Level Set Methods in Imaging and Vision Applications

Selim Esedoglu

Level Set Method

- Represent the interface $\partial\Sigma \subset \boldsymbol{R}^n$ as the 0-level set of a function

$$\phi: \boldsymbol{R}^n \rightarrow \boldsymbol{R}$$

- Example:

$$\partial\Sigma = \{\ (x, y)\ :\ x^2 + y^2\ = r^2\ \} \subset \boldsymbol{R}^2$$

can be represented by

$$\phi(x, y) = r^2 - x^2 + y^2$$

or

$$\phi(x, y) = r - \sqrt{x^2 + y^2}$$

- $\Rightarrow$ Representation is not unique.

Level Set Method

Convention:

$$\Sigma = \{x : \phi(x) > 0\}$$

Basic Set Operations:

- Taking unions:

$$\{x : \phi_1(x) > 0\} \cup \{x : \phi_2(x) > 0\} = \{x : \max\{\phi_1(x), \phi_2(x)\} > 0\}$$

- Taking intersections:

$$\{x : \phi_1(x) > 0\} \cap \{x : \phi_2(x) > 0\} = \{x : \min\{\phi_1, \phi_2\} > 0\}$$

- Taking complements:

$$\left(\overline{\{x : \phi(x) > 0\}}\right)^c = \{x : -\phi(x) > 0\}$$

Level Set Method

- Indicator function:

$$\mathbf{1}_{\Sigma}(x) = H\big(\phi(x)\big)$$

- Unit outer normal:

$$N = -\frac{\nabla\phi}{|\nabla\phi|}$$

- Area of $\{x: \phi(x) > 0\}$:

$$\int_{\Omega} H(\phi)dx$$

- Perimeter of $\{x: \phi(x) > 0\}$:

$$\int_{\Omega} \left|\nabla\big(H(\phi)\big)\right| dx$$

Level Set Method

- Denote the unit normal and tangent vector fields as:

$$N(x) = \frac{\nabla\phi}{|\nabla\phi|} \quad \text{and} \quad T(x) = \frac{\nabla^{\perp}\phi}{|\nabla\phi|}$$

- Let $\gamma(s)$ be an arc-length parametrization of the 0-level set:

$$\{\gamma(s): s \in \boldsymbol{R}\} = \{x: \phi(x) = 0\}$$

- Then, curvature of the curve at $p = \gamma(0)$ is:

$$\frac{d}{ds} N\big(\gamma(s)\big)\Big|_{s=0} = \kappa(p) T(p)$$

- But we have:

$$\frac{d}{ds} N\big(\gamma(s)\big)\Big|_{s=0} = (DN)\Big|_{p} \dot{\gamma}(0)$$
$$= (DN)\Big|_{p} T(p)$$

Level Set Method

- Thus:

$$\kappa(p) = \left\langle (DN)\Big|_p T(p), T(p) \right\rangle$$

- Note that

$$\nabla \cdot N = \mathrm{trace}(DN)$$

$$= \langle (DN)N, N \rangle + \langle (DN)T, T \rangle$$

- But,

$$\langle (DN)N, N \rangle = \langle N, (DN)^T N \rangle = \frac{1}{2}\langle N, D|N|^2 \rangle = 0$$

- Hence,

$$\kappa = \nabla \cdot N = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)$$

Note: In n-dims,

$$\nabla \cdot N = (n-1)H$$

where H = Mean curvature.

Level Set Method

- Particularly useful for moving interfaces:

$$\text{Outward normal speed} = v_n$$

- Let $\gamma(s,t)$ be a parametrization of the 0-level set of $\phi(x,t)$:

$$\{\gamma(s,t): s \in \boldsymbol{R}\} = \{x: \phi(x,t) = 0\}$$

so that

$$\frac{\partial}{\partial t}\gamma \cdot N = -\frac{\partial}{\partial t}\gamma \cdot \frac{\nabla\phi}{|\nabla\phi|} = v_n$$

- Then, we have

$$\frac{d}{dt}\phi(\gamma(s,t),t) = 0 \quad \text{for all} \quad (s,t)$$

- We also have:

$$\frac{d}{dt}\phi(\gamma(s,t),t) = \nabla\phi \cdot \frac{\partial}{\partial t}\gamma + \phi_t = |\nabla\phi|\frac{\nabla\phi}{|\nabla\phi|} \cdot \frac{\partial}{\partial t}\gamma + \phi_t = 0$$

meaning that:

$$\boxed{\phi_t = |\nabla\phi| v_n}$$

Level Set Method

- Example: Motion by mean curvature.

$$\phi_t = |\nabla\phi| \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right)$$

- Discretization: Typically, expand the curvature term:

$$\kappa = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{|\nabla\phi|^3}$$

- Use centered differences, e.g.

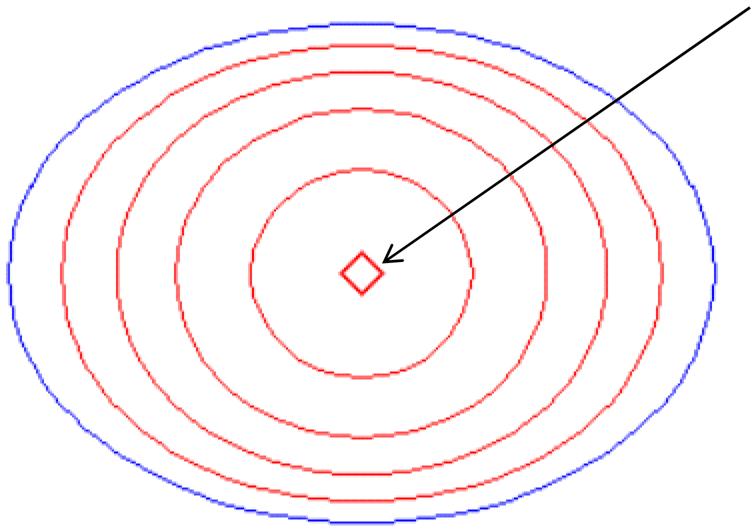
$$\phi_x \approx \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}$$

$$\phi_{xx} \approx \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2}$$

etc.

Level Set Method

- However, there can be issues:



Shrinks to a point, but the point never disappears!

Level Set Method

- Some difference quotients:

$$D_{x_j}^{+}\phi = \frac{\phi(x + he_j) - \phi(x)}{h}$$

$$D_{x_j}^{-}\phi = \frac{\phi(x) - \phi(x - he_j)}{h}$$

$$D_{x_j}^{c}\phi = \frac{\phi(x + he_j) - \phi(x - he_j)}{2h}$$

Level Set Method

- More reliable: Start with a discretization of

$$\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$$

- Arises as L^2 derivative of

$$\int |\nabla u| \, dx$$

- Start with a discretization of the energy:

$$\sum_{ij} h^2 \sqrt{\left(D_x^+ u_{ij}\right)^2 + \left(D_y^+ u_{ij}\right)^2 + \delta}$$

- Take variation of the discrete energy.

Level Set Method

- One gets:

$$\kappa \approx D_x^- \left(\frac{D_x^+ \phi^k}{\sqrt{(D_x^+ \phi)^2 + \left(D_y^+ \phi\right)^2 + \delta}} \right) + D_y^- \left(\frac{D_y^+ \phi^k}{\sqrt{(D_x^+ \phi)^2 + \left(D_y^+ \phi\right)^2 + \delta}} \right)$$

- Then:

$$\frac{\phi^{k+1} - \phi^k}{\delta t} =$$
$$|D^c \phi^k| \left\{ D_x^- \left(\frac{D_x^+ \phi^k}{\sqrt{(D_x^+ \phi)^2 + \left(D_y^+ \phi\right)^2 + \delta}} \right) + D_y^- \left(\frac{D_y^+ \phi^k}{\sqrt{(D_x^+ \phi)^2 + \left(D_y^+ \phi\right)^2 + \delta}} \right) \right\}$$

where

$$\left|D^c \phi^k\right| = \sqrt{(D_x^c \phi^k)^2 + \left(D_y^c \phi^k\right)^2 + \varepsilon}$$

Level Set Method

- For small enough δt, decreases discrete TV for sure.
- CFL condition:

$$\delta t \leq O(h^2)$$

- Analogue of convexity splitting: P. Smereka (2002):

$$\frac{\phi^{k+1} - \phi^k}{\delta t} = |\nabla \phi^k| \nabla \cdot \left(\frac{\nabla \phi^k}{|\nabla \phi^k|} \right) + \Delta \phi^{k+1} - \Delta \phi^k.$$

- At every time step, solve

$$\phi^{k+1} = \phi^k + (\delta t)(I - (\delta t)\Delta)^{-1} |\nabla \phi^k| \nabla \cdot \left(\frac{\nabla \phi^k}{|\nabla \phi^k|} \right)$$

smoothing operator

- Appears to be unconditionally stable.

A Word on Flow Networks

and Their Application to Segmentation:

Some basic notions.

