Continuous max flows

From soap bubbles to segmentation

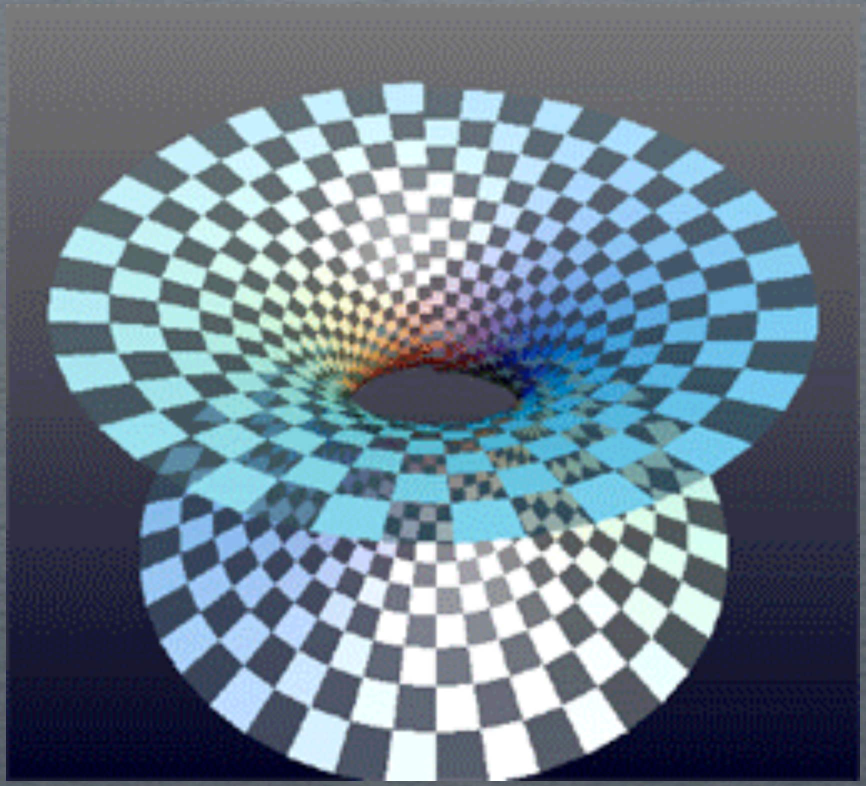
Ben Appleton (Google Inc), Lászlo Márak & Hugues Talbot (ESIEE)

IPAM - March 27 2008

ESIEE PARIS

Continuous max flows

From soap bubbles to segmentation



Ben Appleton (Google Inc), Lászlo Márak & Hugues Talbot (ESIEE)

IPAM - March 27 2008

Motivation

- We are interested in the general problem of object recognition in images (2D or 3D);
- Objects are often recognised by both their content and their contours;
- Assuming the content of an object is semantically describable, their may exist a metric describing this content, constant within the object.

Segmentation

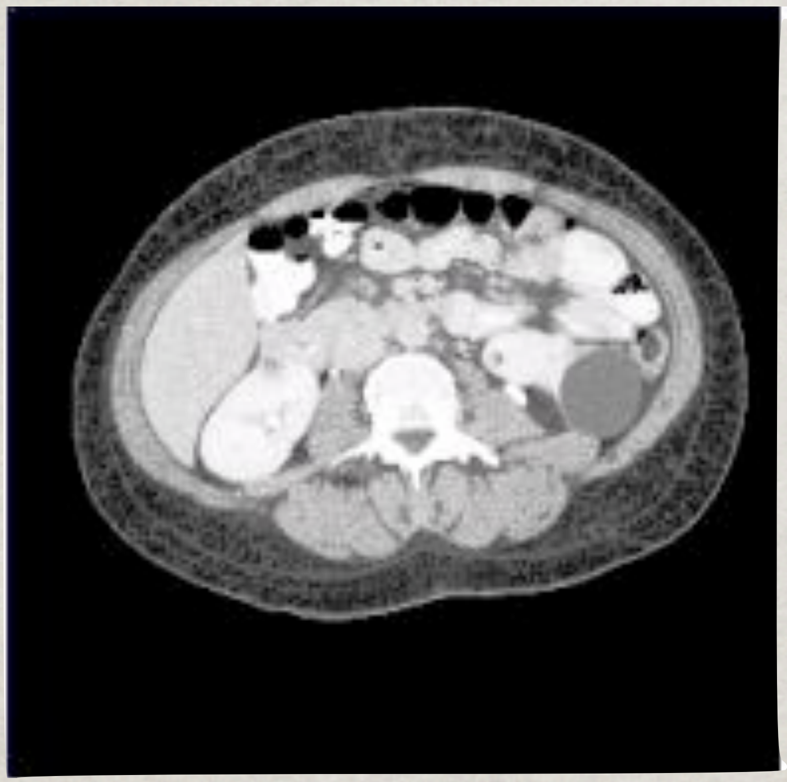
So we need to identify

1. A measure of similarity for the content of the objects, from which we derive a metric ;

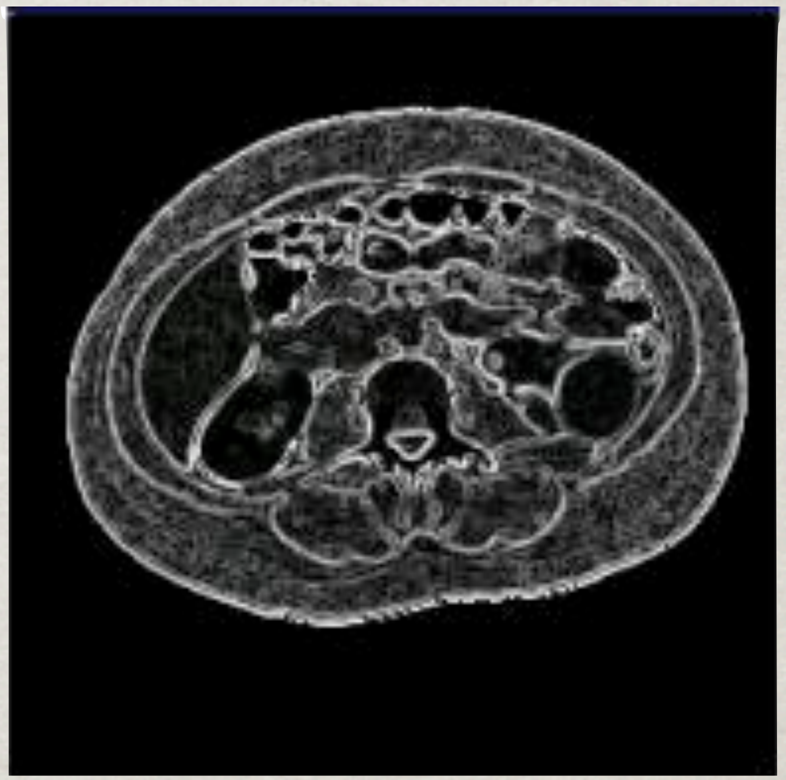
2. An optimisation methodology for the placement of the contours.

