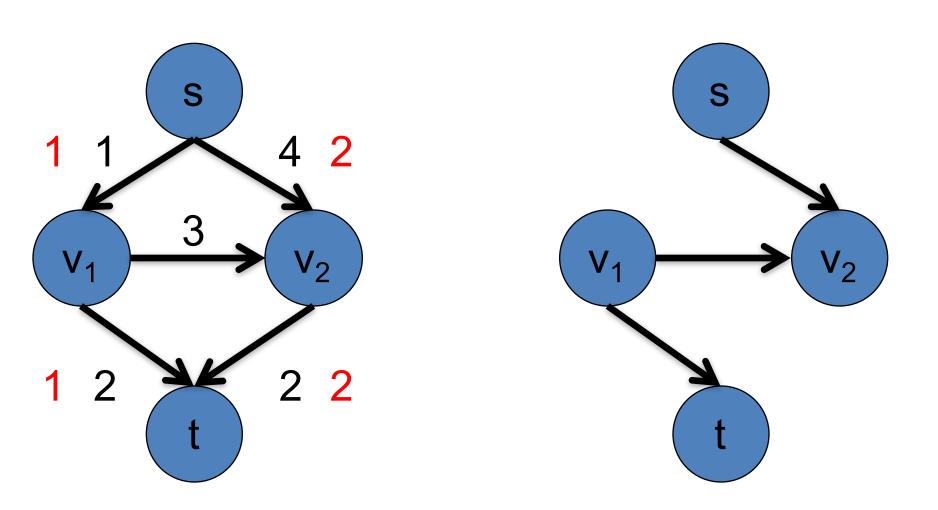


Let the subset of vertices U be reachable from s.



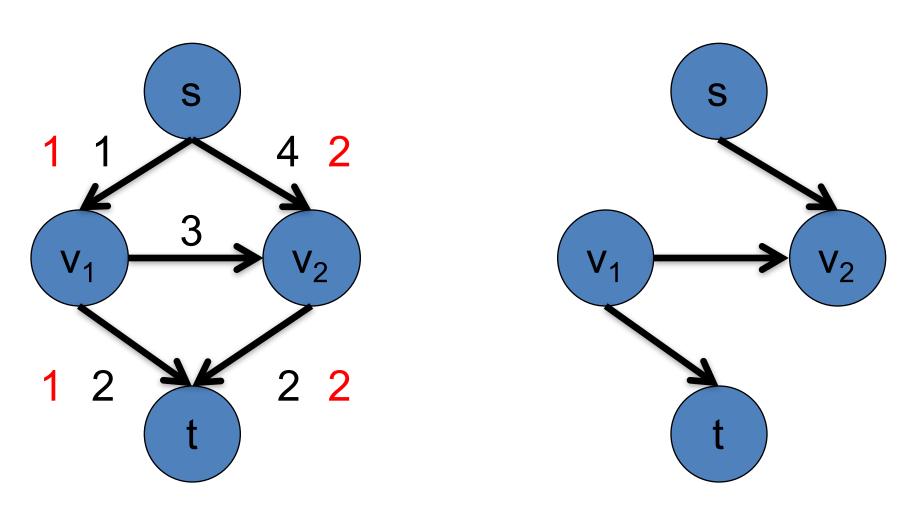
Or else a will be in the residual graph.

For all $a \in \text{out-arcs}(U)$, flow(a) = c(a).



Or else inverse of a will be in the residual graph.

For all $a \in \text{in-arcs}(U)$, flow(a) = 0.

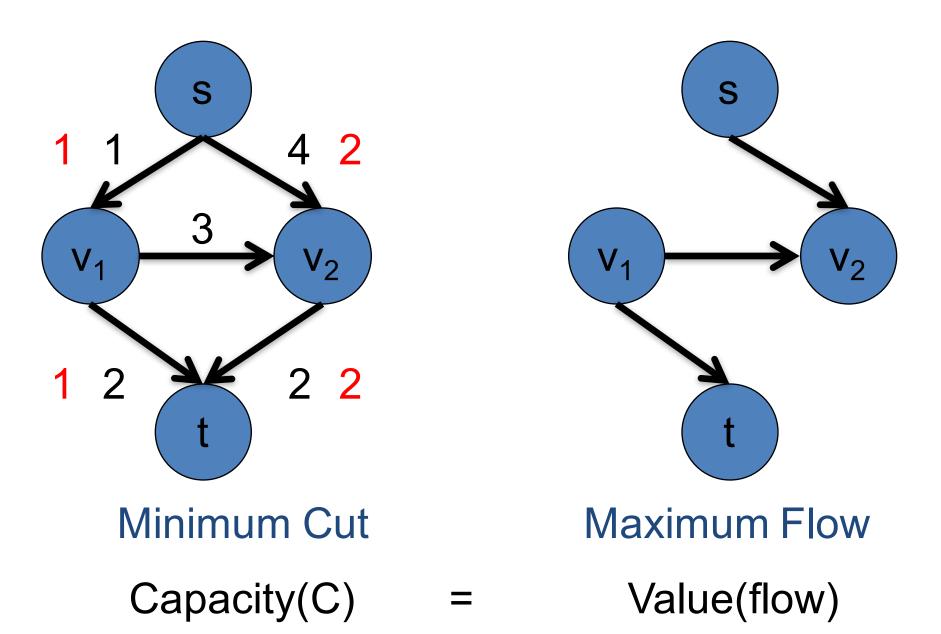


For all $a \in \text{out-arcs}(U)$, flow(a) = c(a). For all $a \in \text{in-arcs}(U)$, flow(a) = 0.

Flows vs. Cuts

Value of flow
$$= -E_{flow}(s)$$

 $= -E_{flow}(s) - \Sigma_{v \in U \setminus \{s\}} E_{flow}(v)$
 $= -E_{flow}(U)$
 $= flow(out-arcs(U))$
 $- flow(in-arcs(U))$
 $= Capacity of C$



Outline

Preliminaries

Maximum Flow

- Algorithms
 - Ford-Fulkerson Algorithm
 - Dinits Algorithm

Energy minimization with max flow/min cut

Start with flow = 0 for all arcs.

Find an s-t path in the residual graph.

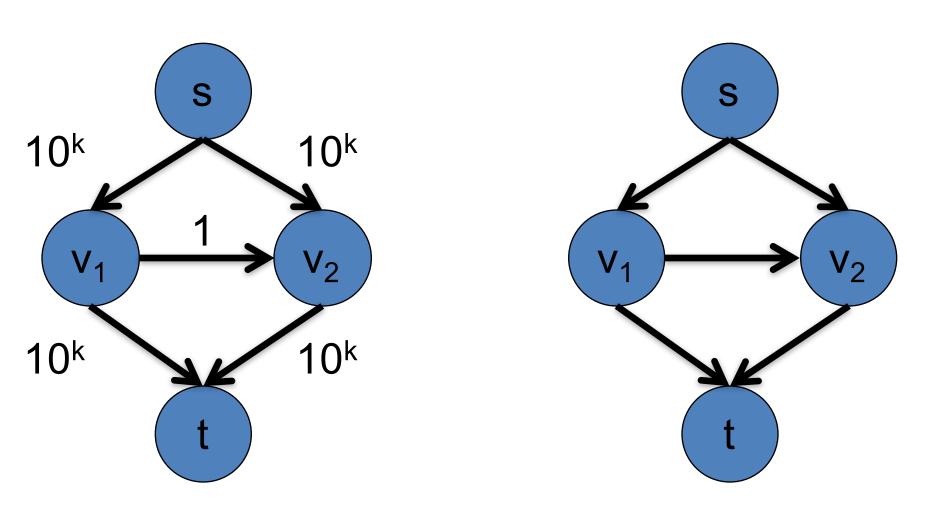
Pass maximum allowable flow.

Subtract from inverse arcs.

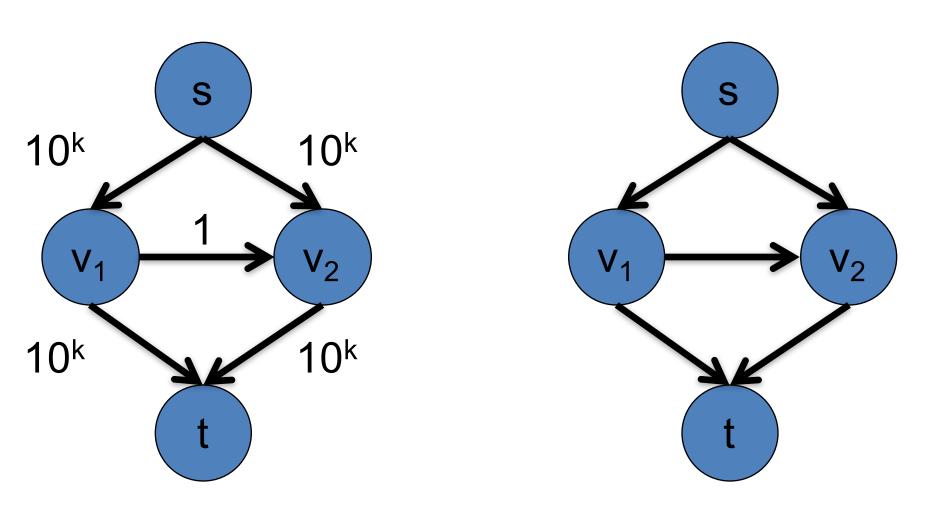
Add to forward arcs.