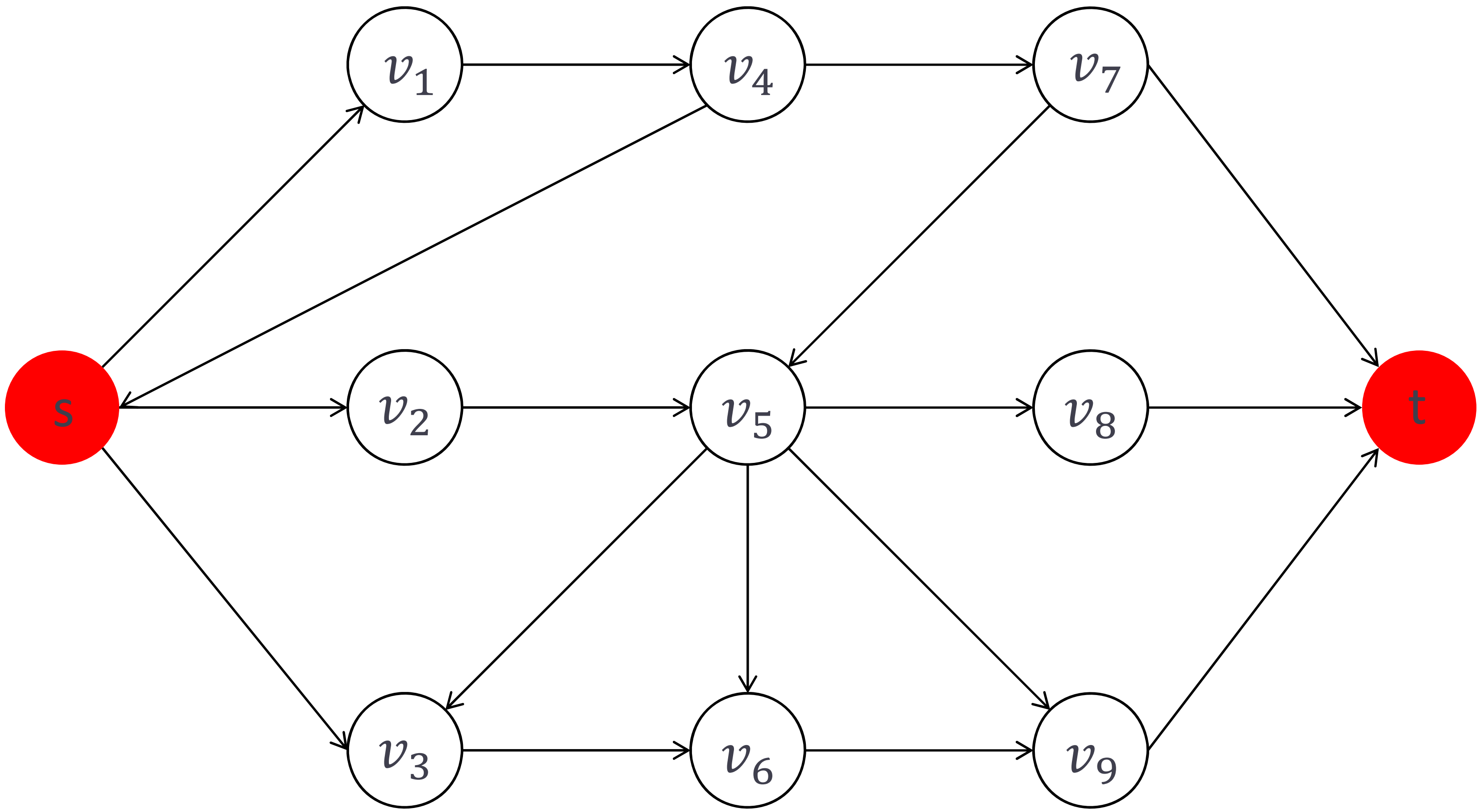
Flow Networks

- G=(V,E) is a directed graph.
  - V = Vertices of the graph.
  - E = Edges of the graph:

$$E \subset V \times V$$

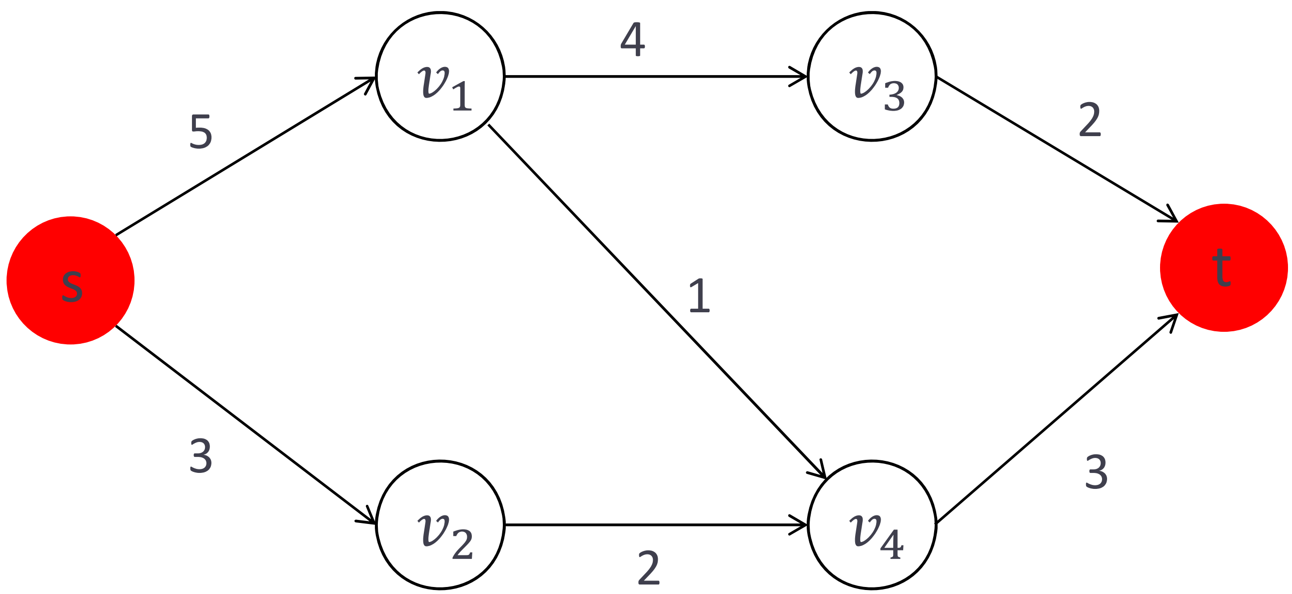
- If $(u, v) \in E$, then $(v, u) \notin E$.
- There are two distinguished vertices:
  - Source: s
  - Sink: t
- Each $v \in V$ lies on a path from s to t ($\Rightarrow$ G is connected).
- No loops: $(u, u) \notin E$ for any $u \in V$.

Flow Networks



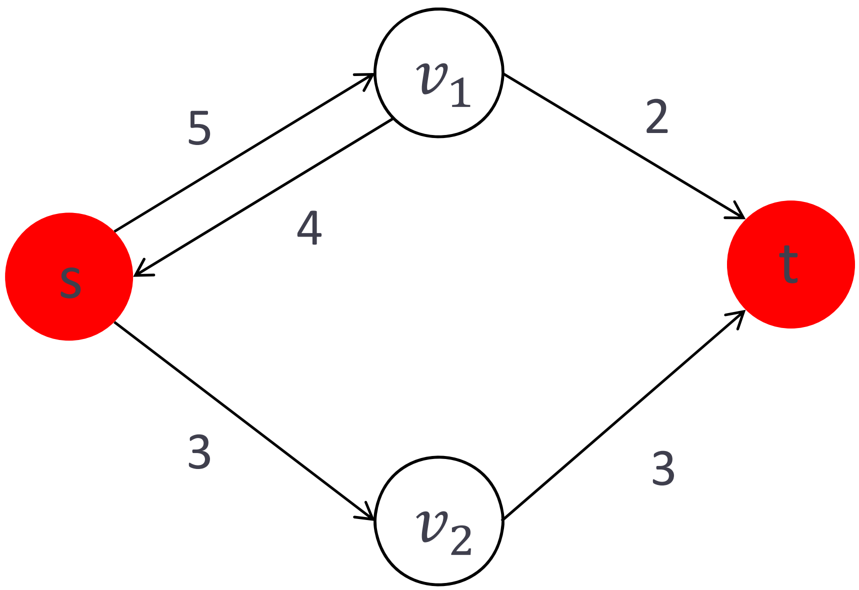
Flow Networks

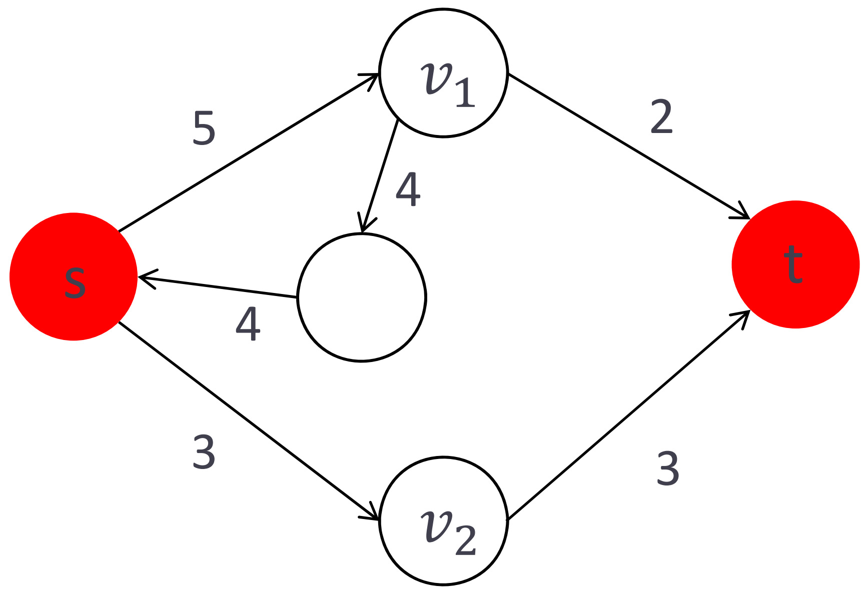
- Each edge $(u, v) \in E$ is assigned a non-negative capacity $c(u, v)$:
$$c: E \rightarrow \mathbb{R}^+$$
- Extend c to all pairs $(u, v) \in V \times V$:
$$c(u, v) = 0 \text{ if } (u, v) \notin E.$$



Flow Networks

- The restriction
$$(u, v) \in E \;\Rightarrow\; (v, u) \notin E$$
can sometimes be alleviated:





Flow Networks

- A flow on G: A real valued function

$$f: V \times V \to \mathbb{R}$$

with the interpretation

$$f(u, v) = \text{Flow from vertex } u \text{ to vertex } v$$

conforming to the following constraints:

1. Capacity constraint: For all $u, v \in V$,

$$0 \leq f(u, v) \leq c(u, v)$$

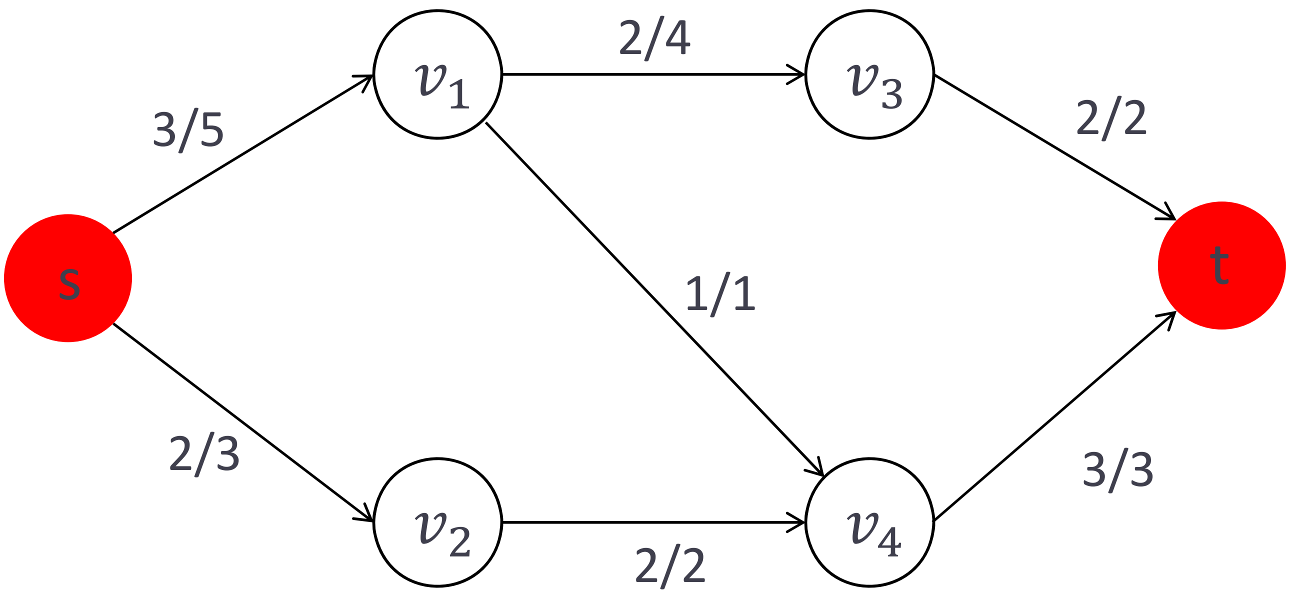
2. Flow conservation:

$$\underbrace{\sum_{v \in V} f(v, u)}_{\text{Total flow into u}} = \underbrace{\sum_{w \in V} f(u, w)}_{\text{Total flow out of u}}$$

(Note: If $(u, v) \notin E$, then $f(u, v) = 0$).