3. Identifications require a-priori high-level knowledge.

A simple model



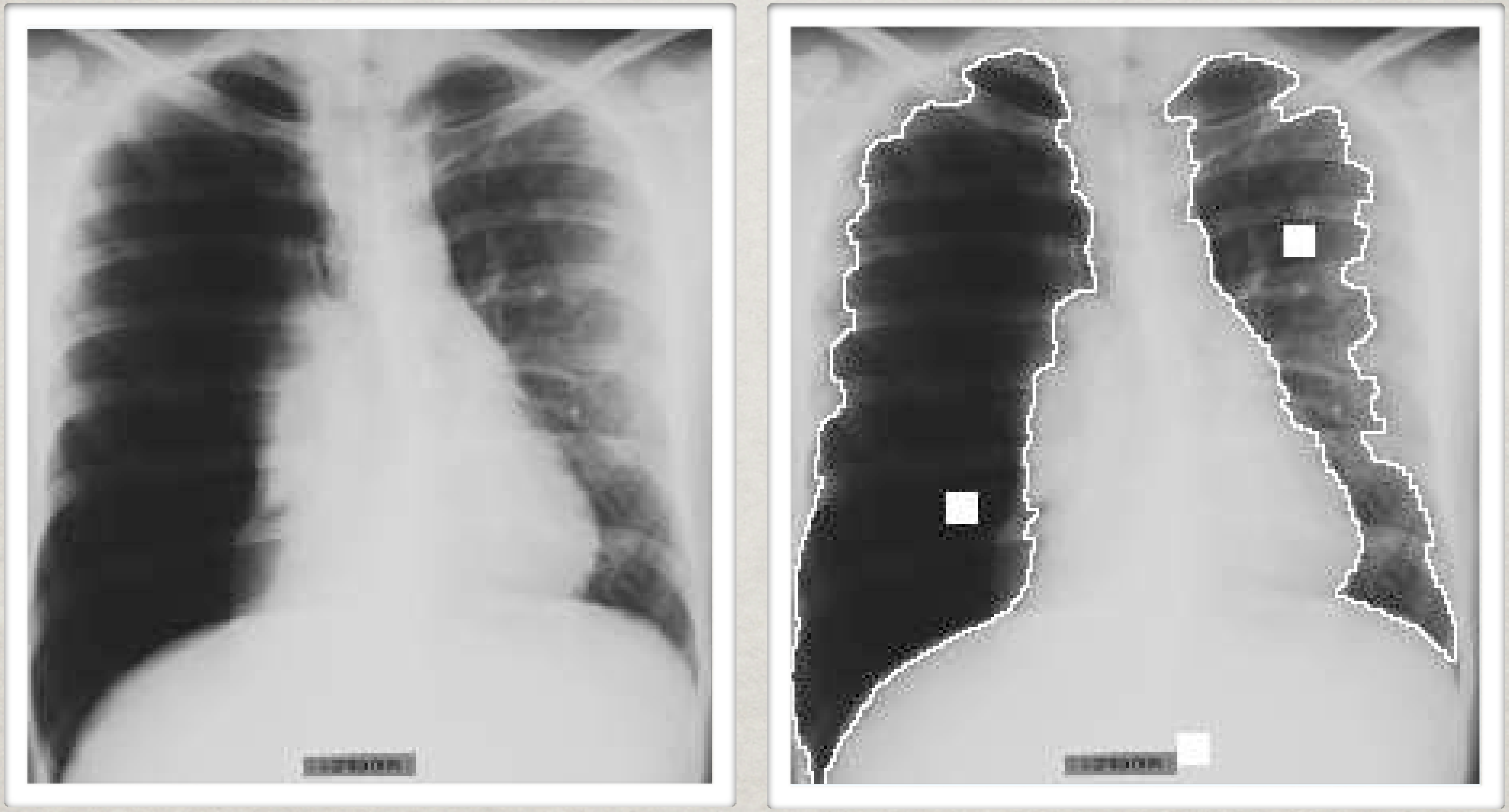
an MRI slice



Contours

Here, the similarity measure is simply the grey-level. Contours are estimated by a gradient. Pb: the gradient may be weak, etc.

Necessary knowledge



High-level knowledge can be obtained manually or automatically : seed placement in this case based on anatomical knowledge on lungs.

Optimisation methods

- Based on contours
  - Contour detection (Marr, Canny)
  - Active contours (snakes)
- Based on regions
  - Watershed, region growing, some LS formulations
- Optimisation : level sets, graph cuts

Soap bubbles

- A soap bubble is a minimal surface. Surface tension physically maximises the soap thickness.

Content of the talk

- We are not going to talk about content similarity measures (but we'll use a metric);
- We worry about the optimisation of the contour placement;
- This is so because content similarity is problem-dependent (texture signatures, etc)

Optimisation for segmentation

- Contour placement can be performed in many ways (hundreds of published methods).

- Optimisation from semantics to mathematics : using a numerical methodology.

- Many ways exist to optimise a cost function, not all equivalent.

- A suitable cost function is not so obvious.

Formulation

- A classic formulation is the following : let I be an image, ∇I its gradient. We look for the contour or surface that minimises

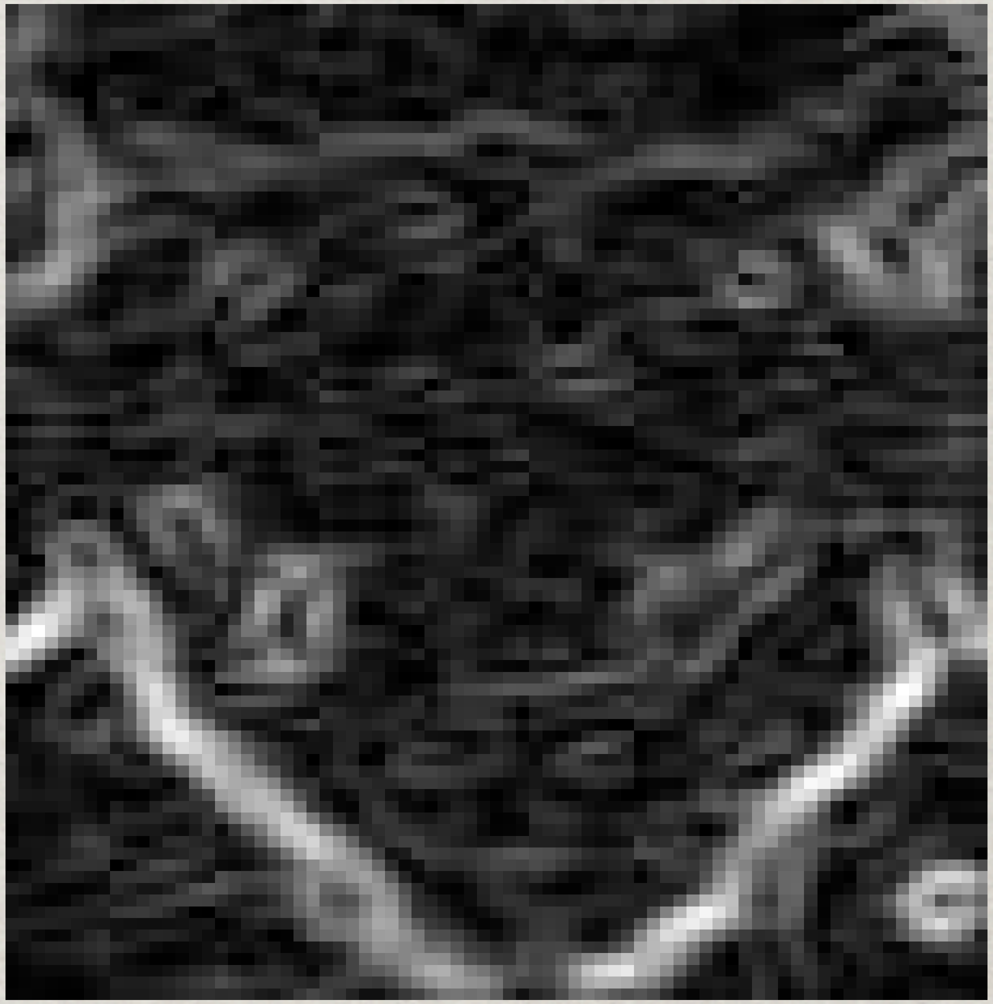
$$\int g(s)ds$$

- With s a measure and g a metric, e.g, s=I and

$$g(s) = \frac{1}{1 + \|\nabla I\|^2}$$

Image gradient

Ultrasound image of a prostate gland



Magnitude of the gradient of the image

Appearance of the metric

An X-ray radiography

Associated metric

The metric has low values (black) near the contours of the image.