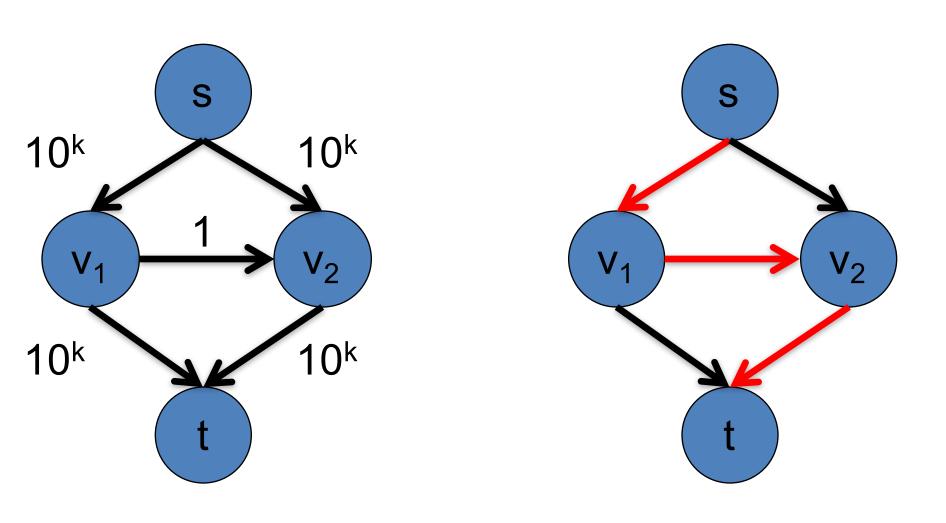
REPEAT

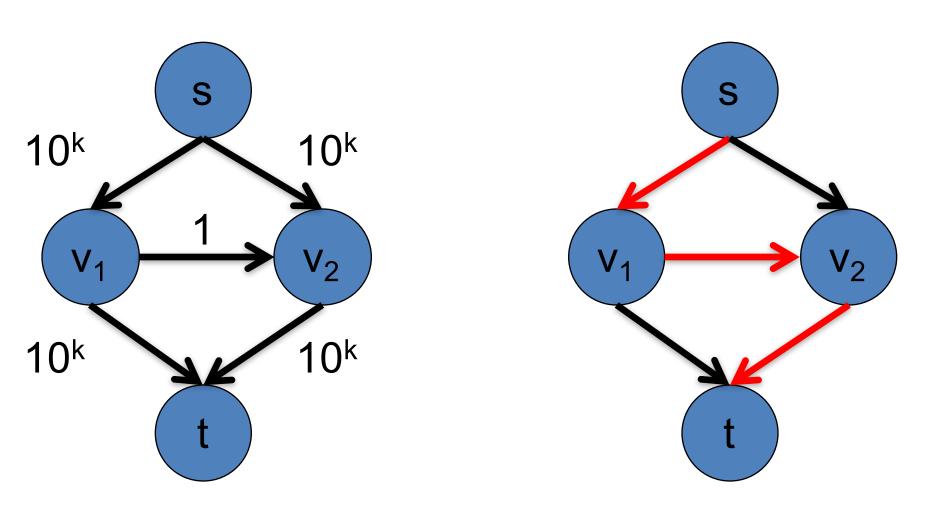
Until s and t are disjoint in the residual graph.



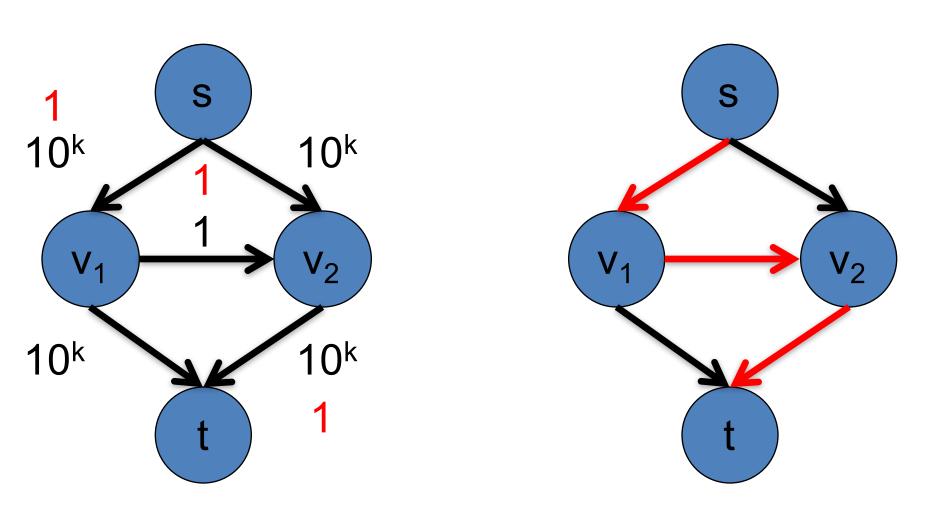
Start with zero flow



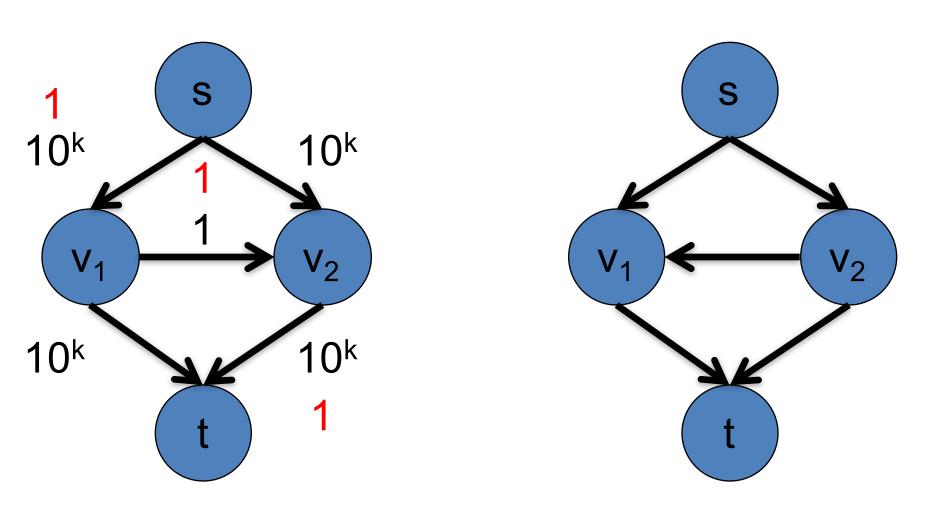




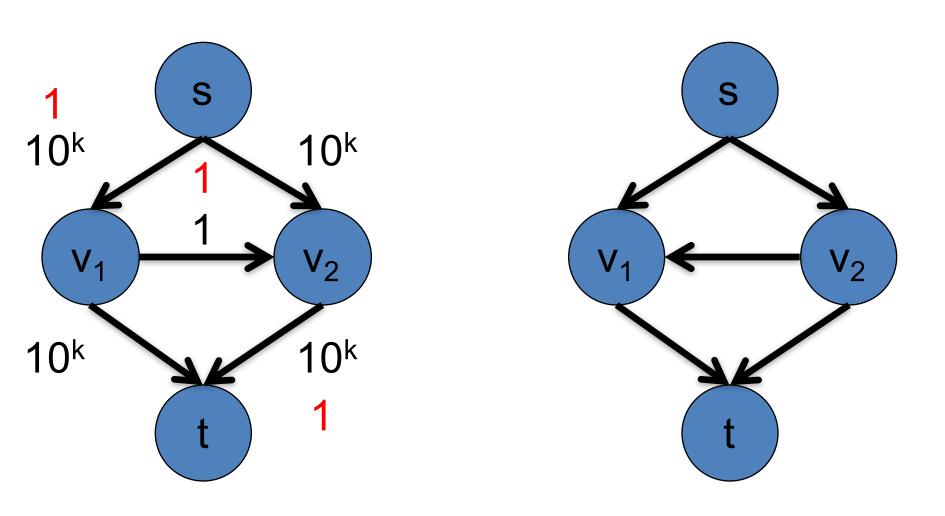
Pass the maximum allowable flow.



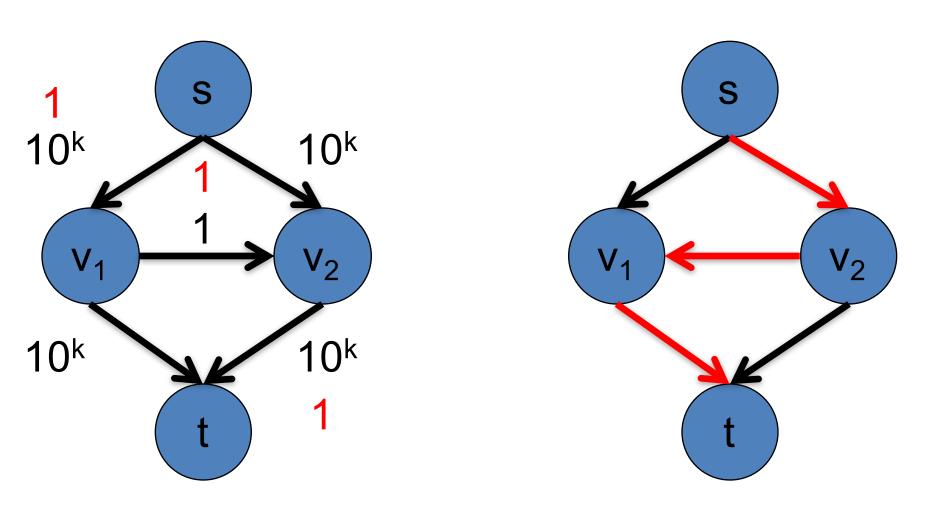
Pass the maximum allowable flow.