Flow Networks

Example of a valid flow. Edges (1,4), (3,t), (2,4), and (4,t) are saturated.



Flow Networks

- Value of a flow:

$$|f| = \sum_{v \in V} f(s,v) - \sum_{w \in V} f(w,s)$$

Total flow out of source

Total flow into source

- Max flow problem: Given the network $G = (E, V)$ and capacity function c, find the flow f on G such that

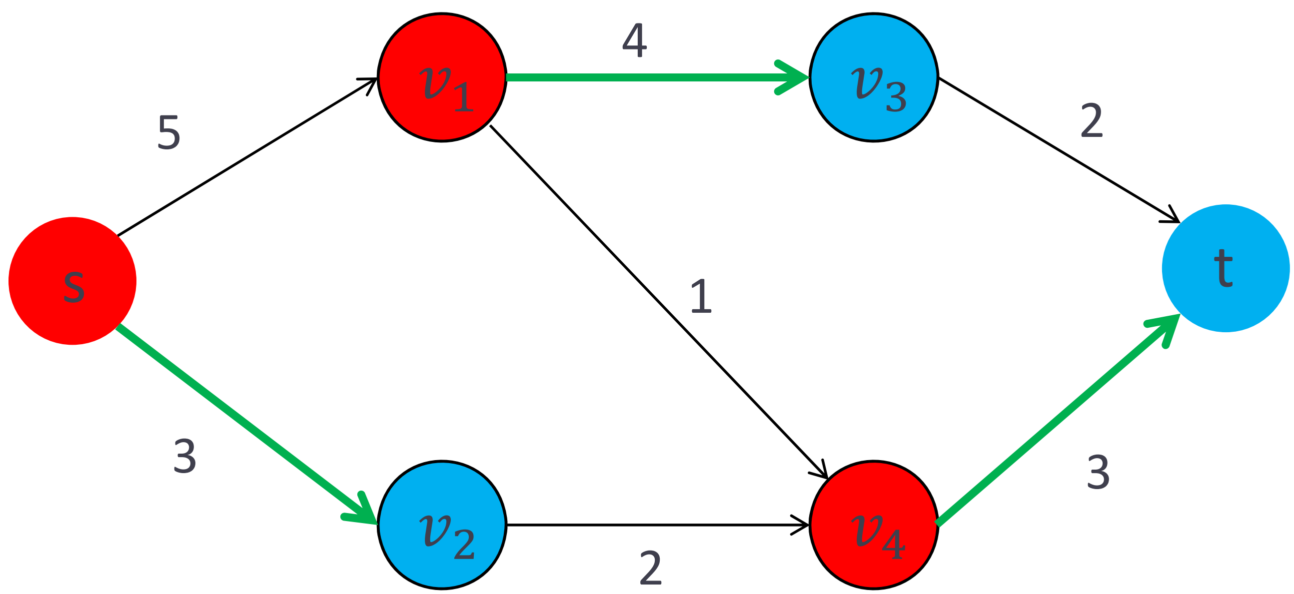
$$|f| \text{ is maximum.}$$

Cuts of Flow Networks

- A cut (S,T) of a flow network G=(V,E) is a partition of V into sets
$$S \subset V \text{ and } T = V \setminus S$$
such that:
$$s \in S \text{ and } t \in T.$$
- Capacity of a cut (S,T) is
$$c(S,T) = \sum_{u \in S, v \in T} c(u,v)$$
- Note: Only edges from S to T are counted.

- Minimum cut problem: Given the network $G = (V, E)$ and capacity function c, find a cut (S, T) of G such that:
$$c(S,T) \text{ is minimum.}$$

Cuts of Flow Networks



c(S,T)=4+3+3=10.

Cuts of Flow Networks

- Net flow $f(S,T)$ across a cut (S,T) is

$$f(S,T) = \sum_{u \in S, v \in T} f(u,v) - \sum_{u \in S, v \in T} f(v,u)$$
$$= (\text{Flow from } S \text{ } to \text{ } T)$$
$$-(\text{Flow from } T \text{ } to \text{ } S).$$

- Claim: Let f be a flow in a flow network G, and let (S,T) be a cut of G. Then:

$$f(S,T) = |f|$$

i.e. net flow across any cut is the same.

Cuts of Flow Networks

- Proof:

$$f(S,T) = \sum_{u \in S, v \in T} f(u,v) - \sum_{u \in S, v \in T} f(v,u)$$

We have:

$$\sum_{u \in S, v \in T} f(u,v) = \sum_{v \in T} f(s,v) + \boxed{\sum_{\substack{u \in S \setminus \{s\} \\ v \in T}} f(u,v)}$$

and

$$\sum_{u \in S, v \in T} f(v,u) = \sum_{v \in T} f(v,s) + \boxed{\sum_{\substack{u \in S \setminus \{s\} \\ v \in T}} f(v,u)}$$

Cuts of Flow Networks

For each $u \in S \setminus \{s\}$, we have

$$\sum_{v \in V} f(u,v) = \sum_{w \in V} f(w,u)$$

by the flow conservation constraint. Summing over $u \in S \setminus \{s\}$,

$$\sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(u,v) = \sum_{u \in S \setminus \{s\}} \sum_{w \in V} f(w,u)$$

Split the inner sums using $V = S \cup T = (S \setminus \{s\}) \cup \{s\} \cup T$:

$$\sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(u,v) = \sum_{u \in S \setminus \{s\}} f(u,s) + \boxed{\sum_{u \in S \setminus \{s\}} \sum_{v \in S \setminus \{s\}} f(u,v)} + \sum_{u \in S \setminus \{s\}} \sum_{v \in T} f(u,v)$$

$$\sum_{u \in S \setminus \{s\}} \sum_{w \in V} f(w,u) = \sum_{u \in S \setminus \{s\}} f(s,u) + \boxed{\sum_{u \in S \setminus \{s\}} \sum_{w \in S \setminus \{s\}} f(w,u)} + \sum_{u \in S \setminus \{s\}} \sum_{w \in T} f(w,u)$$

these are =

Cuts of Flow Networks

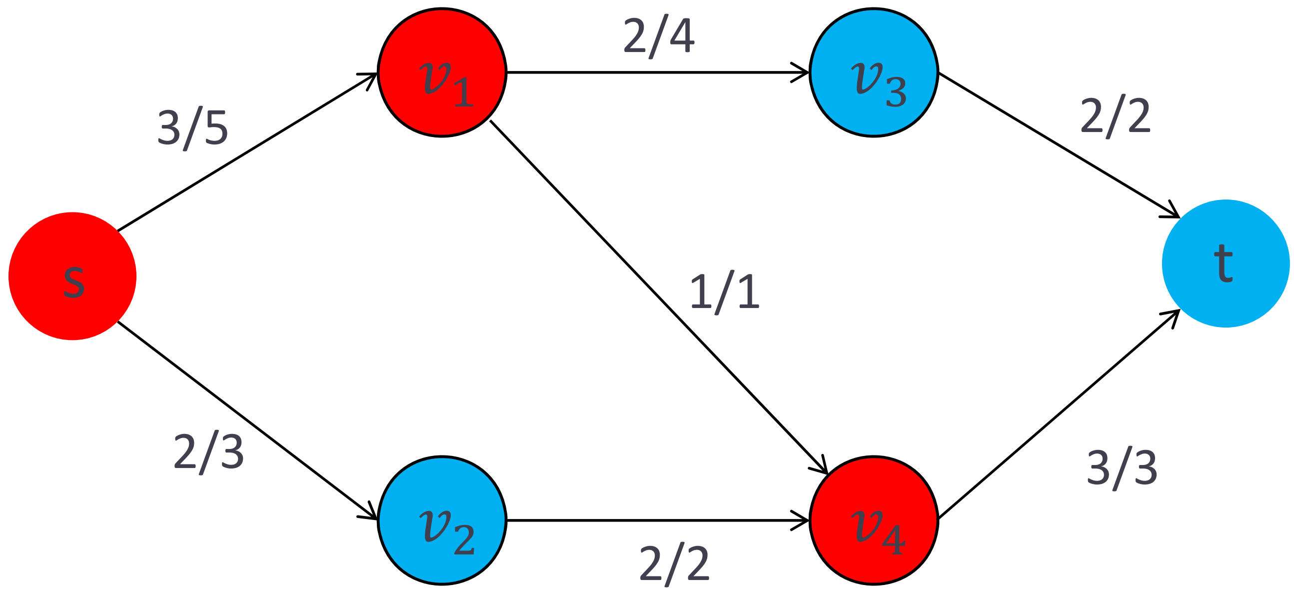
We get:

$$\sum_{\substack{u \in S \setminus \{s\} \\ v \in T}} f(u,v) + \sum_{u \in S \setminus \{s\}} f(u,s) = \sum_{\substack{u \in S \setminus \{s\} \\ w \in T}} f(w,u) + \sum_{u \in S \setminus \{s\}} f(s,u)$$

Combining:

$$\begin{aligned} f(S,T) &= \sum_{v \in T} f(s,v) + \sum_{u \in S \setminus \{s\}} f(s,u) \\ &\quad - \sum_{v \in T} f(v,s) - \sum_{u \in S \setminus \{s\}} f(u,s) \\ &= \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) \\ &= |f|. \end{aligned}$$

Cuts of Flow Networks



f(S,T)=2+2+3-2=5

Cuts of Flow Networks

- Corollary: Let f be a flow on the network G=(V,E). Then:

$$|f| = \sum_{v \in V} f(v,t) - \sum_{v \in V} f(t,v)$$

  In words,

$$|f| = \text{Net flow into sink.}$$

- Proof: Take

$$S = V \setminus \{t\}$$
$$T = \{t\}.$$

- Thus, we see that max flow problem is:

Find a flow such that net flow into sink is maximal.

Cuts of Flow Networks

- Claim: Given a flow network G, any flow f on G, and any cut (S,T) of G, we have:

  $$|f| \leq c(S,T)$$

  In particular,

  $$\text{Max flow } \leq \text{Min cut.}$$

- Proof:

$$|f| = f(S,T) = \sum_{u \in S, v \in T} f(u,v) - \sum_{u \in S, v \in T} f(v,u)$$

$$\leq \sum_{u \in S, v \in T} f(u,v)$$

$$\leq \sum_{u \in S, v \in T} c(u,v) = c(S,T).$$

Note: “=“ can be attained only if there is no flow from T to S, and all edges from S to T are saturated.