HOW TO OPTIMISE?

- How to perform the optimisation

  1. Variational methods

  2. Graph-based methods

  3. Continuous maxium flows

Variational methods

- Based on the study of the first variation

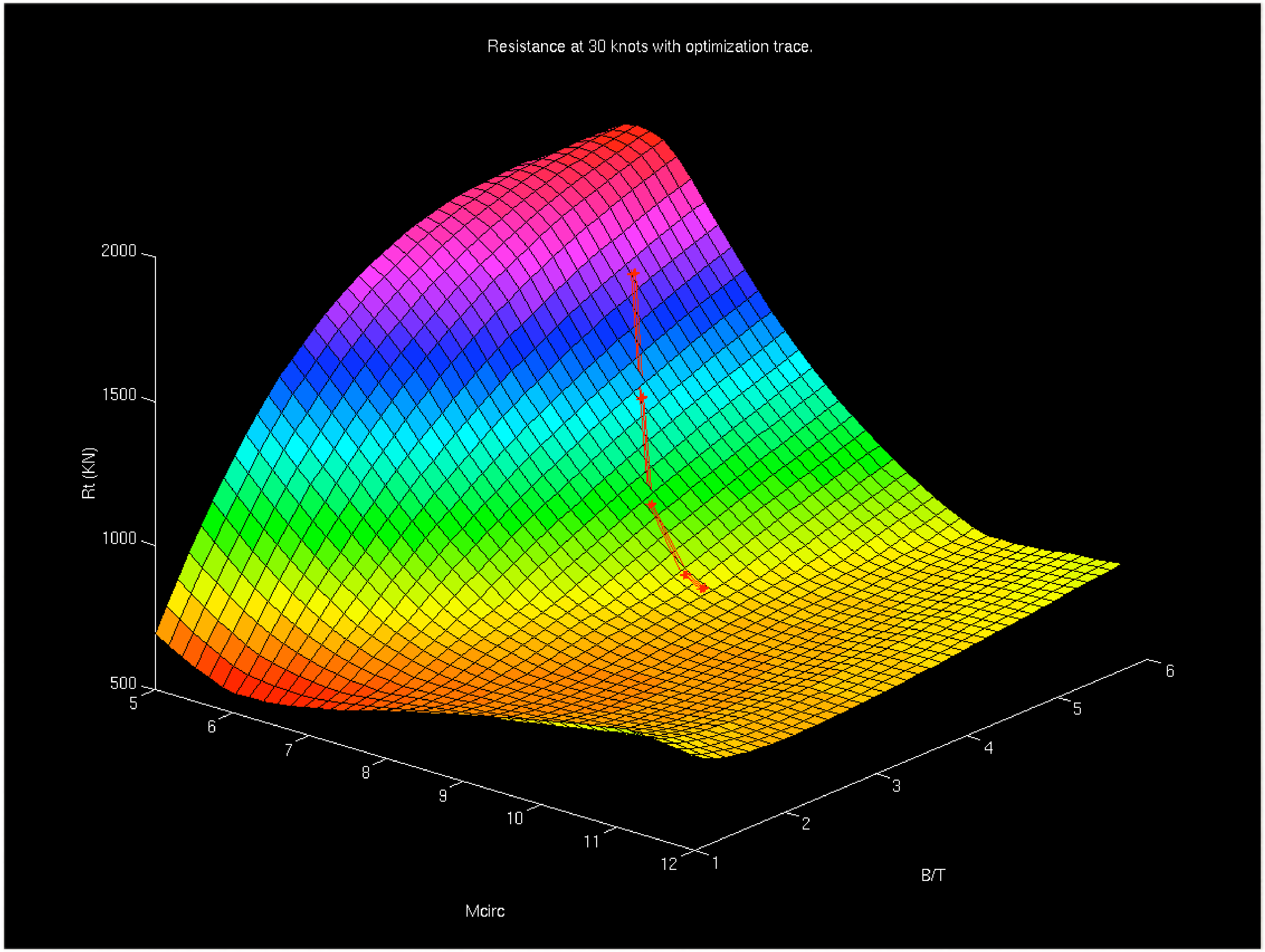
$$L(r) = \int_a^b \mathcal{L}(x, r(x), r_x(x))\mathrm{d}x$$

$$\bar{r} = \arg\ \min_r L(r)$$

- Solved by derivation : NNSC Euler-Lagrange

$$\frac{\delta L(r)}{\delta r} = \frac{\partial \mathcal{L}}{\partial r} - \frac{\mathrm{d}}{\mathrm{dx}}\left(\frac{\partial \mathcal{L}}{\partial r_x}\right) = 0$$

Gradient descent



To find the minimum, follow the path of steepest descent.

VARIATION AND DESCENT

- The previous expression is a derivative, that vanishes at an extremum.

- A gradient descent is obtained by considering a time variable:

$$\frac{\delta r}{\delta t} = \frac{\delta \mathcal{L}(r)}{\delta r}$$

- A contour can be represented explicitely (Lagrangian method) or implicitely (Eulerian method).

Contours actifs géodésiques

- With the equation we gave in the beginning, the EL/GD formulation is :

$$\varphi_t = \mathrm{div}\left(g\frac{\nabla\varphi}{|\nabla\varphi|}\right)|\nabla\varphi|$$

- To simplify the expression, the contour (surface) is implicitely represented by the zero-level-set of a function φ.

- This is the evolution equation of the *geodesic active contours* (GAC).