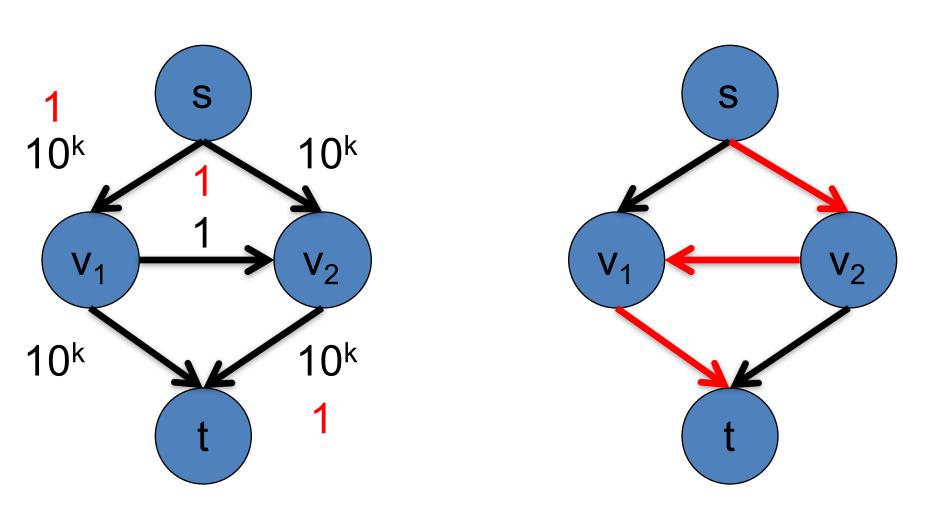
Update the residual graph.



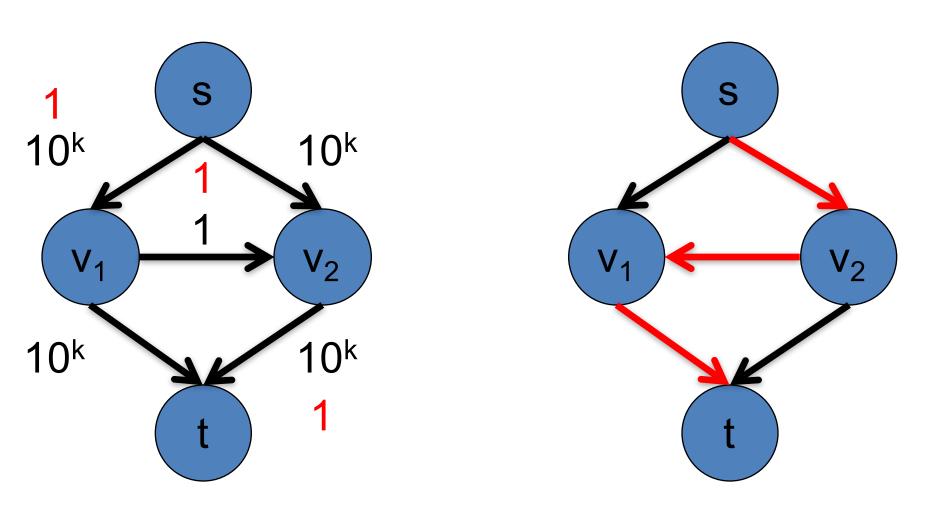
Find an s-t path in the residual graph.



Find an s-t path in the residual graph.

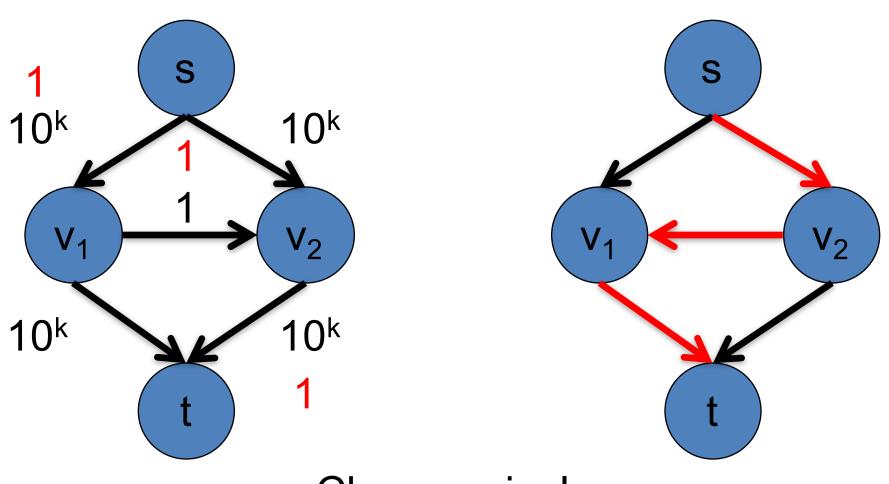


Complexity is exponential in k.



For examples, see Uri Zwick, 1993

Irrational arc lengths can lead to infinite iterations.



Choose wisely.

There are good paths and bad paths.

Outline

Preliminaries

Maximum Flow

- Algorithms
 - Ford-Fulkerson Algorithm
 - Dinits Algorithm

Energy minimization with max flow/min cut

Start with flow = 0 for all arcs.

Find the minimum s-t path in the residual graph.

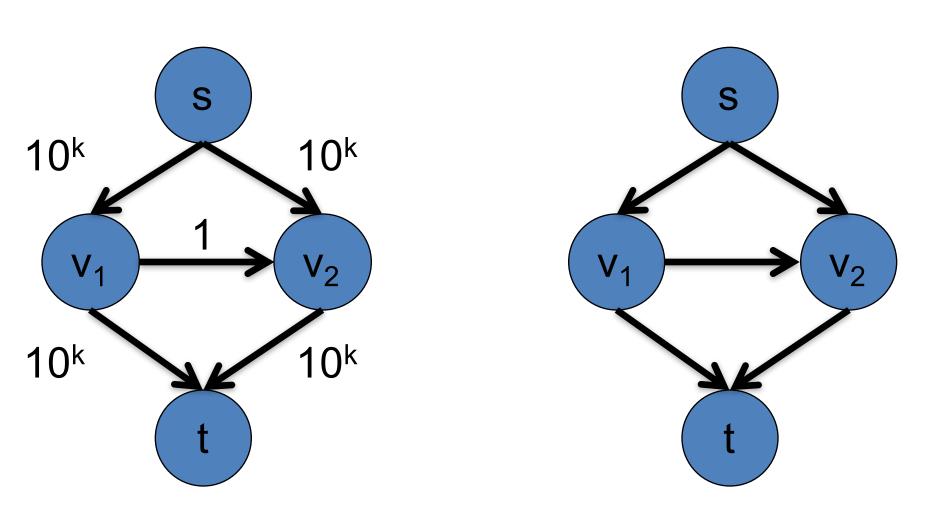
Pass maximum allowable flow.

Subtract from inverse arcs.

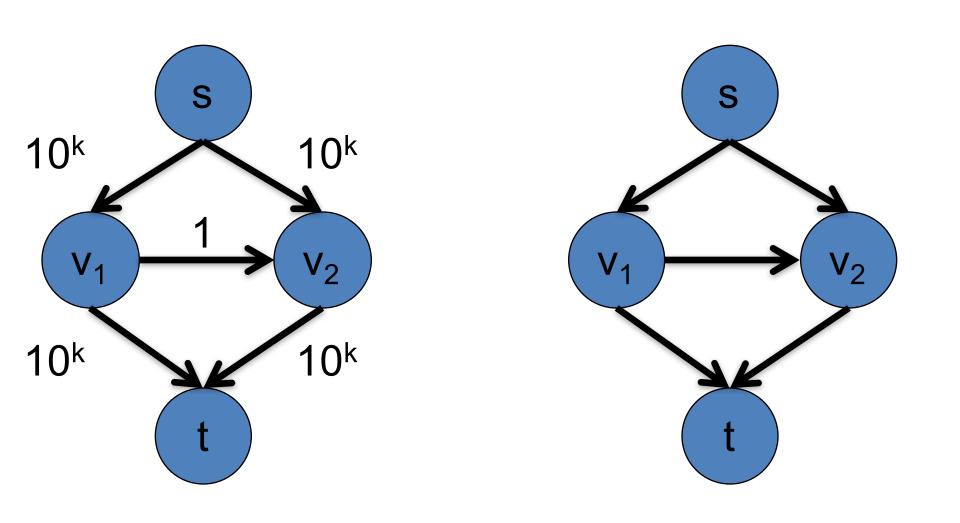
Add to forward arcs.