Cuts of Flow Networks

- Theorem: Let G=(V,E) be a flow network. A flow f on G is a maximal flow iff there exists a cut (S,T) of G such that
$$|f| = c(S,T).$$
In particular,
$$\text{Max. flow} = \text{Min. cut}$$

- Proof: Let f be a maximal flow on G. Suppose there is no cut (S,T) of G for which
$$|f| = c(S,T).$$
That means: For any cut (S,T) of G we have:

1. For some $u \in S$ and $v \in T$ we have $f(u,v) < c(u,v)$; OR
2. For some $u \in S$ and $v \in T$ we have $f(v,u) > 0$.

Start with:
$$T_0 = \{t\}.$$

Cuts of Flow Networks

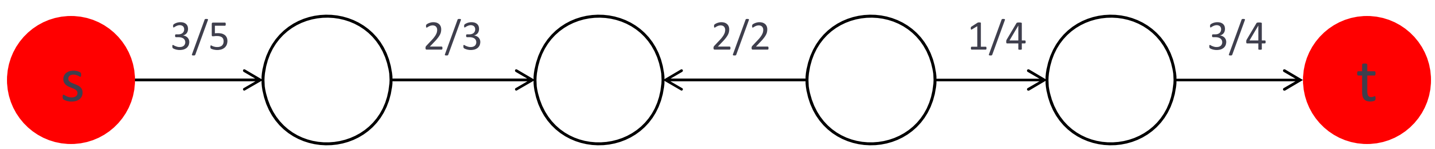
- Define T_{k+1} in terms of T_k recursively as:

$$T_{k+1} = T_k \cup \{u \in S_k : \exists v \in T_k \text{ with } f(u,v) < c(u,v)\}$$
$$\cup \{u \in S_k : \exists v \in T_k \text{ with } f(v,u) > 0\}.$$

In words: T_{k+1} is obtained from T_k by adding vertices $u \in S_k$ s.t.

1. Either $(u, v) \in E$ form some $v \in T_k$ is unsaturated,
2. Or $(v, u) \in E$ for some $v \in T_k$ carries flow out of T_k.

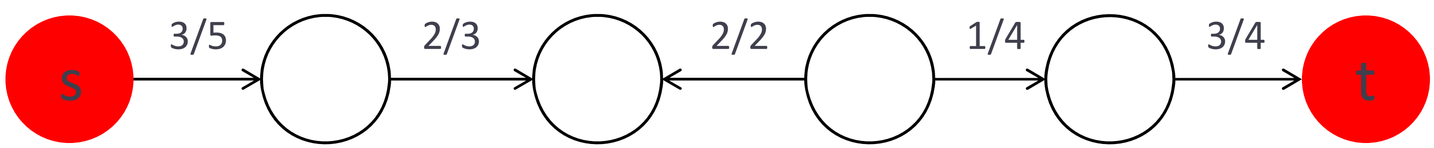
- At some point, we will have $s \in T_n$.
- That means, ∃ a path p on G from s to t of the form:



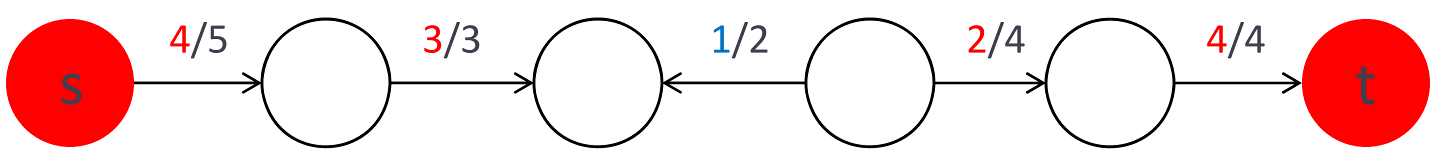
Cuts of Flow Networks

Properties of the path:

- Edges pointing towards t: Unsaturated.
- Edges pointing towards s: Non-zero flow.



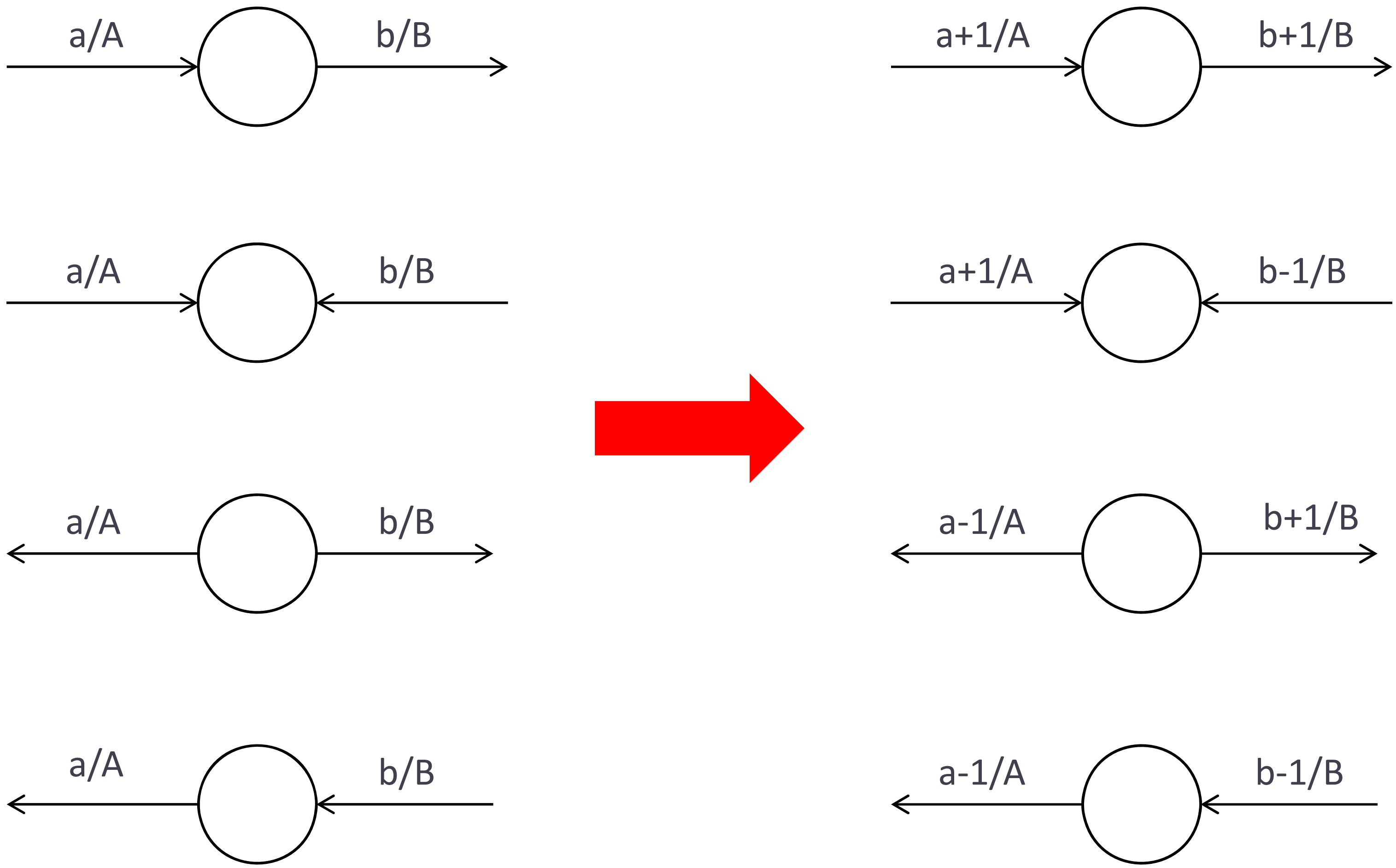
- Flow can be modified along such a path to increase flow into the sink:



- Both capacity constraint and flow conservation are maintained.

Cuts of Flow Networks

- In general, we encounter four types of nodes along such a path:



- Increase flow towards t in arrow on right; adjust arrow on left.

Flow Network

- We thus see that there has to exist a cut (S, T) of G such that
$$|f| = c(S, T).$$
- But then, since
$$|f| = f(S, T) \leq c(S, T)$$
for any flow f and cut (S, T), we conclude:
    - (S, T) is a minimal cut, and
    - Min cut = Max flow.

- Conversely, if there exists a cut (S, T) of G such that
$$|f| = c(S, T)$$
it immediately follows that f is a maximal flow.