Local minima

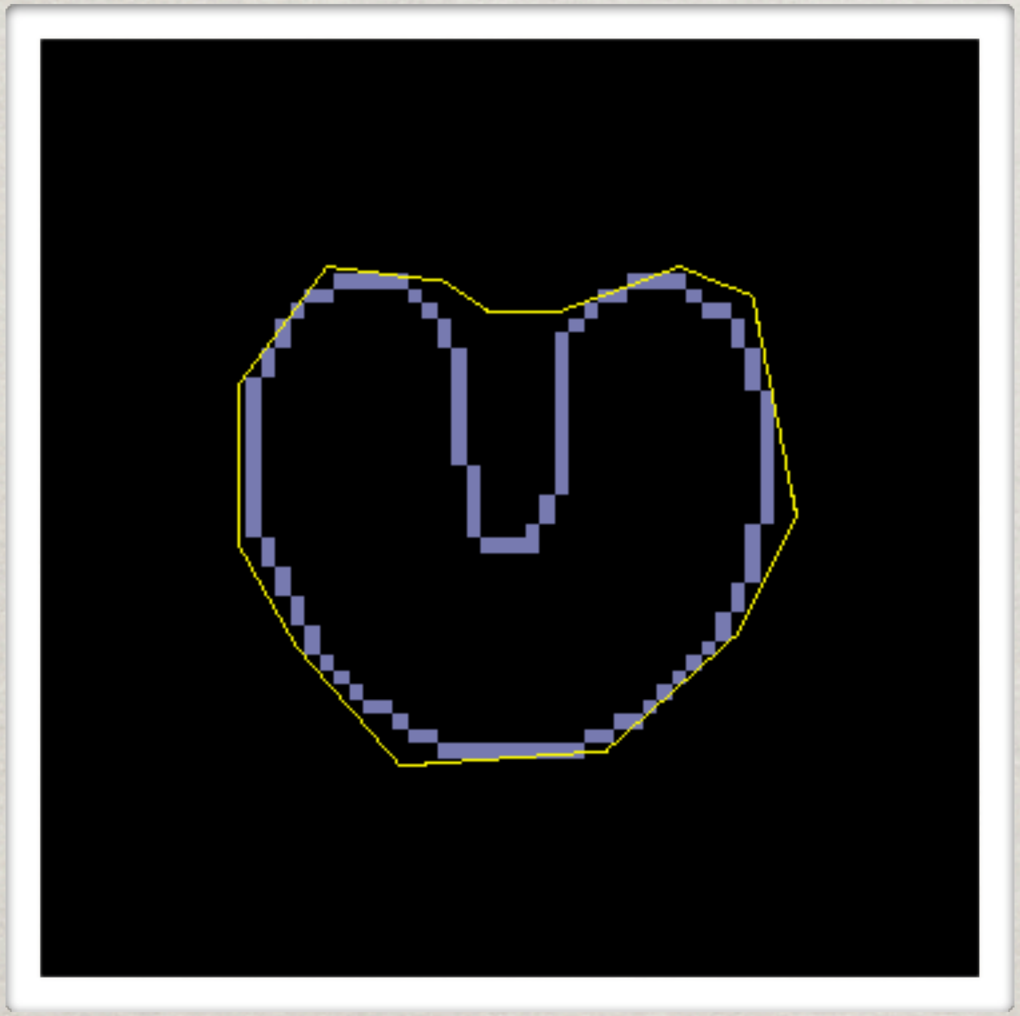
Gradient descent generally only optimises locally