REPEAT

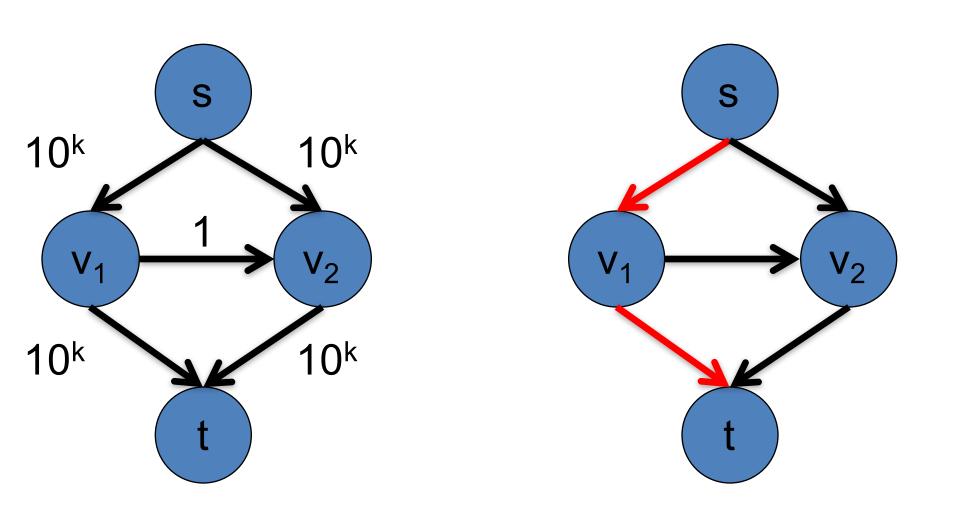
Until s and t are disjoint in the residual graph.



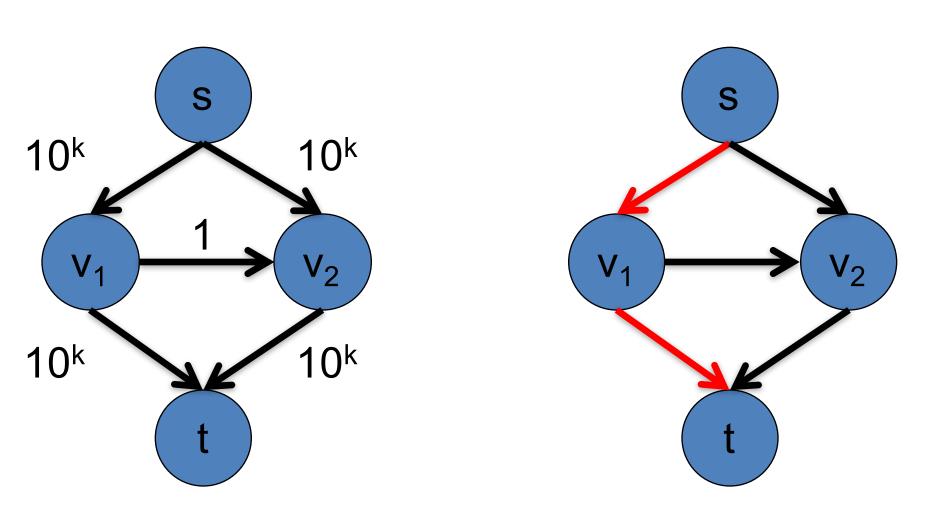
Start with zero flow



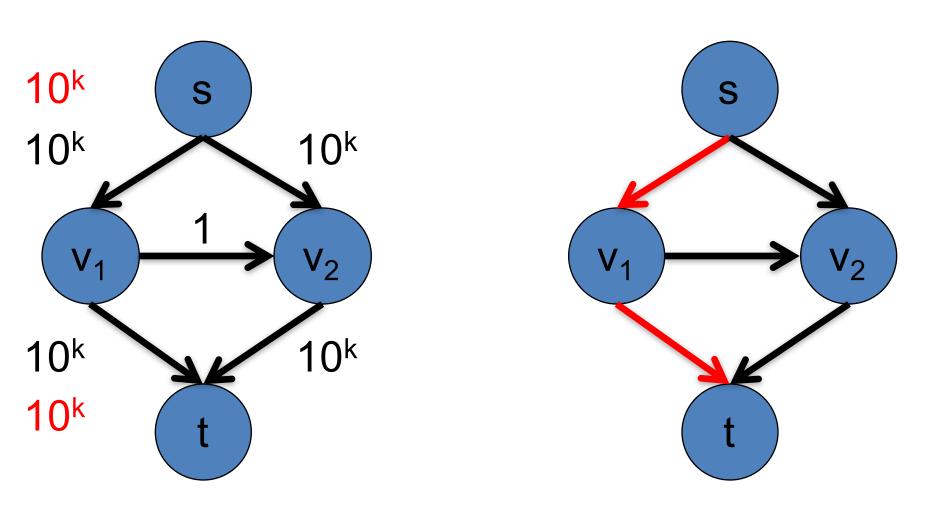
Find the minimum s-t path in the residual graph.



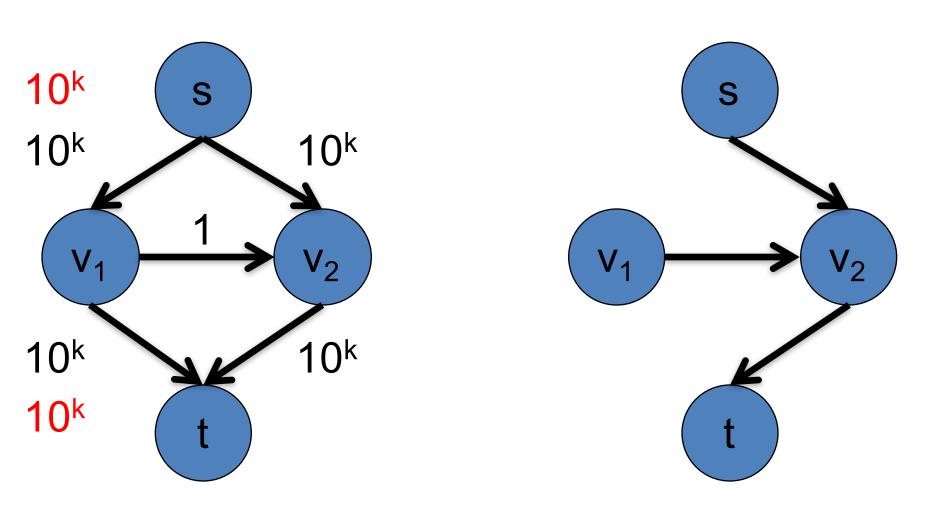
Find the minimum s-t path in the residual graph.



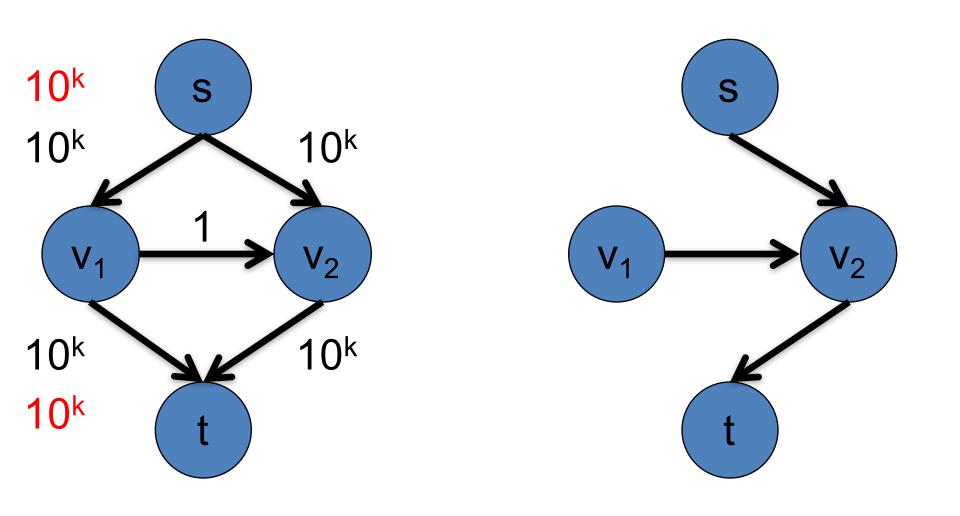
Pass the maximum allowable flow.