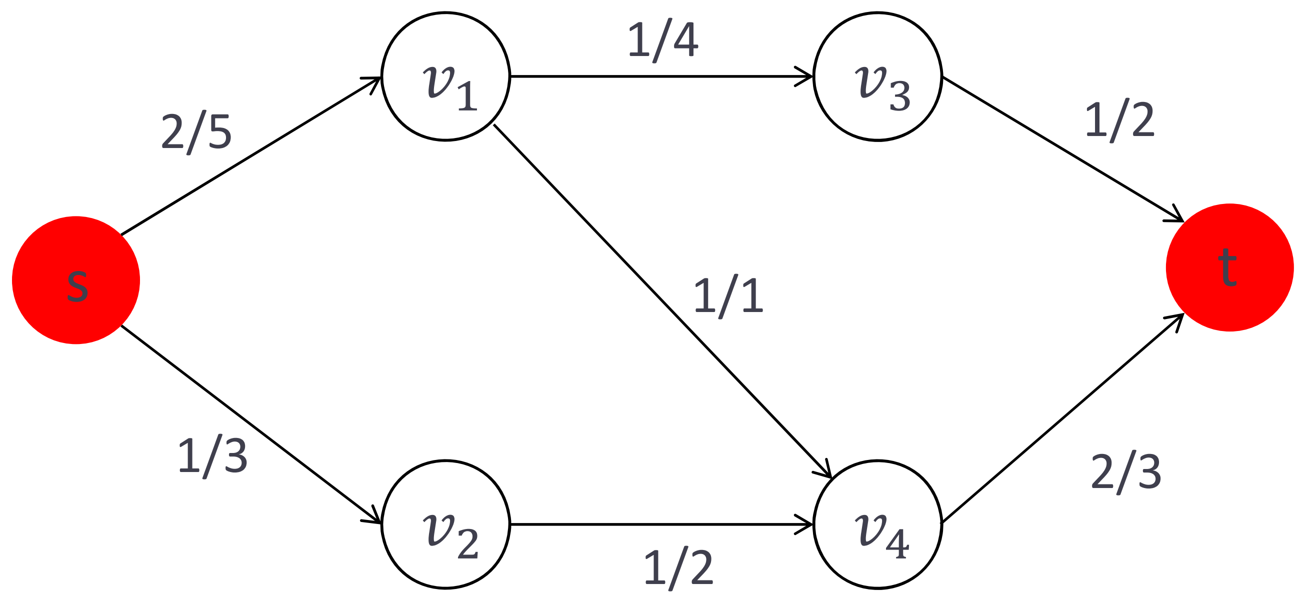
Max Flow Algorithms

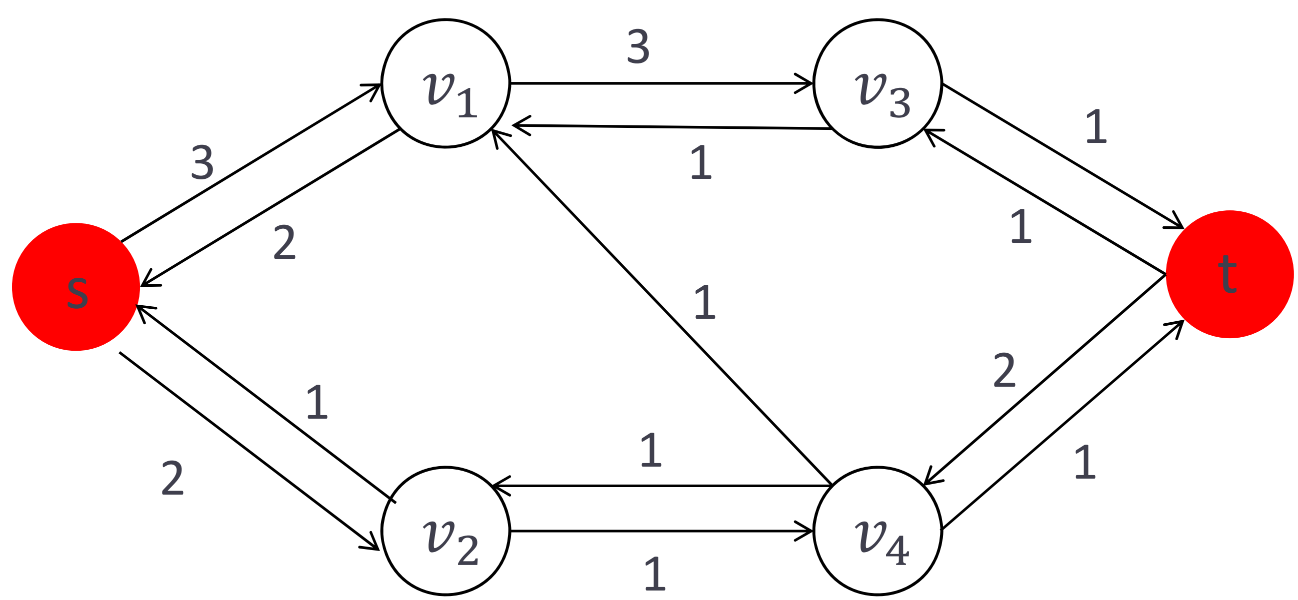
- There are many.
- We focus on: Ford & Fulkerson algorithm.
- IDEA: Iteratively improve the flow by:
  - Constructing the residual network for the flow,
  - Finding an augmenting path on the residual network,
  - Flow augmentation along the augmenting path.

Ford-Fulkerson Algorithm

- Residual network G_f corresponding to a flow f on the network $G = (V, E)$ is constructed as follows:
- Write $G_f = (V_f, E_f)$.
- $V_f = V$.
- Edges:
  - If $(u, v) \in E$ and $f(u, v) < c(u, v)$, then $(u, v) \in E_f$ and
    $$c_f(u, v) = c(u, v) - f(u, v).$$
  - If $(u, v) \in E$ and $f(u, v) > 0$, then $(v, u) \in E_f$ and
    $$c_f(v, u) = f(u, v).$$

Ford-Fulkerson Algorithm

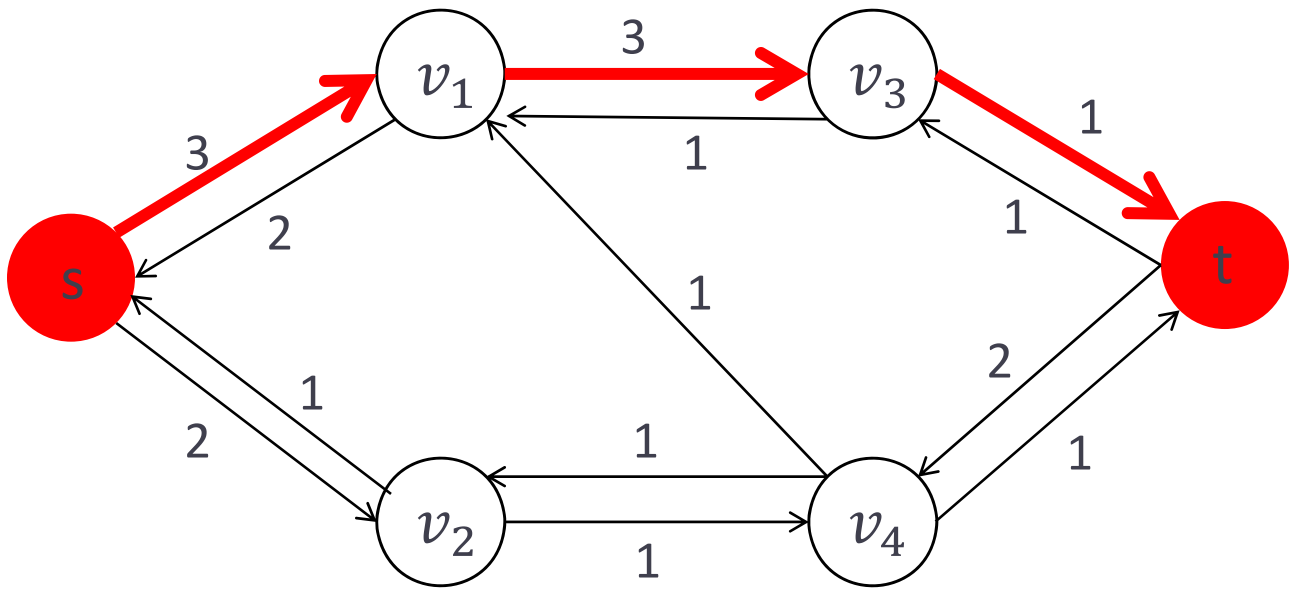




Ford-Fulkerson Algorithm

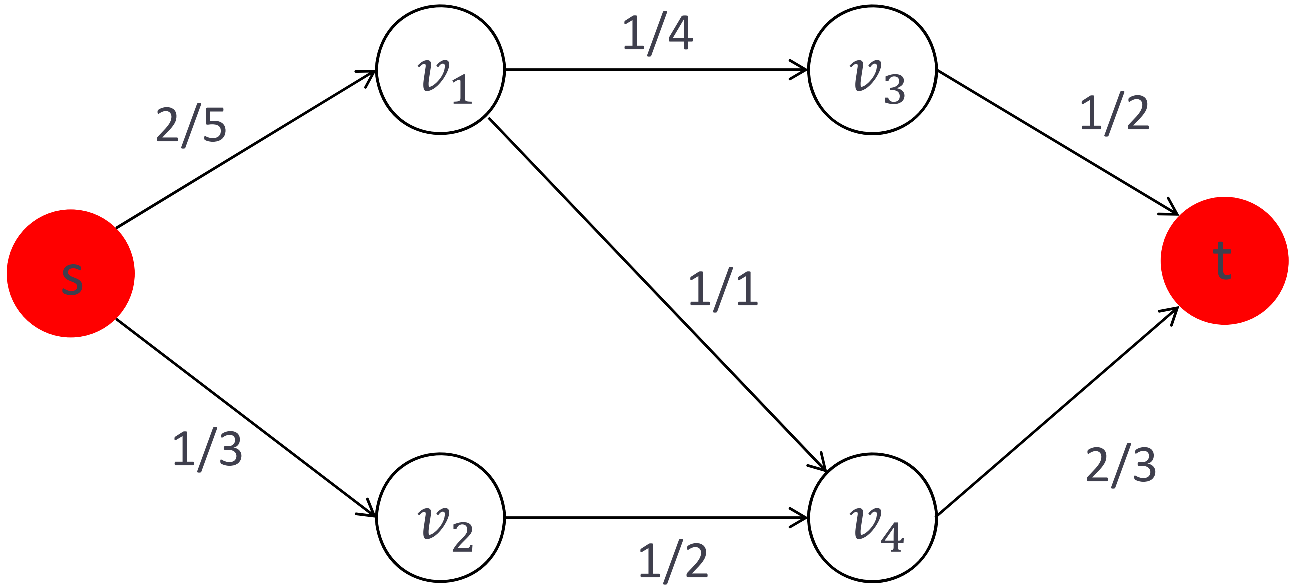
Find an **augmenting path** from s to t in the residual network G_f:

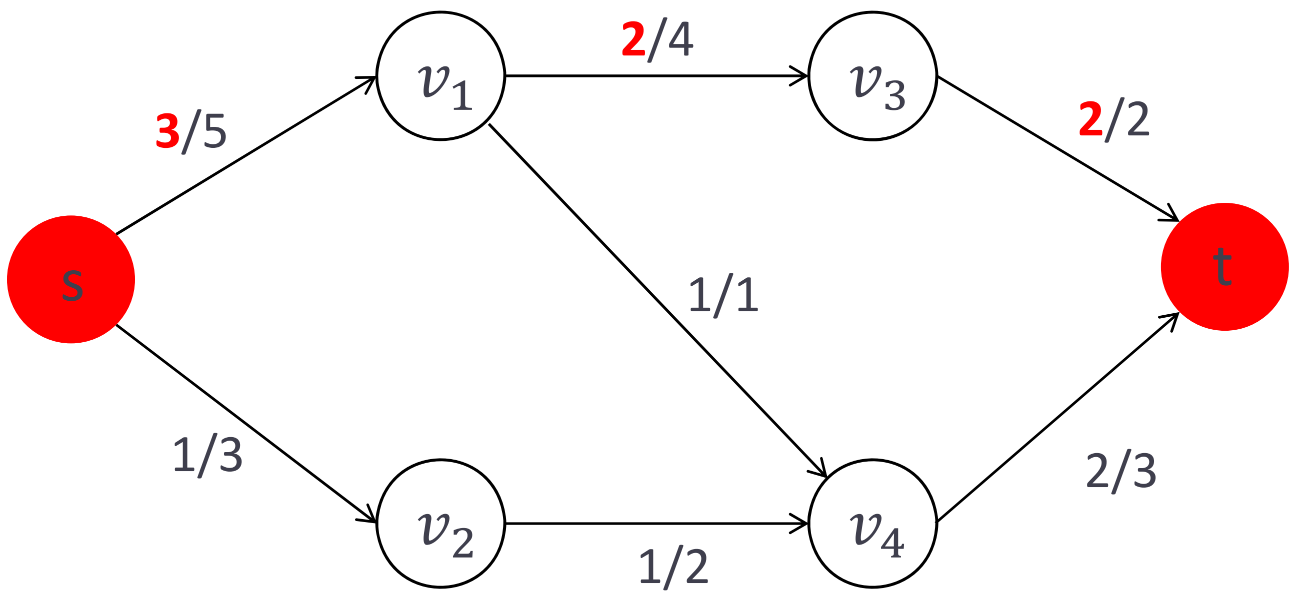
A simple path from **s** to **t** in G_f.