Local optimum

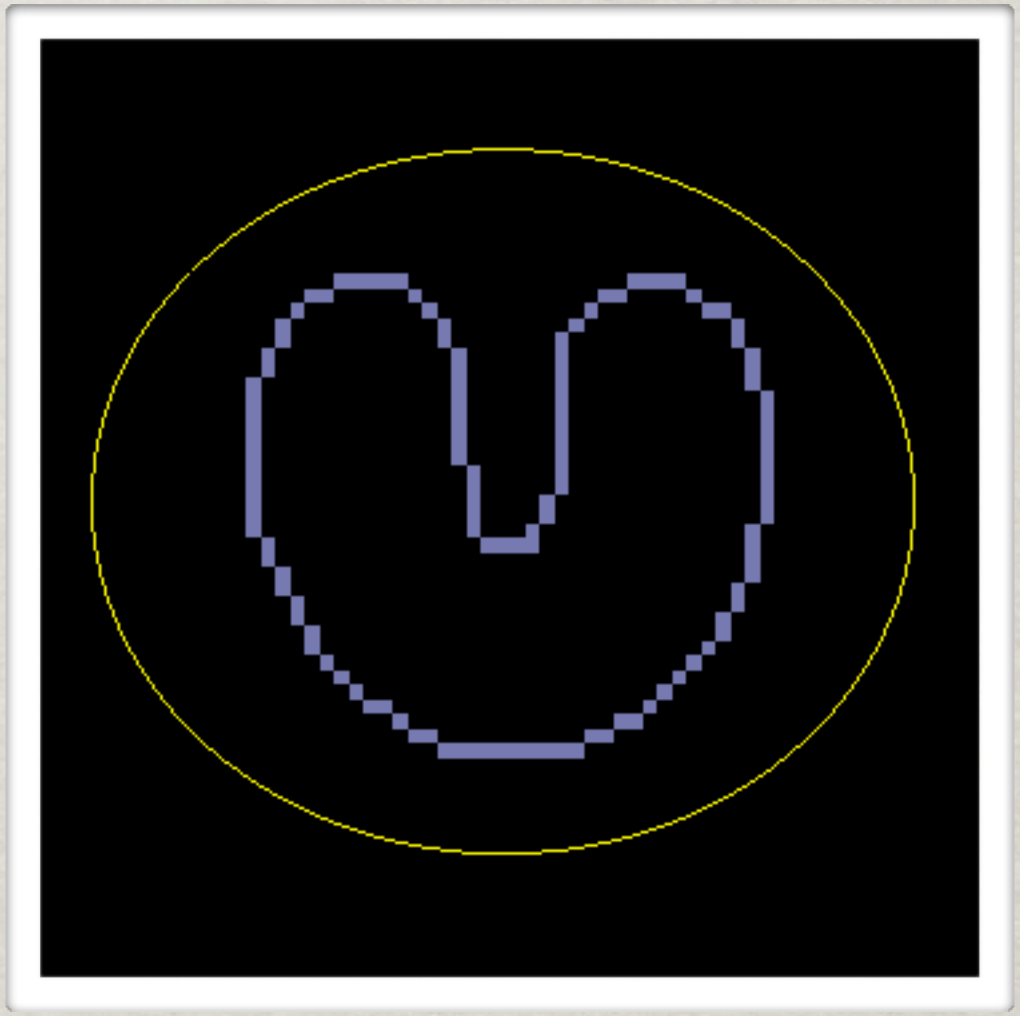
Global optimum

Local minima

Gradient descent generally only optimises locally



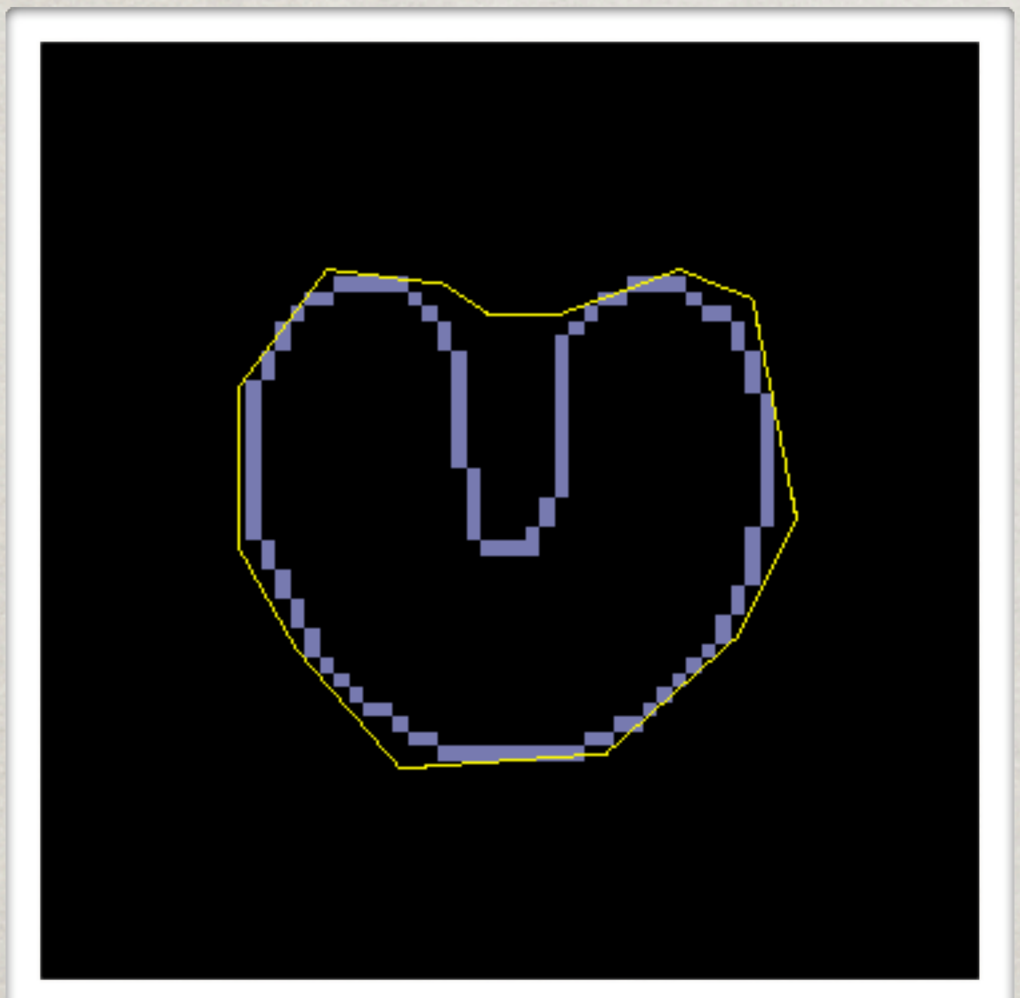
Local optimum



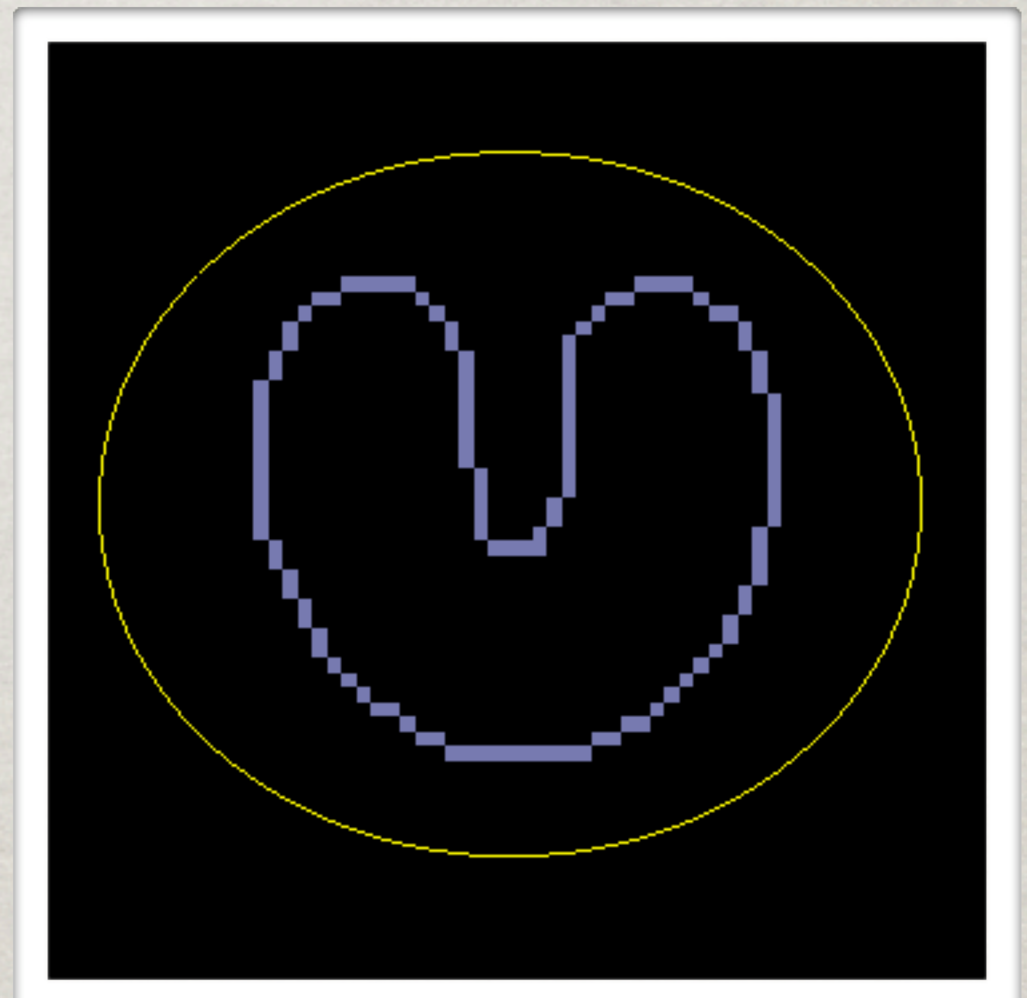
Global optimum

Local minima

Gradient descent generally only optimises locally



Local optimum

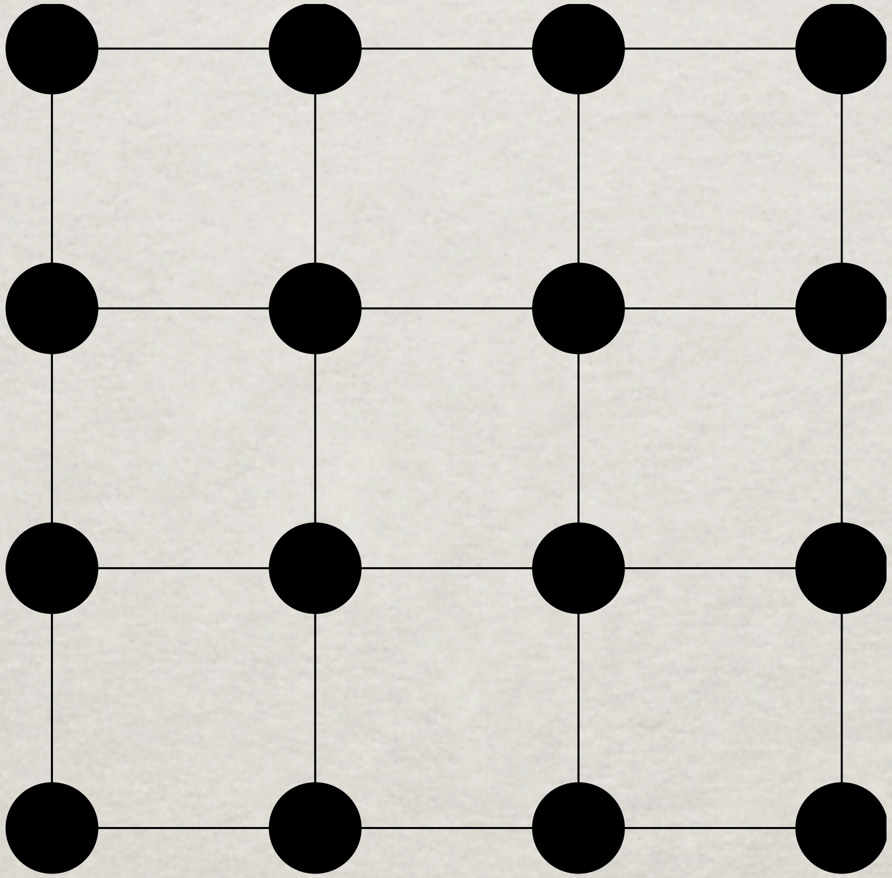


Global optimum

Graph-based methods

- The initial variational problem can be seen in 2D as a minimal path problem.
- In 3D (and more), it can be formalized as a maxflow problem ;
- Max-flow / min-cut can be solved on graph representations.

Example of graphs



Typical graph used in image analysis

Flots maximaux dans les graphes

- Let G be a graph with weighted edges
- so that : $\Gamma_G = \{V_1, V_2, \ldots\}$

$$\bigcup_{V_i \in \Gamma_G} V_i = V, V_i \cap V_j = \emptyset \quad \text{for} \quad i \neq j.$$

- The cost of a partition is the sum of the edges that cross a partition :

$$C\left(\Gamma_G\right) = \sum_{e \in E^*} C_E\left(e\right).$$

- The cut E^* is the set of shared edges.