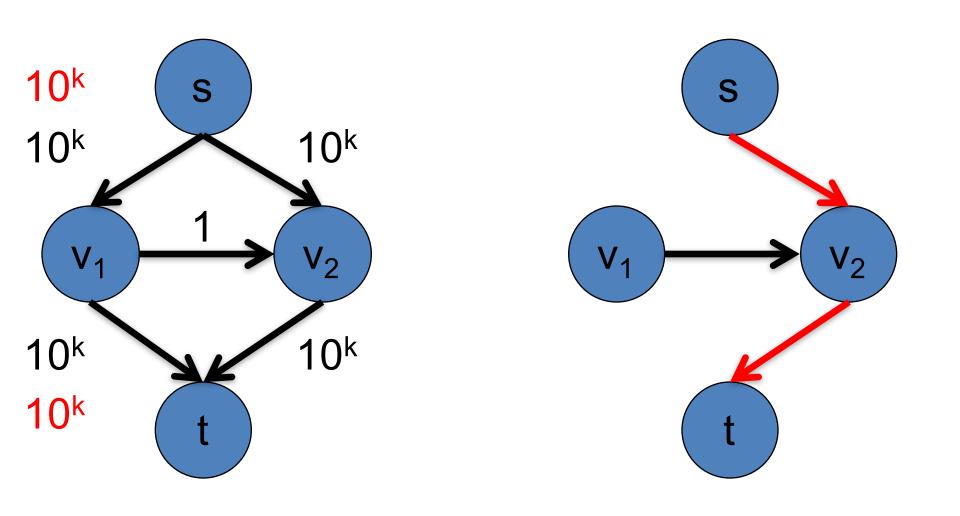
Pass the maximum allowable flow.



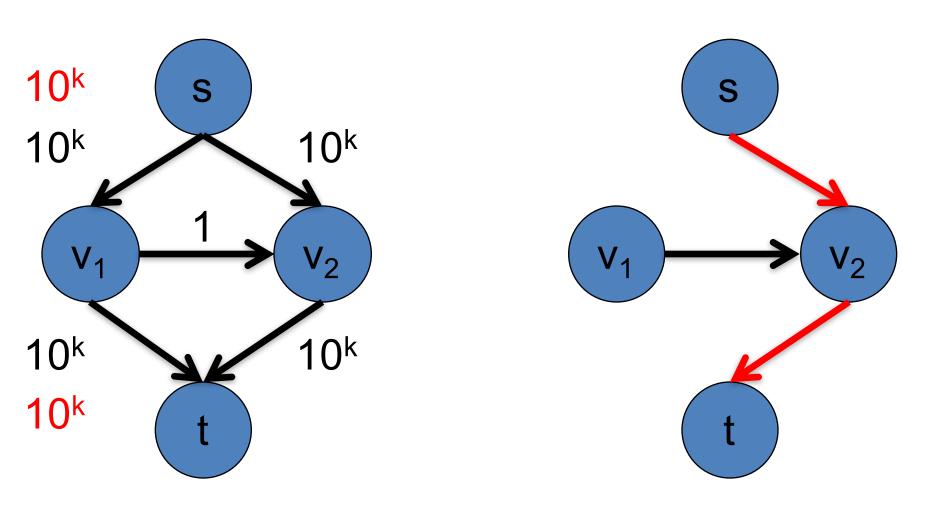
Update the residual graph.



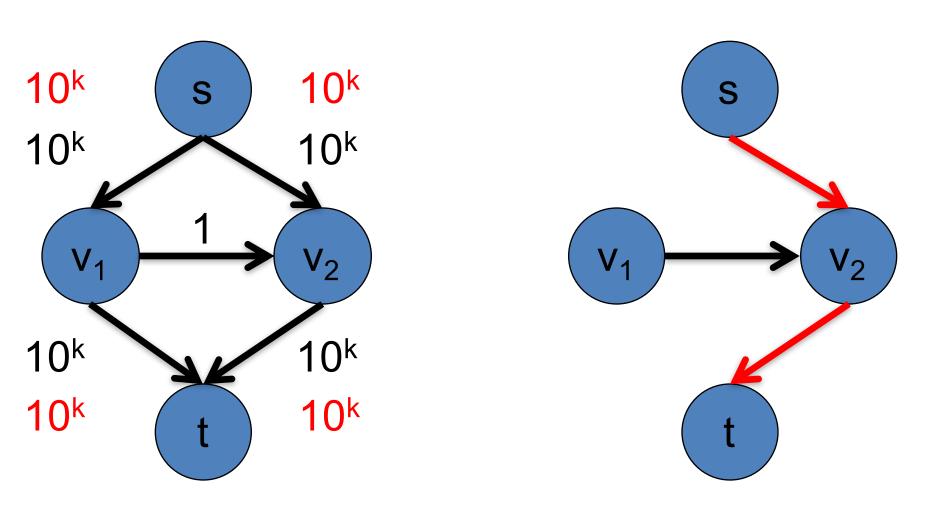
Find the minimum s-t path in the residual graph.



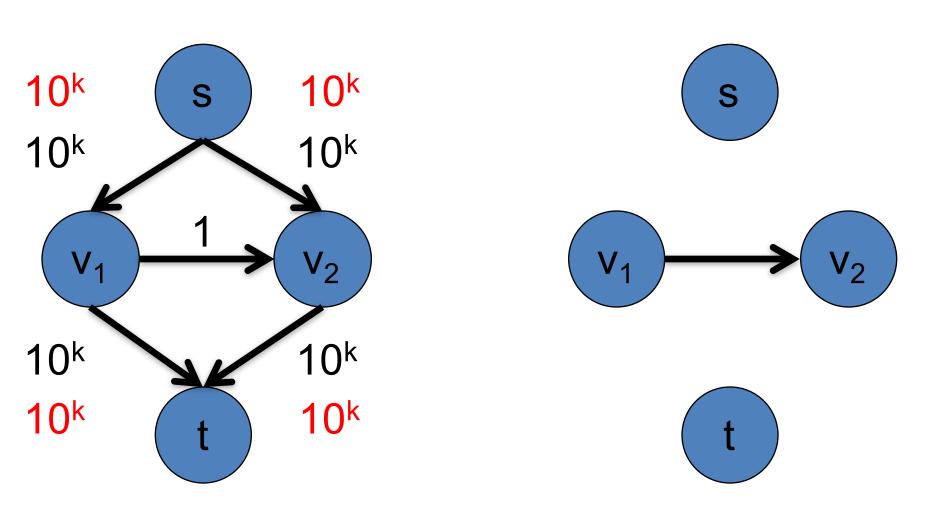
Find the minimum s-t path in the residual graph.



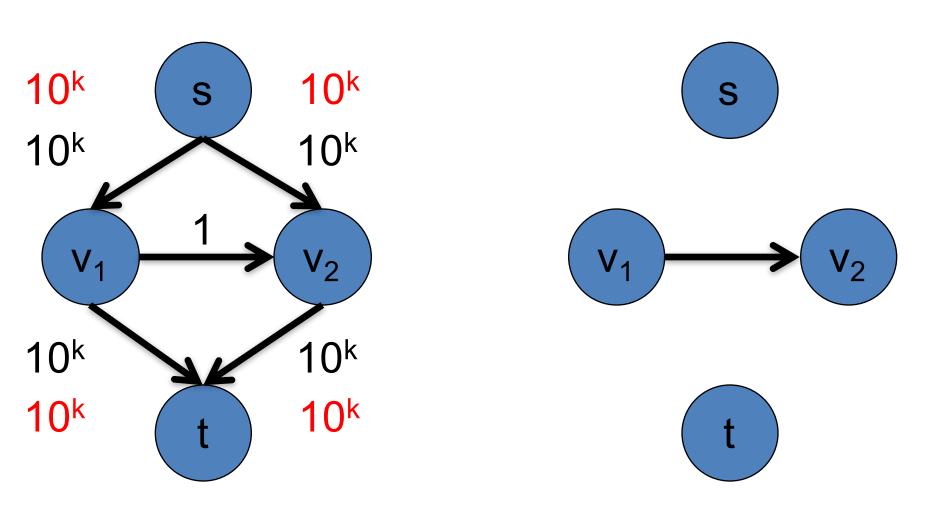
Pass the maximum allowable flow.



Pass the maximum allowable flow.



Update the residual graph.



No more s-t paths. Stop.

Solvers for the Minimum-Cut Problem

Augmenting Path and Push-Relabel

year	discoverer(s)	bound
1951	Dantzig	$O(n^2mU)$
1955	Ford & Fulkerson	$O(m^2U)$
1970	Dinitz	$O(n^2m)$
1972	Edmonds & Karp	$O(m^2 \log U)$
1973	Dinitz	$O(nm \log U)$
1974	Karzanov	$O(n^3)$
1977	Cherkassky	$O(n^2m^{1/2})$
1980	Galil & Naamad	$O(nm\log^2 n)$
1983	Sleator & Tarjan	$O(nm \log n)$
1986	Goldberg & Tarjan	$O(nm\log(n^2/m))$
1987	Ahuja & Orlin	$O(nm + n^2 \log U)$
1987	Ahuja et al.	$O(nm\log(n\sqrt{\log U}/m))$
1989	Cheriyan & Hagerup	$E(nm + n^2 \log^2 n)$
1990	Cheriyan et al.	$O(n^3/\log n)$
1990	Alon	$O(nm + n^{8/3} \log n)$
1992	King et al.	$O(nm + n^{2+\epsilon})$
1993	Phillips & Westbrook	$O(nm(\log_{m/n} n + \log^{2+\epsilon} n))$
1994	King et al.	$O(nm\log_{m/(n\log n)} n)$
1997	Goldberg & Rao	$O(m^{3/2}\log(n^2/m)\log U)$
		$O(n^{2/3}m\log(n^2/m)\log U)$

n: #nodes

m: #arcs

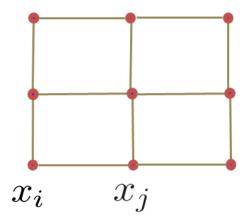
U: maximum

arc length

[Slide credit: Andrew Goldberg]