Existing flow can now be augmented along this path by 1.

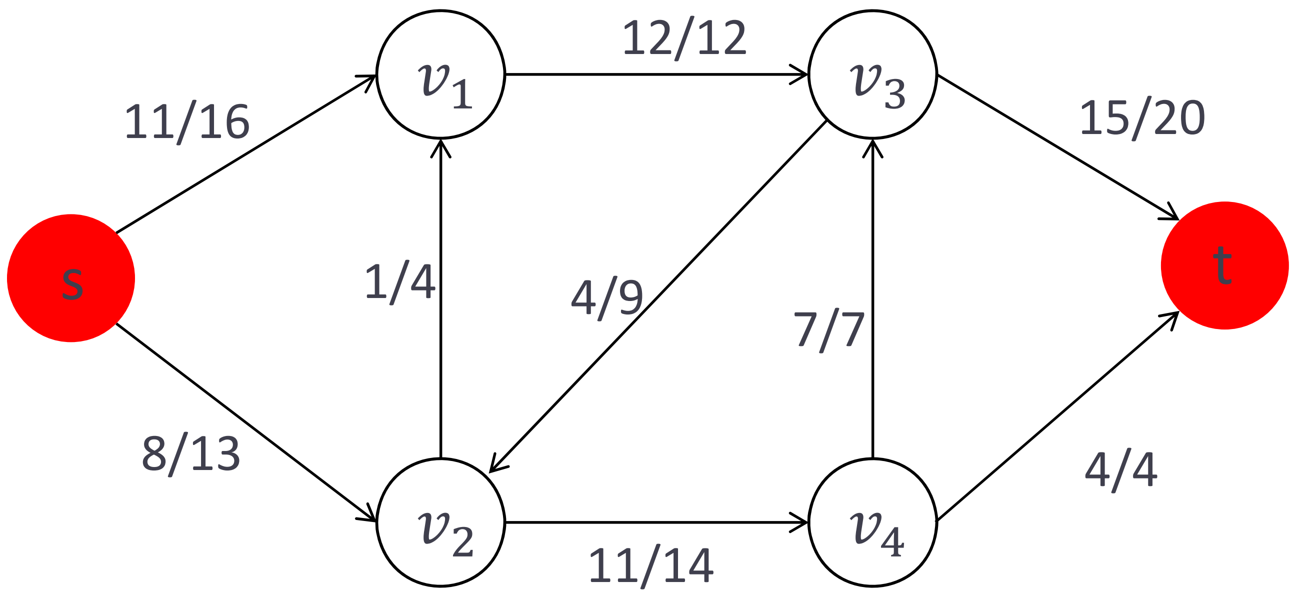
Ford-Fulkerson Algorithm



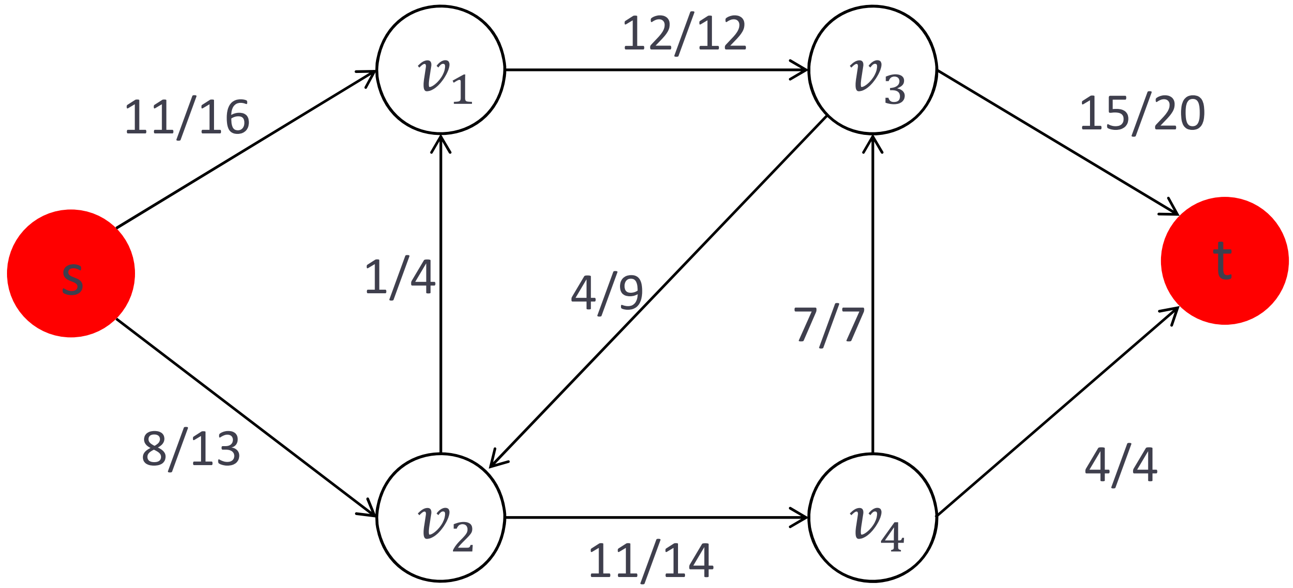


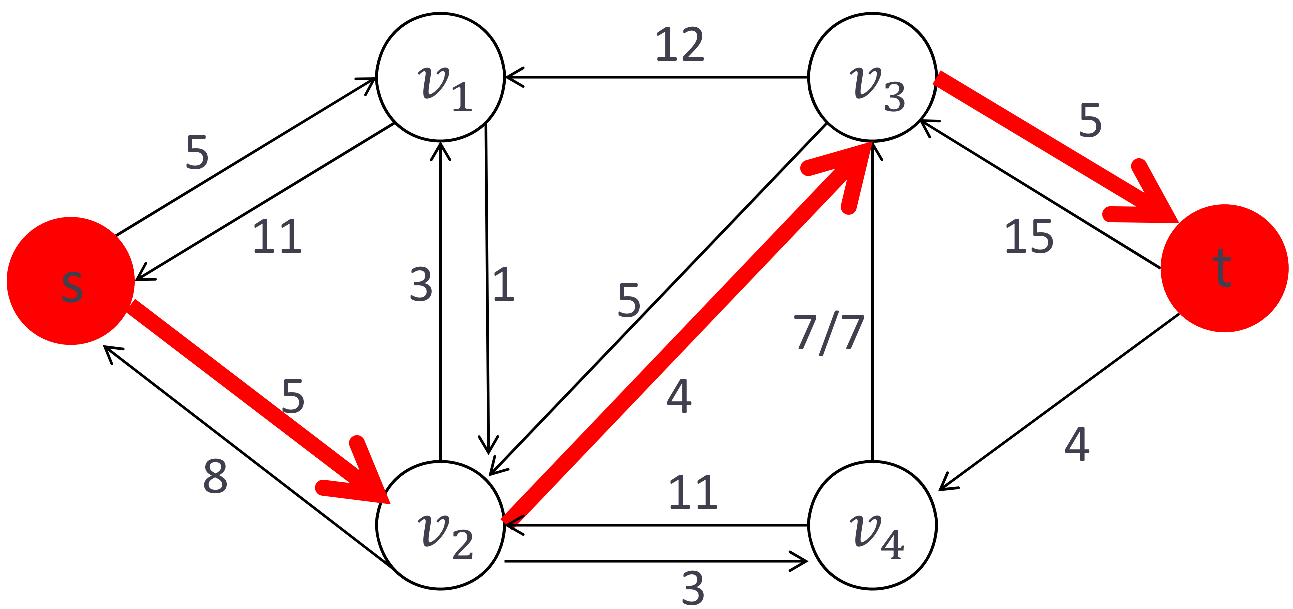
Ford-Fulkerson Algorithm

- Note that it may be necessary to cross the reversed edges:

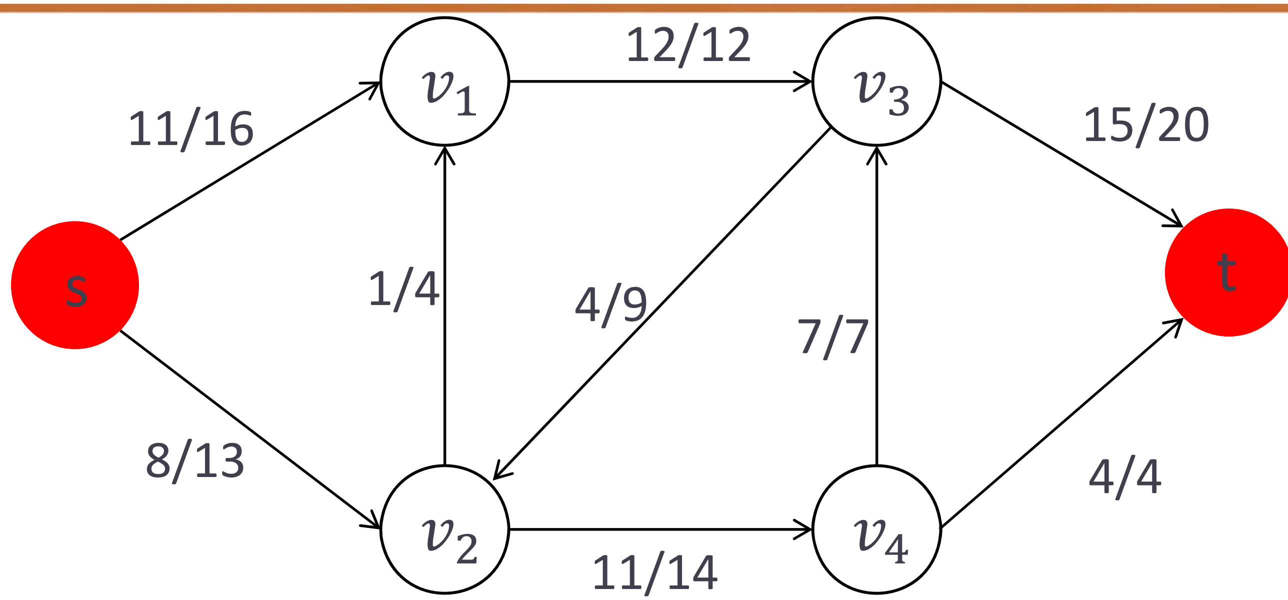


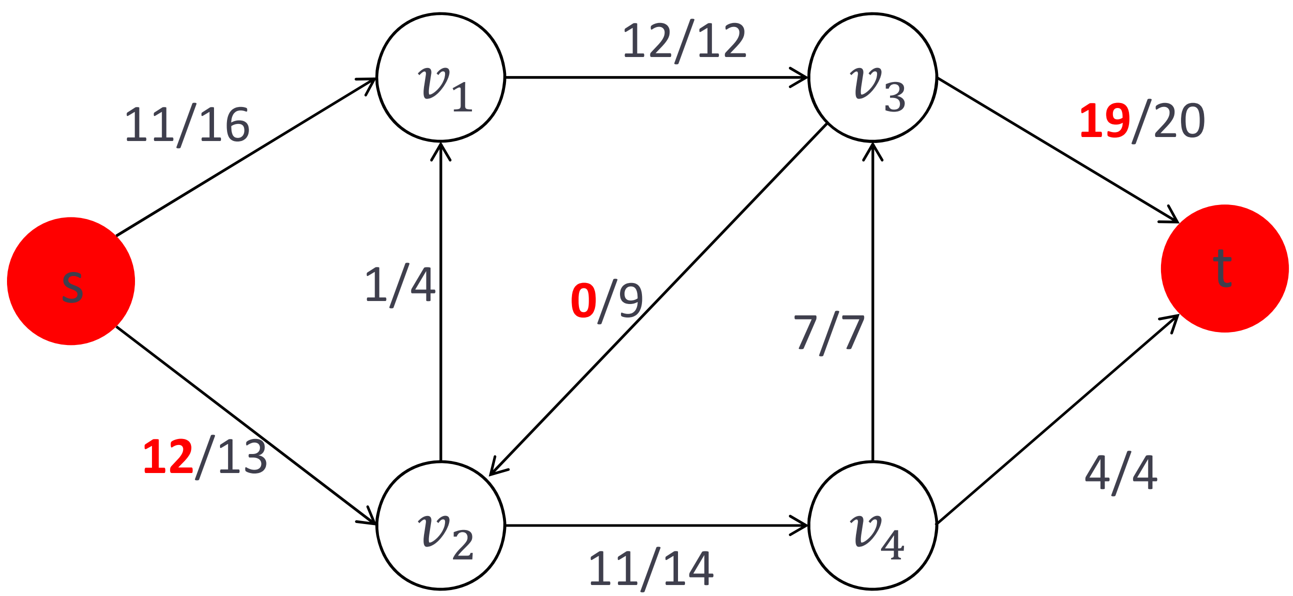
Ford-Fulkerson Algorithm



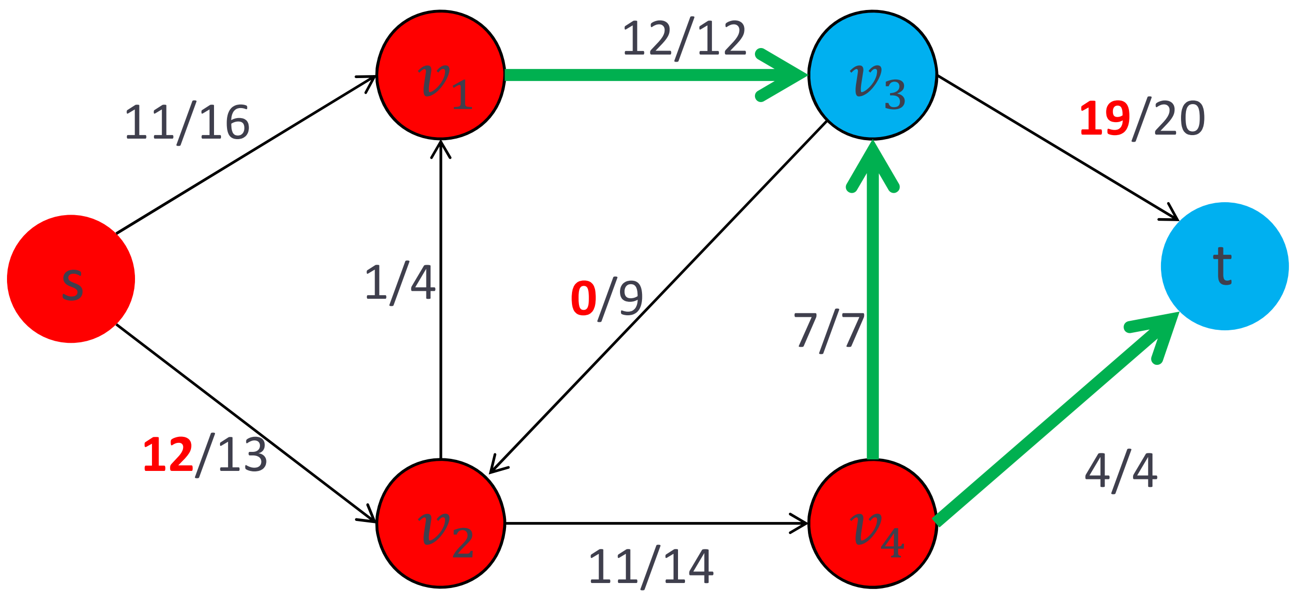


Ford-Fulkerson Algorithm





Ford-Fulkerson Algorithm

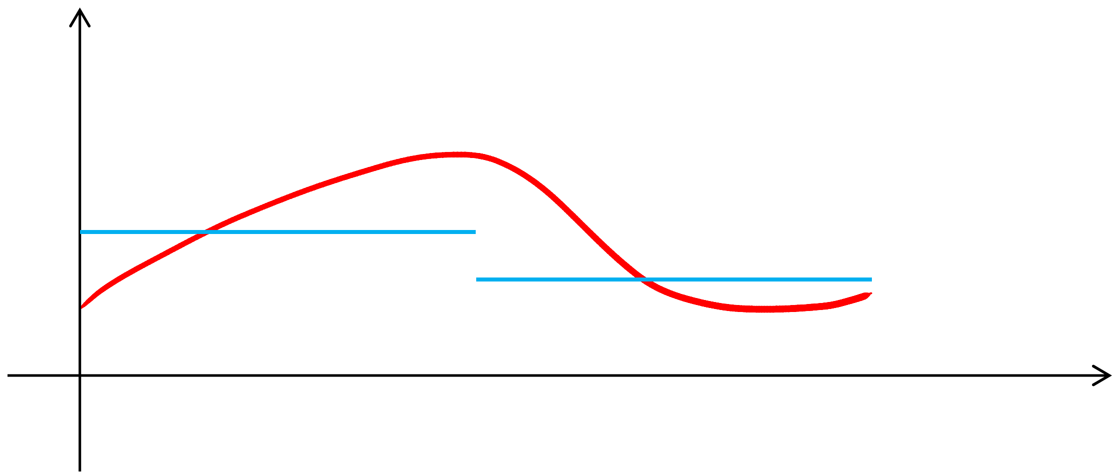


We see that the flow is now maximal; the min cut is shown.

Application to Segmentation

- Many authors: Picard & Ratliff; Grieg & Porteous; Boykov & Kolmogorov; etc.
- Consider the 1D, two-phase segmentation problem:

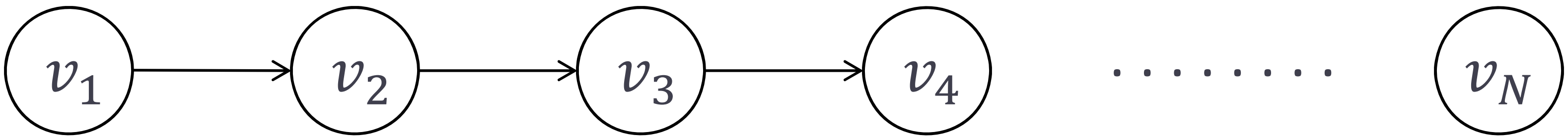
$$\min_{\Sigma} \mathrm{Per}(\Sigma) + \int_{\Sigma} (f - c_1)^2 \, dx + \int_{[0,1]\setminus\Sigma} (f - c_2)^2 \, dx \, .$$



Suppose c_1 and c_2 are given and fixed.

Application to Segmentation

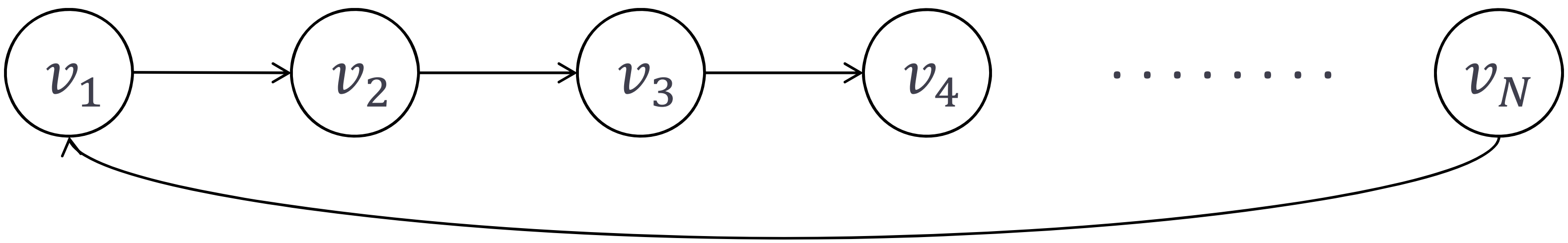
- Discrete version: Grid points $v_j = (j-1)\Delta x$, with $\Delta x = \frac{1}{N+1}$ and $j = 1, \ldots, N$.
- These are the pixels in the image.



- Connect them with edges of capacity 1, left to right.
- Interpixel edges will represent the geometric penalty: Per(Σ).