Maximum flows

- If G is a graph with costs reinterpreted as *capacities*,
  - A flow F from the source s to the sink t has the following properties :

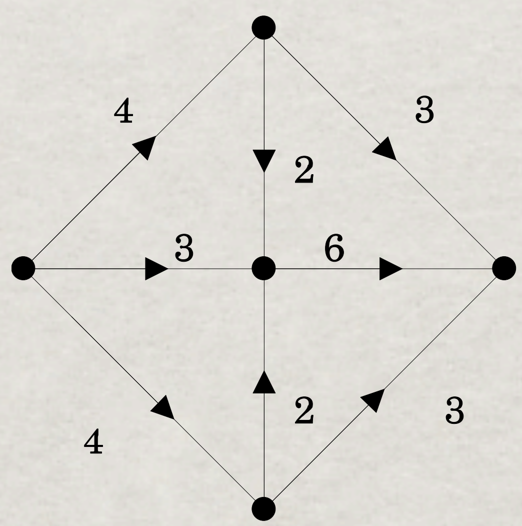
    1. Flow conservation : the flow entering an edge is the same as that exiting that edge;

    2. Capacity constraint :

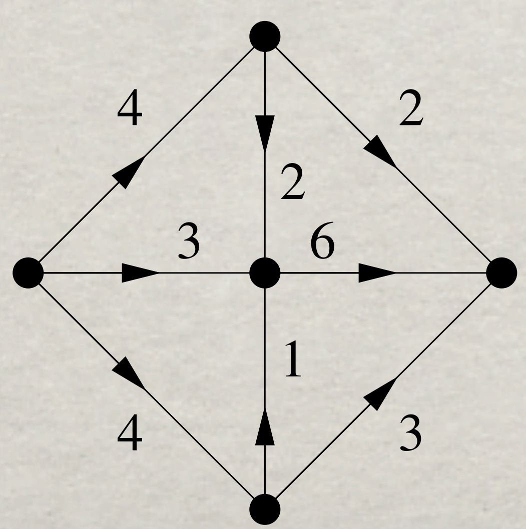
$$\forall e \in E, \quad F(e) \leq C_E(e)$$

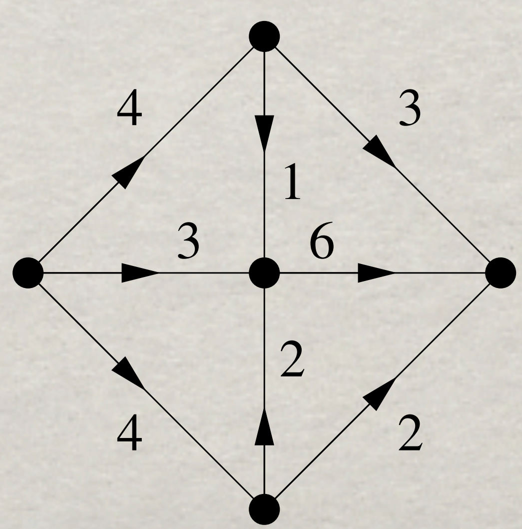
Examples of max flows

- Capacity graph:



- Two possible solutions:





Cut

- The cut is the set of edges separating the partitions ;
- The min cut is the cut of minimal cost (sum of all the capacities of the cut) ;
- The solution to the problem with one source and sink has been known since Ford & Fulkerson (1962) ;
- The general problem with S sources and T sinks is NP-hard.

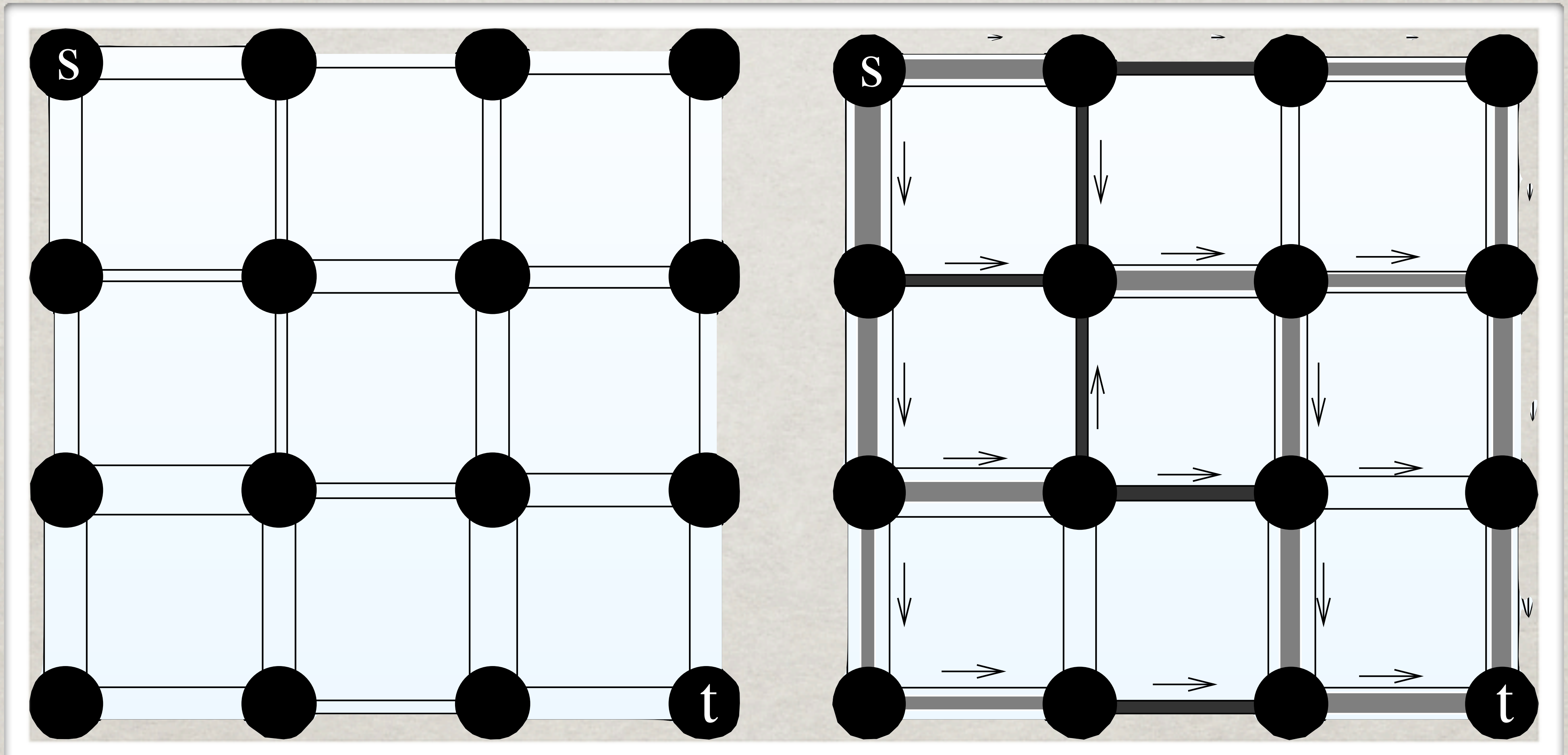
Min cut

- An edge along which the flow is equal to the flow is said to be *saturated* ;

- A max flow in a weighted graph G maximises the flow through the network from s to t. F&F showed that calculating the maxflow s-t is equivalent to computing the mincut s-t.