Max-Flow in Computer Vision

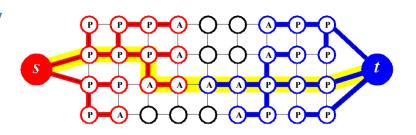
- Specialized algorithms for vision problems
 - Grid graphs
 - Low connectivity (m ~ O(n))



 Dual search tree augmenting path algorithm

[Boykov and Kolmogorov PAMI 2004]

- Finds approximate shortest augmenting paths efficiently
- High worst-case time complexity
- Empirically outperforms other algorithms on vision problems



[Slide credit: Pushmeet Kohli]

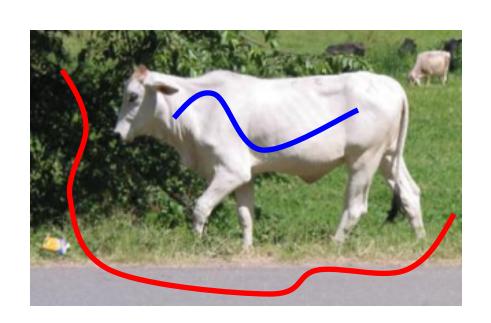
Outline

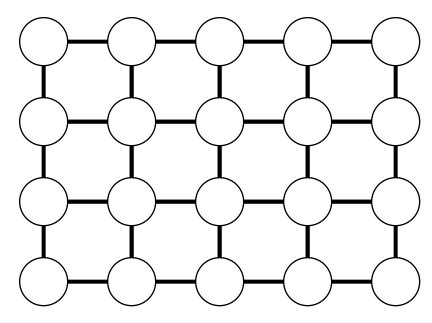
Preliminaries

Maximum Flow

Algorithms

- Energy minimization with max flow/min cut
 - Two-Label Energy Functions

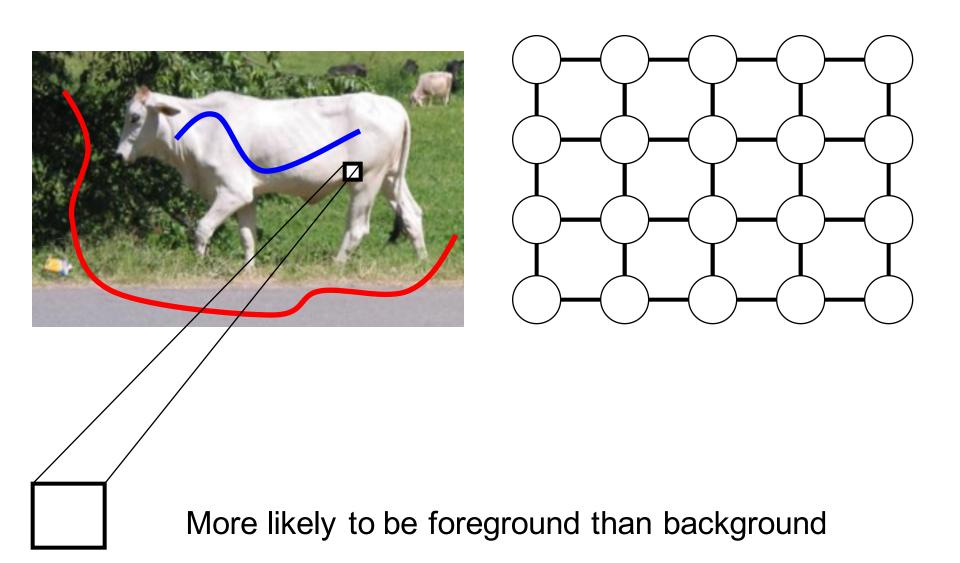


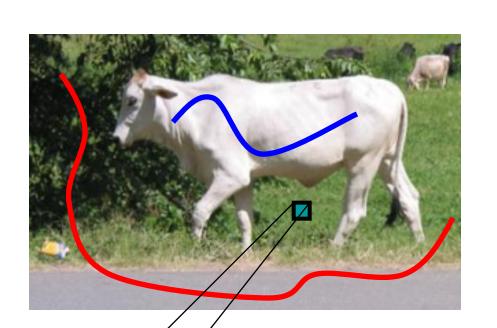


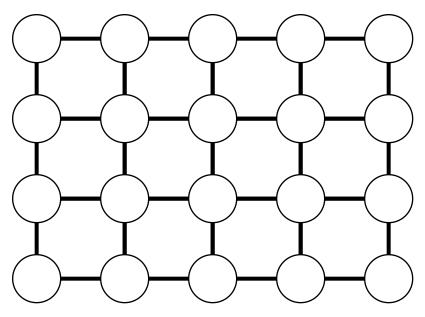
Foreground histogram of RGB values FG

Background histogram of RGB values BG

'1' indicates foreground and '0' indicates background



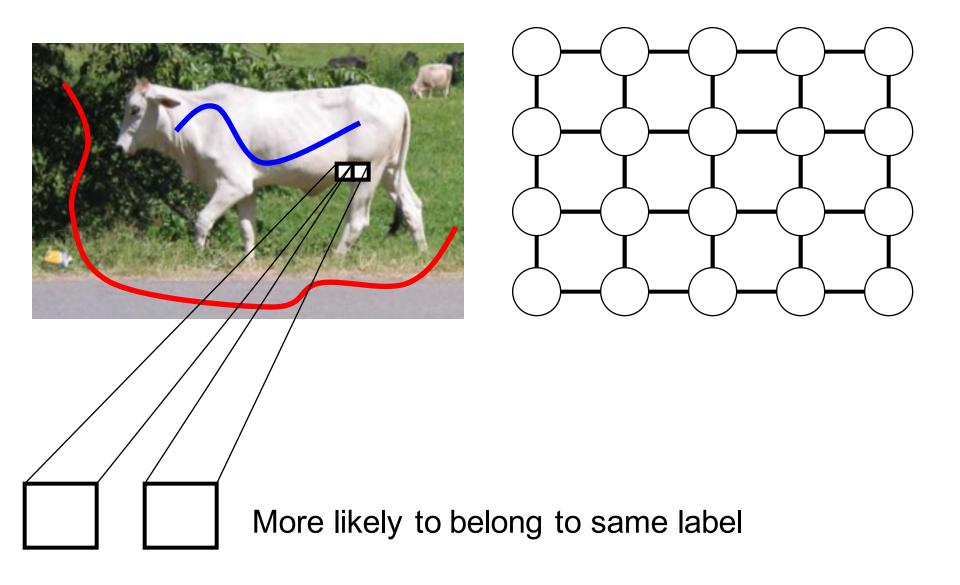


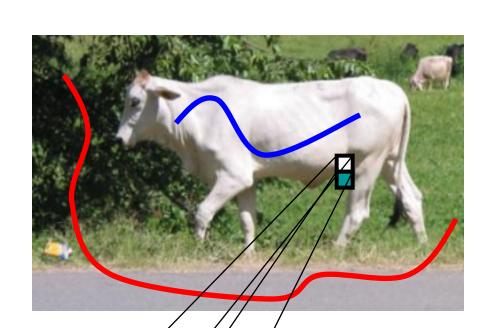


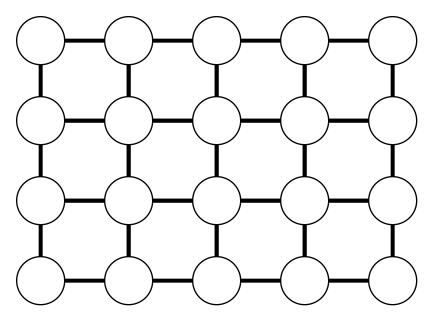
 $\theta_p(0)$ proportional to $-log(BG(d_a))$

 $\theta_p(1)$ proportional to $-\log(FG(d_a))$

More likely to be background than foreground







 $\theta_{pq}(i,k)$ proportional to $exp(-(d_a-d_b)^2)$ if $i \neq k$

$$\theta_{pq}(i,k) = 0$$
 if $i = k$

Less likely to belong to same label

Overview

Energy E

One vertex per random variable

+ Additional vertices "s" and "t"

Digraph

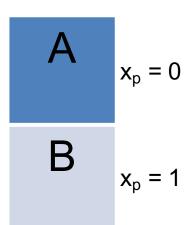
Compute Minimum Cut

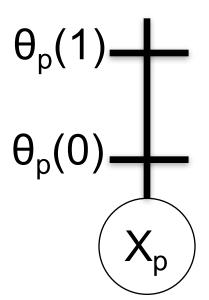
Labeling **x**

$$v_p \in U$$
 implies $x_p = 0$

$$v_p \in V\backslash U$$
 implies $x_p = 1$

U and V\U





A $x_p = 0$

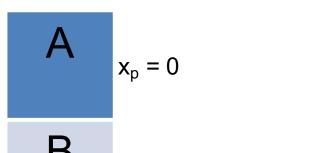
В

 $x_{p} = 1$

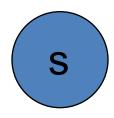
S

 V_p

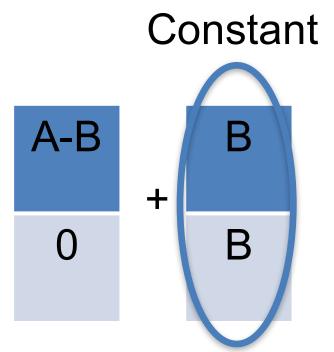
 $\left(\begin{array}{c}t\end{array}\right)$