Application to Segmentation

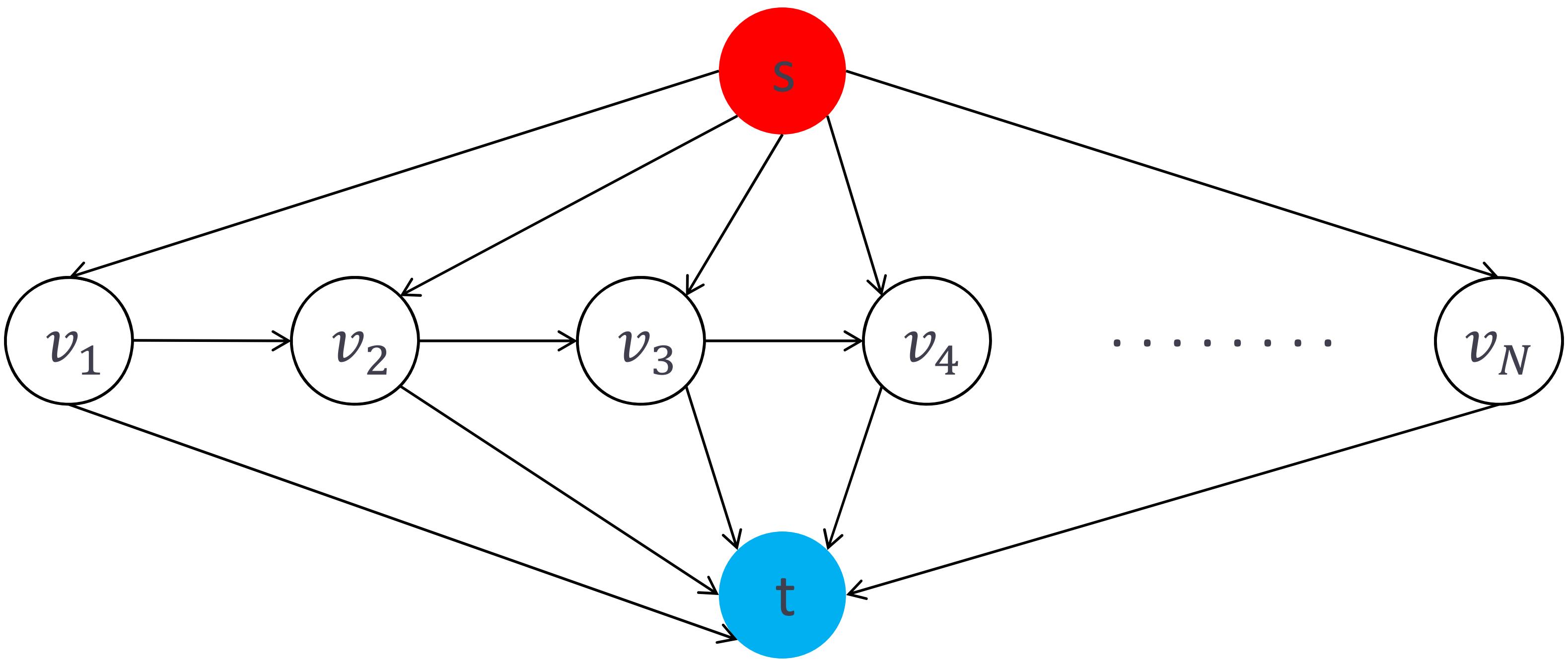
- Discrete version: Grid points $v_j = (j-1)\Delta x$, with $\Delta x = \frac{1}{N+1}$ and $j = 1, \dots, N$.
- These are the pixels in the image.



- Connect them with edges of capacity 1, left to right.
- Interpixel edges will represent the geometric penalty: Per(Σ).
- Also connect v_N to v_1 for periodicity.

Application to Segmentation

- Introduce auxiliary nodes: Source and Sink s and t.
- For each j, introduce edges from s to v_j, and from v_j to t.
- Edges between pixels (v_j) and s or t represent the fidelity term.

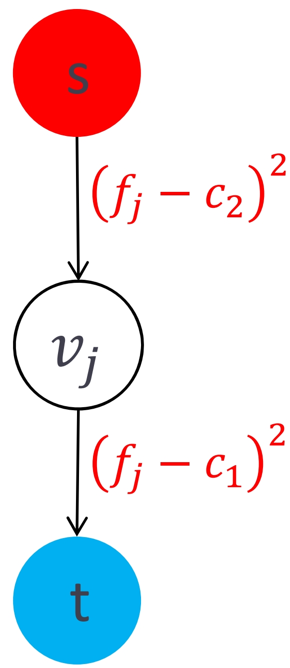


- For a cut (S, T) of the graph, we'll identify:

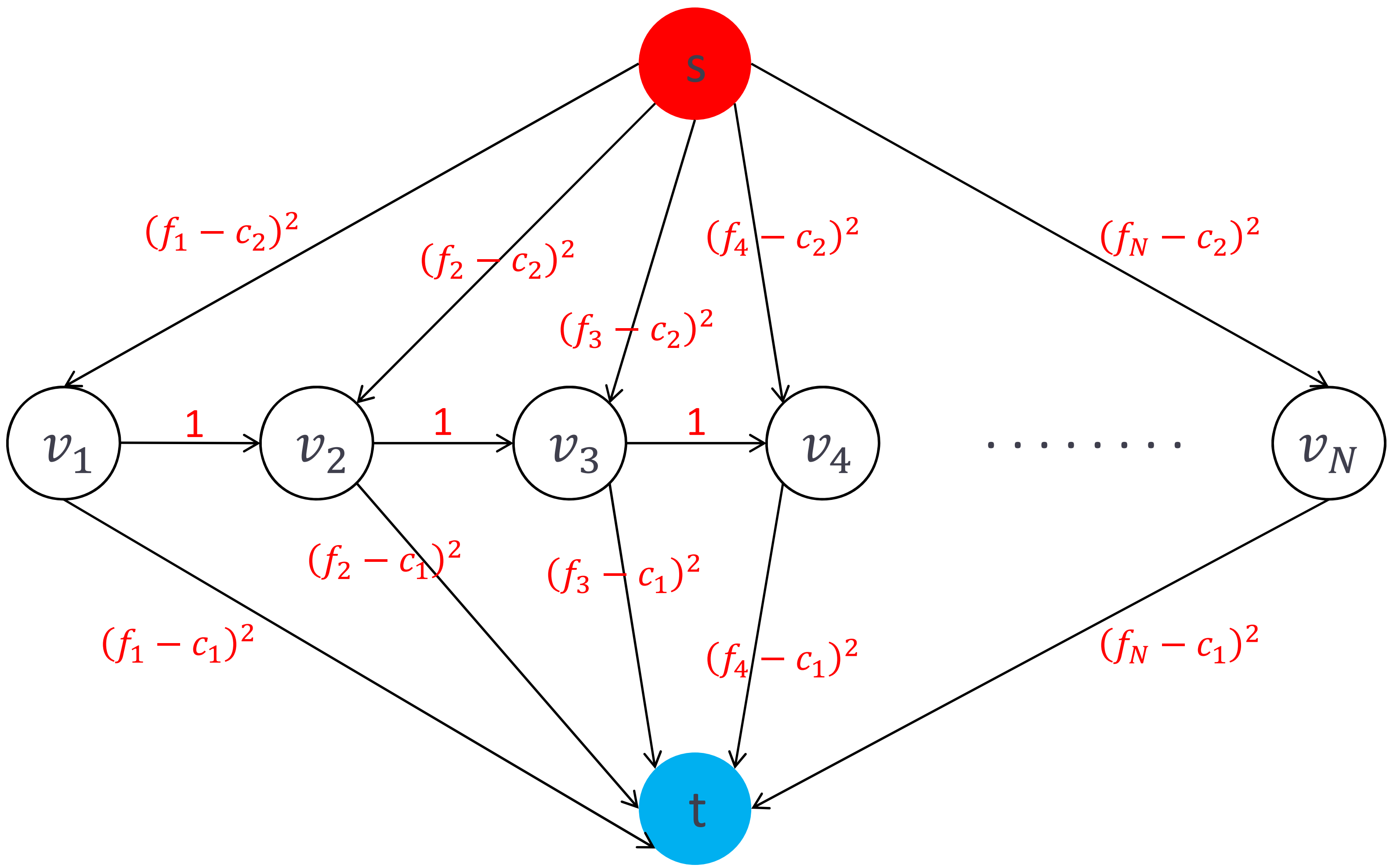
$$\Sigma = \bigcup_{v_j \in S} \{v_j\}$$

Application to Segmentation

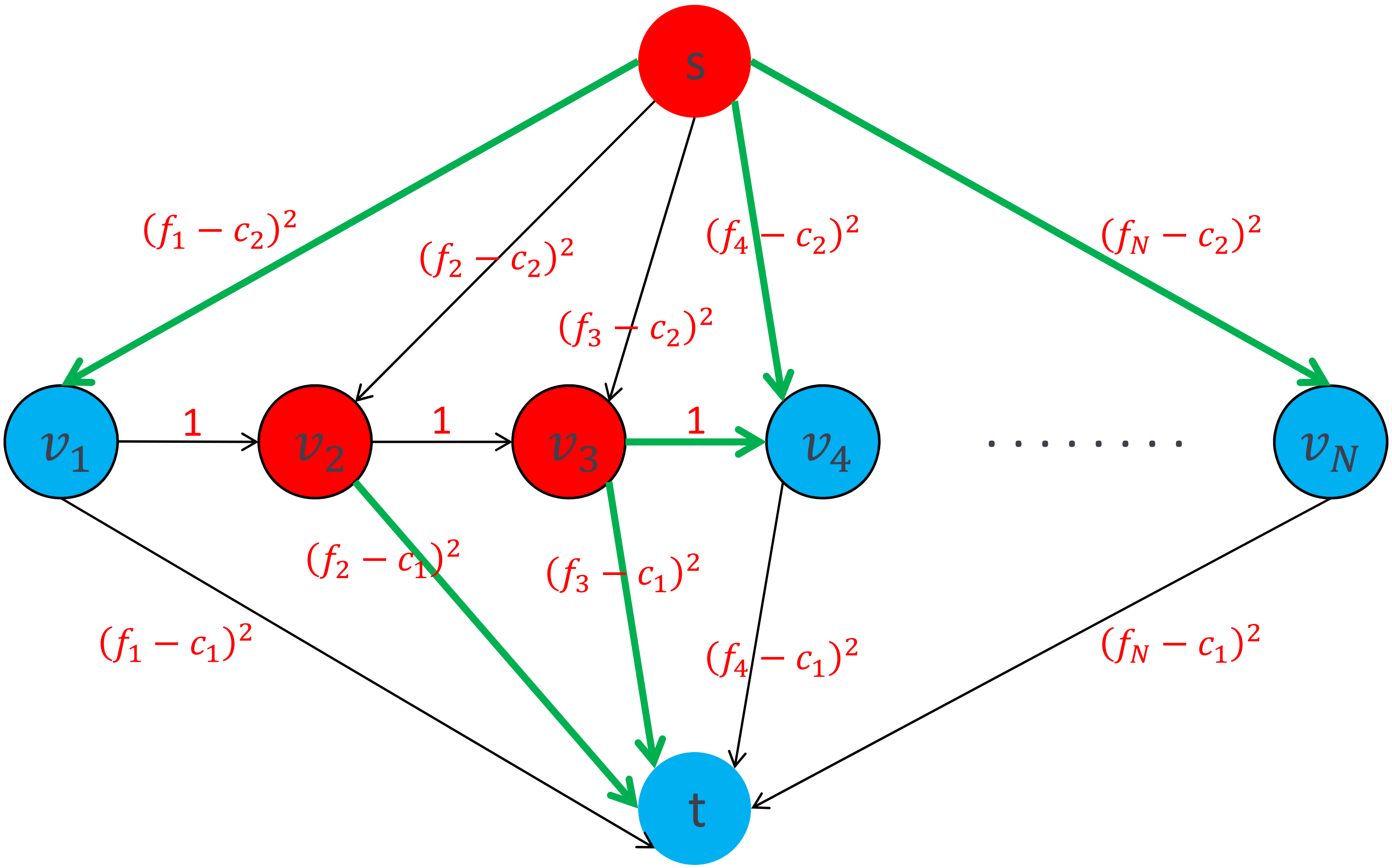
- Fidelity related edges:
  - $e(s, v_j) = (f_j - c_2)^2$.
  - $e(v_j, t) = (f_j - c_1)^2$.



Application to Segmentation



Application to Segmentation



For this cut: $\Sigma = \{v_2, v_3\}$.