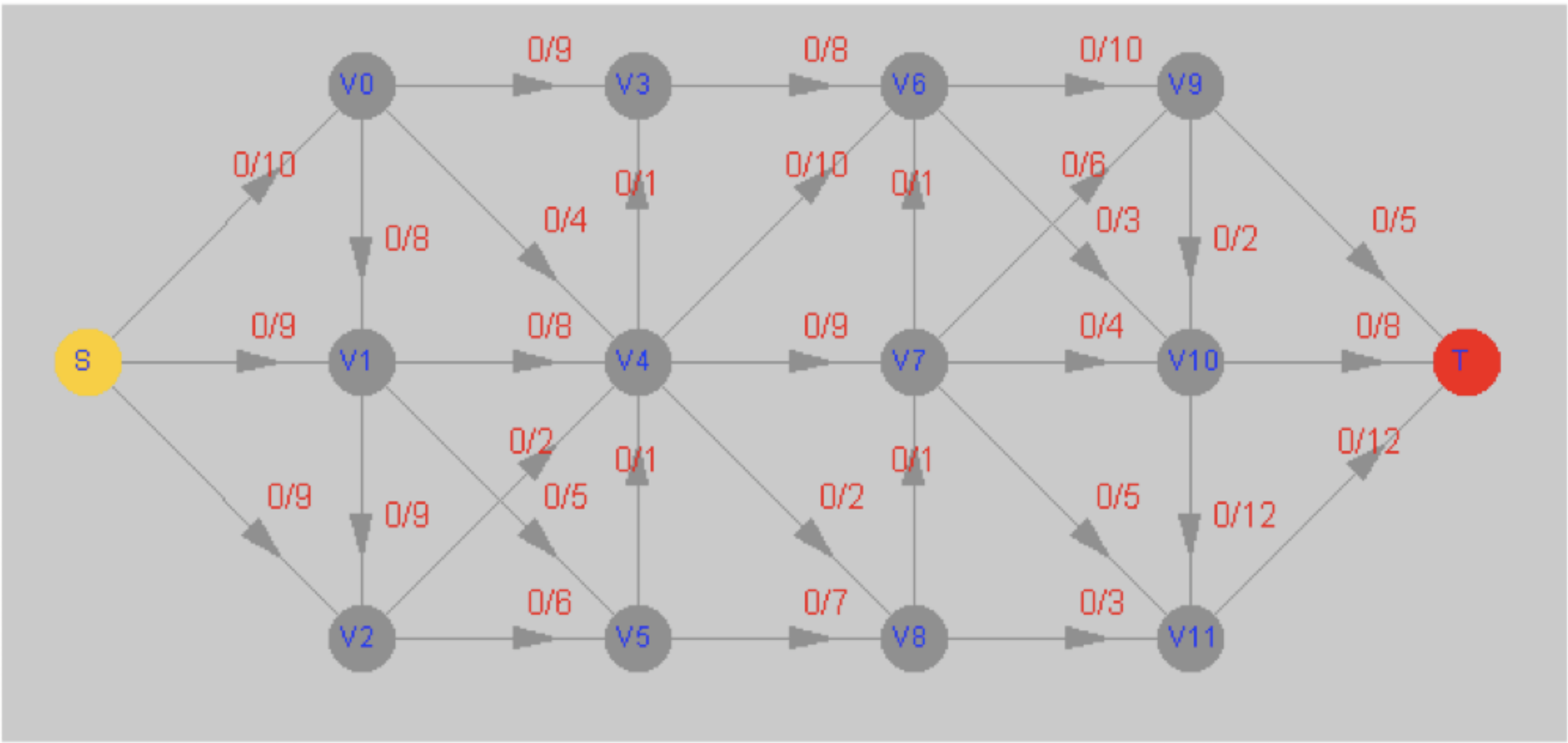
- The mincut s-t is a minimal path in 2D and a minimal surface in 3D (in a suitable dual graph).

Minimal cut



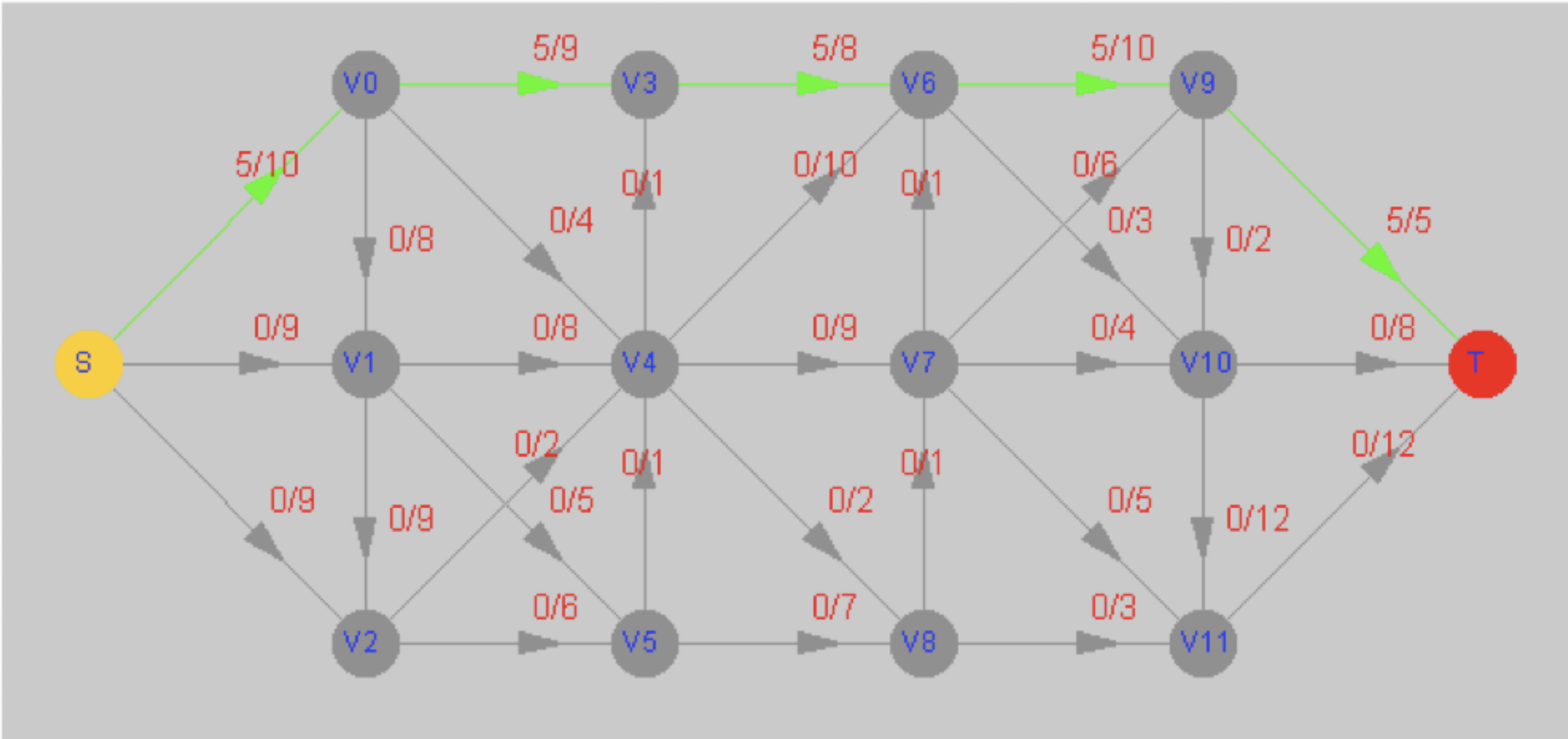
A network for which capacities are illustrated by edge width. The saturated edges separate s from t and are of minimal cost.