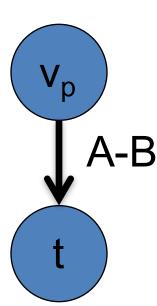


 $x_p = 1$



Let A ≥ B



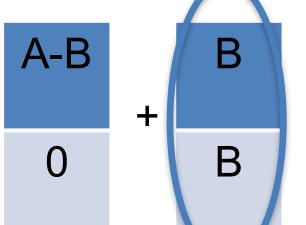




B

$$x_p = 1$$

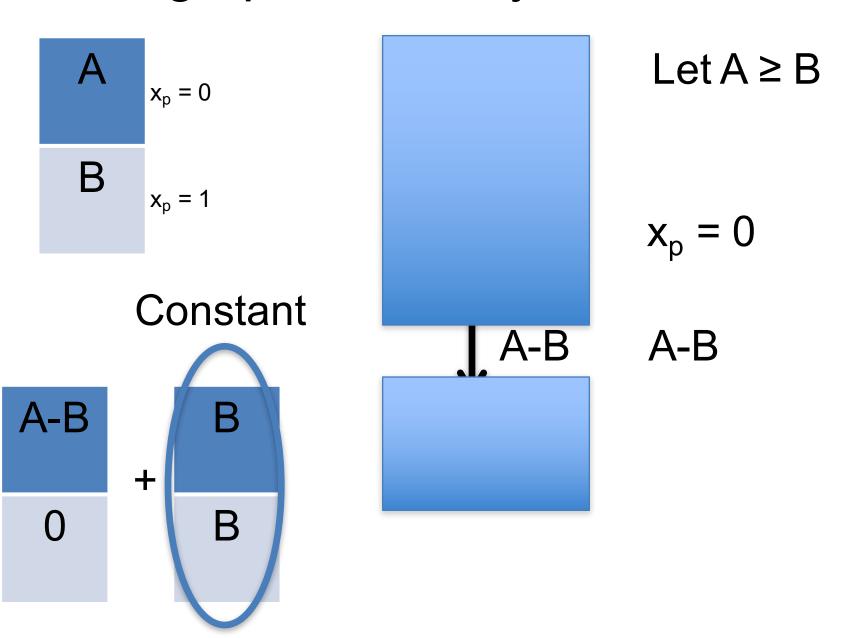
Constant

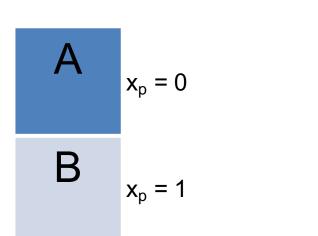




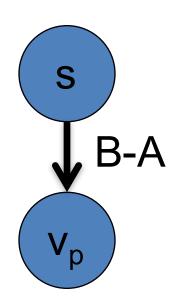
$$\chi_p = 1$$



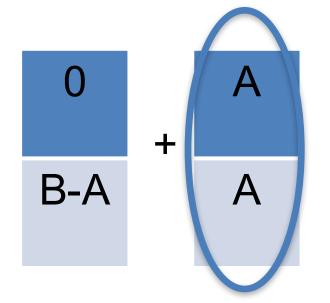


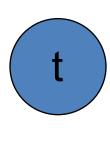


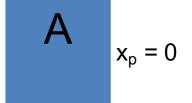
Constant



Let A < B

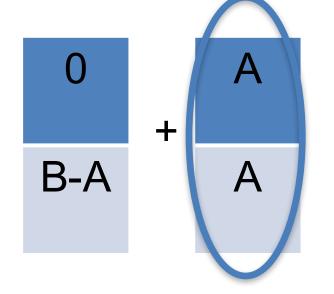


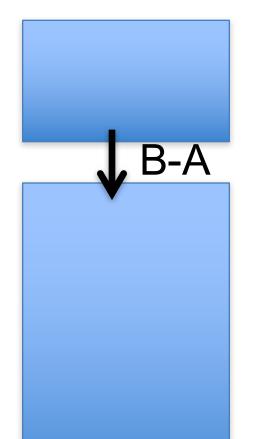




 $B = \begin{cases} x_p = 1 \end{cases}$

Constant





Let A < B

$$x_p = 1$$

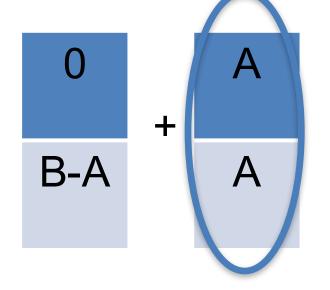
B-A

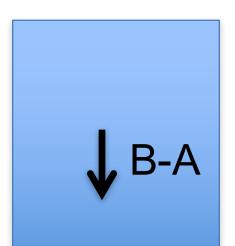
Digraph for Unary Potentials



 $x_{p} = 1$

Constant



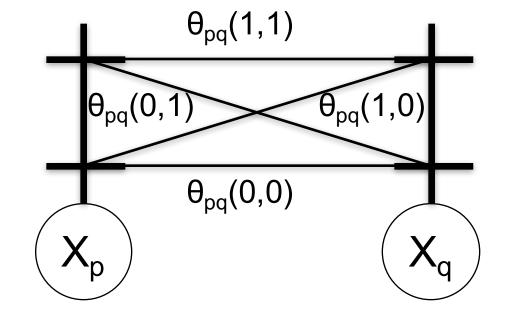


Let A < B

$$x_p = 0$$



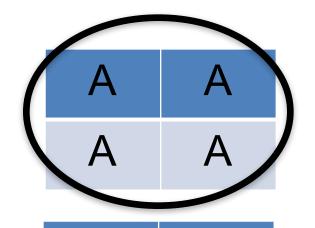
$$x_{p} = 0$$
 $x_{p} = 1$
 $x_{q} = 0$ A C
 $x_{q} = 1$ B D



A	Α
Α	Α

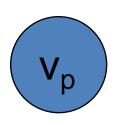
$$x_{p} = 0$$
 $x_{p} = 1$ $x_{q} = 0$ $x_{p} = 1$ $x_{q} = 0$ $x_{p} = 1$ $x_{q} = 1$





B-A

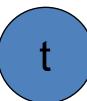
+

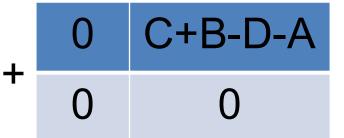


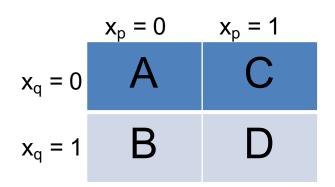
D-B



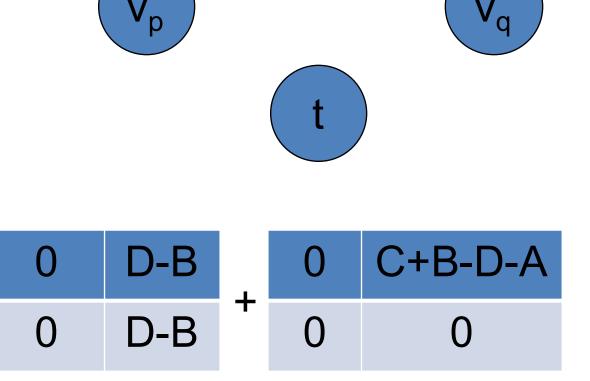
S	





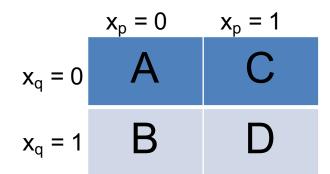


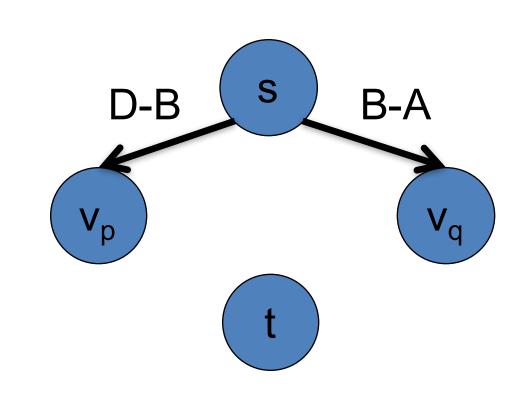
Unary Potential $x_{\alpha} = 1$



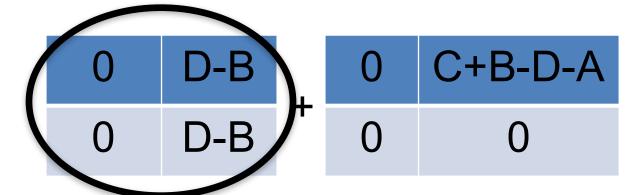
S

B-A

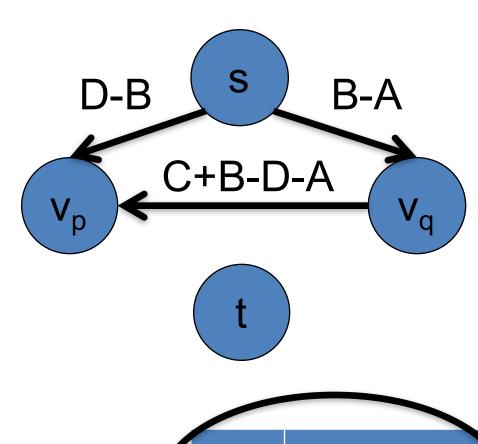




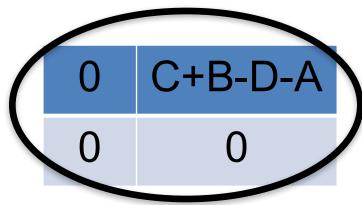
Unary Potential $x_p = 1$

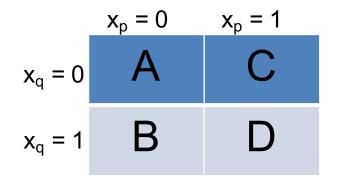


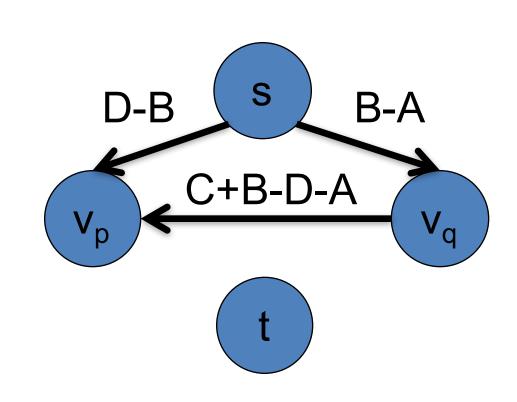
$$x_{p} = 0$$
 $x_{p} = 1$ $x_{q} = 0$ $x_{p} = 1$ $x_{q} = 0$ $x_{q} = 1$ $x_{q} = 1$



Pairwise Potential $x_p = 1$, $x_q = 0$







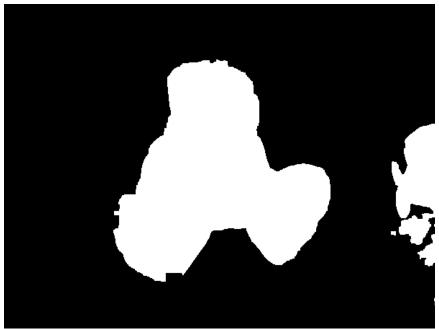
C+B-D-A ≥ 0

Submodular Energy

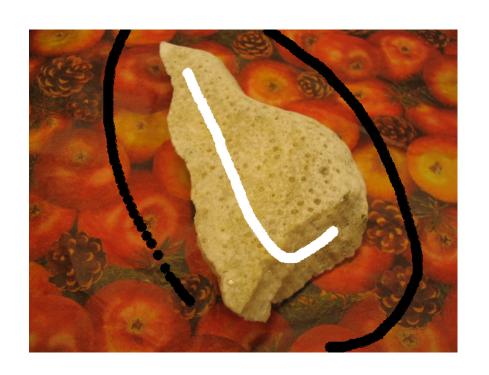
General 2-label MAP estimation is NP-hard

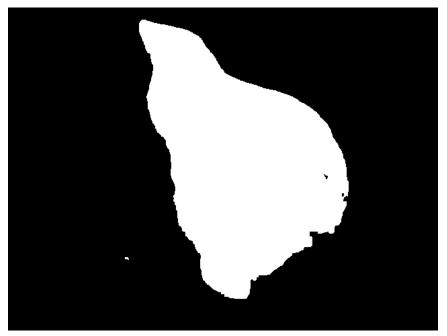
Results – Image Segmentation





Results – Image Segmentation



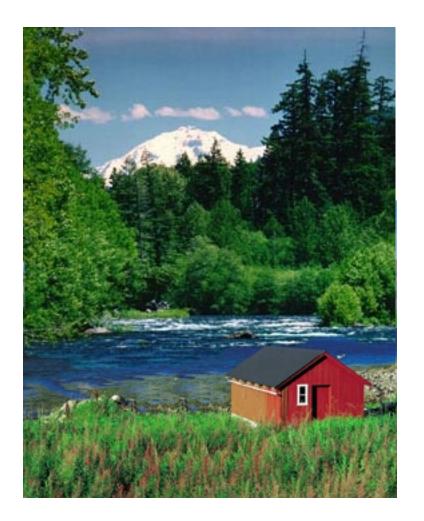


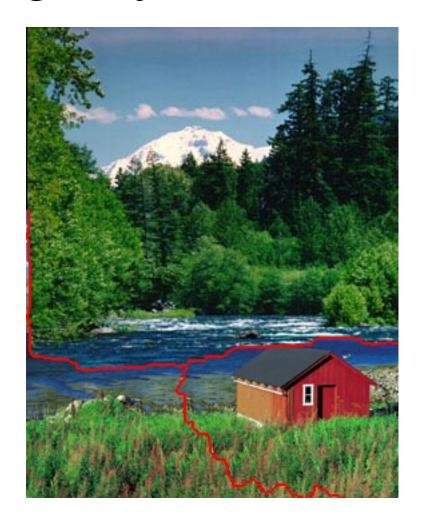
Results – Image Segmentation





Results – Image Synthesis





Results – Image Synthesis





Outline

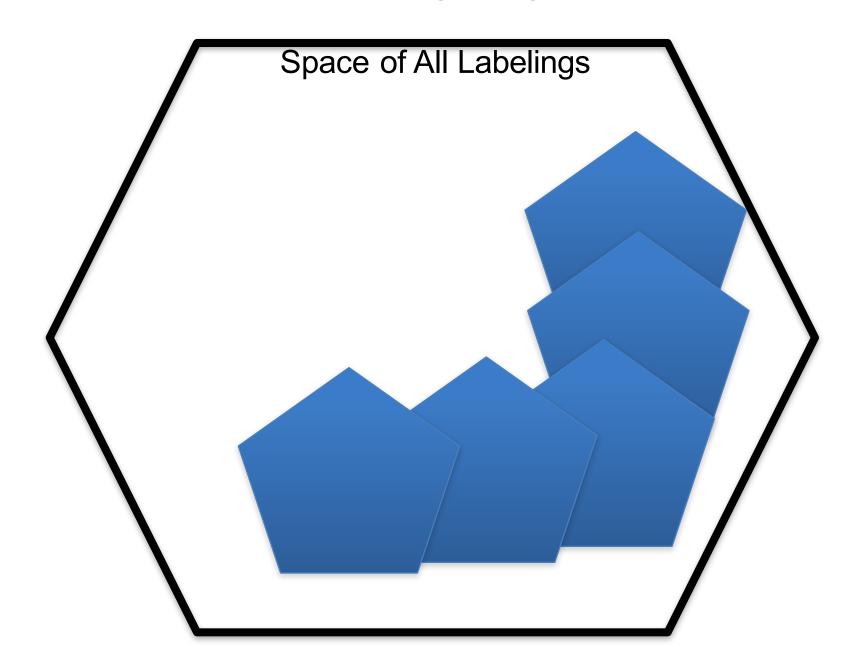
Preliminaries

Maximum Flow

Algorithms

- Energy minimization with max flow/min cut
 - Multi-Label Energy Functions

Move-Making Algorithms

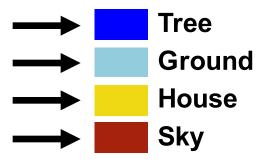


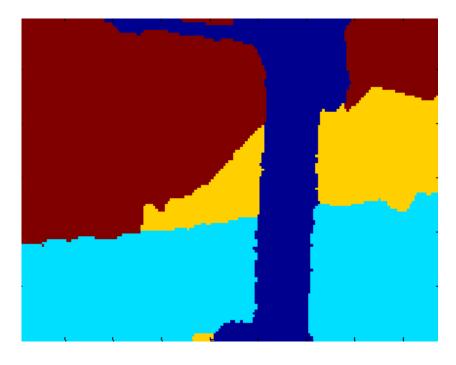
Variables take label I_{α} or retain current label

Variables take label I_{α} or retain current label

Status: Exiparide Skips Edee







Slide courtesy Pushmeet Kohli

Initialize labeling $\mathbf{x} = \mathbf{x}^0$ (say $x_p^0 = 0$, for all X_p)

For
$$\alpha$$
 = 1, 2, ..., h-1
$$\mathbf{x}^{\alpha} = \operatorname{argmin}_{\mathbf{x}'} E(\mathbf{x}')$$

$$s.t. \ \mathbf{x'}_{p} \in \{\mathbf{x}_{p}\} \ U \ \{\mathbf{I}_{\alpha}\}$$

$$Update \ \mathbf{x} = \mathbf{x}^{\alpha}$$
 End

Repeat until convergence

Restriction on pairwise potentials?

$$\theta_{pq}(i,k) + \theta_{pq}(\alpha,\alpha) \le \theta_{pq}(i,\alpha) + \theta_{pq}(\alpha,k)$$

Results - Denoising + Inpainting





Results - Denoising + Inpainting





Results – Denoising + Inpainting





Results - Denoising + Inpainting