Path augmentation



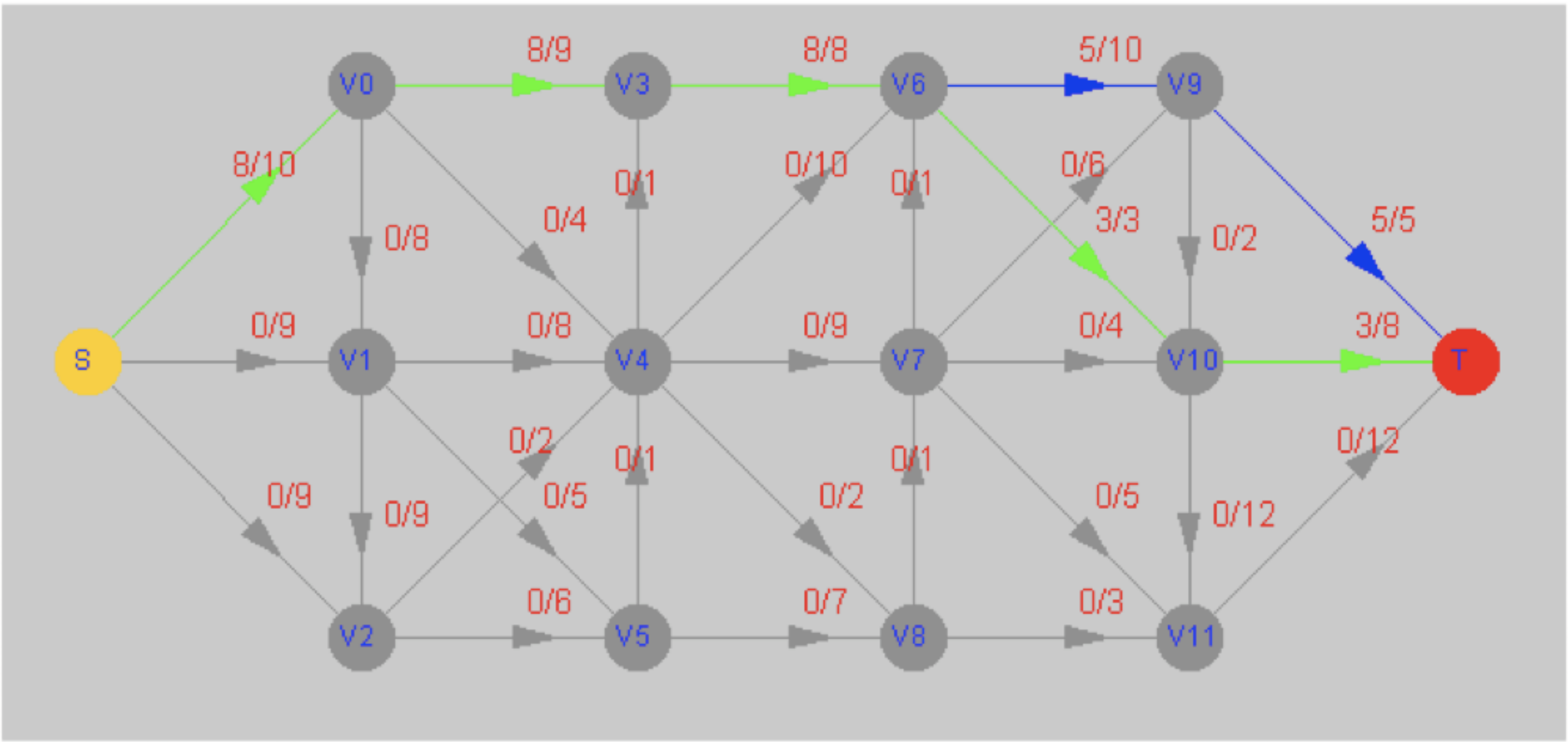
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



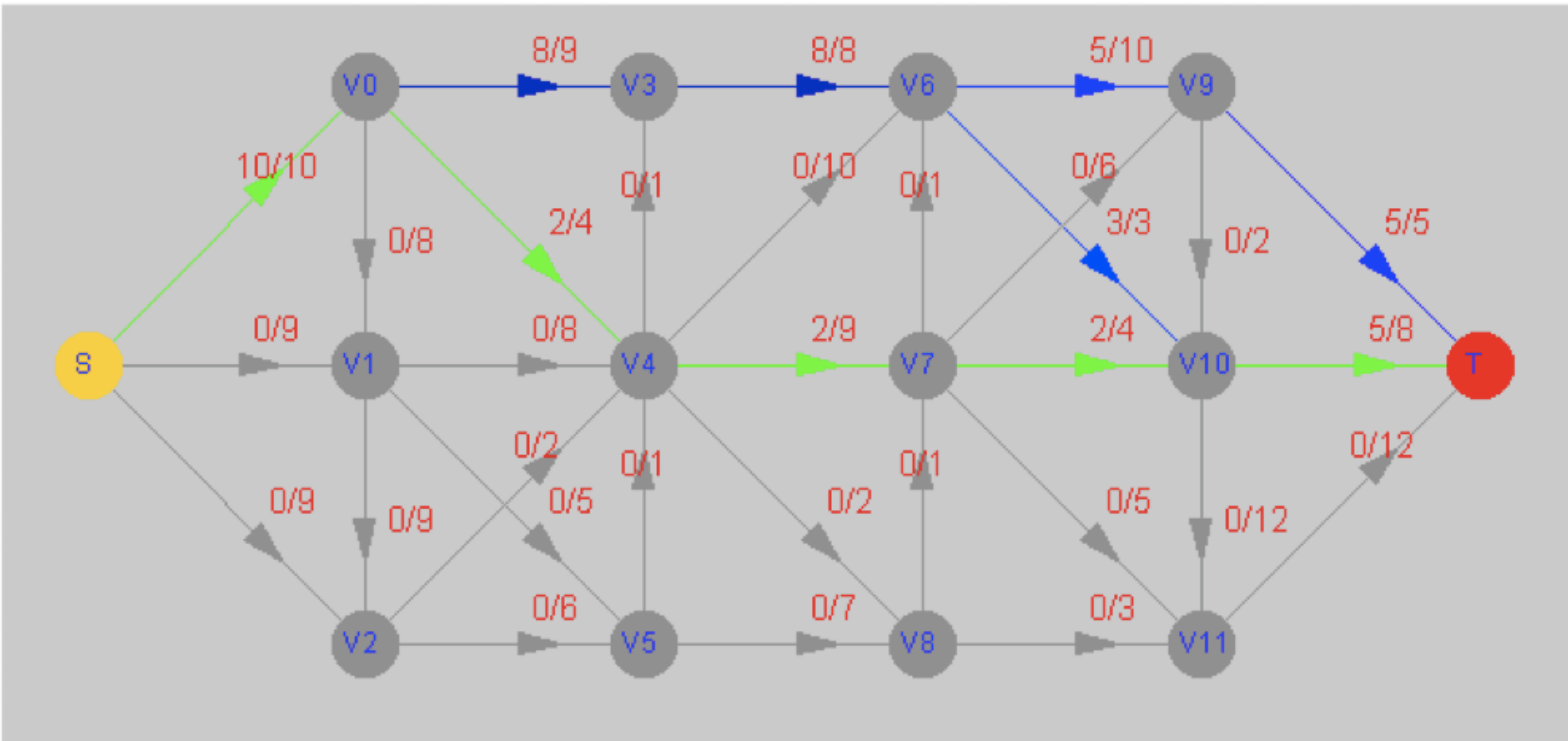
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



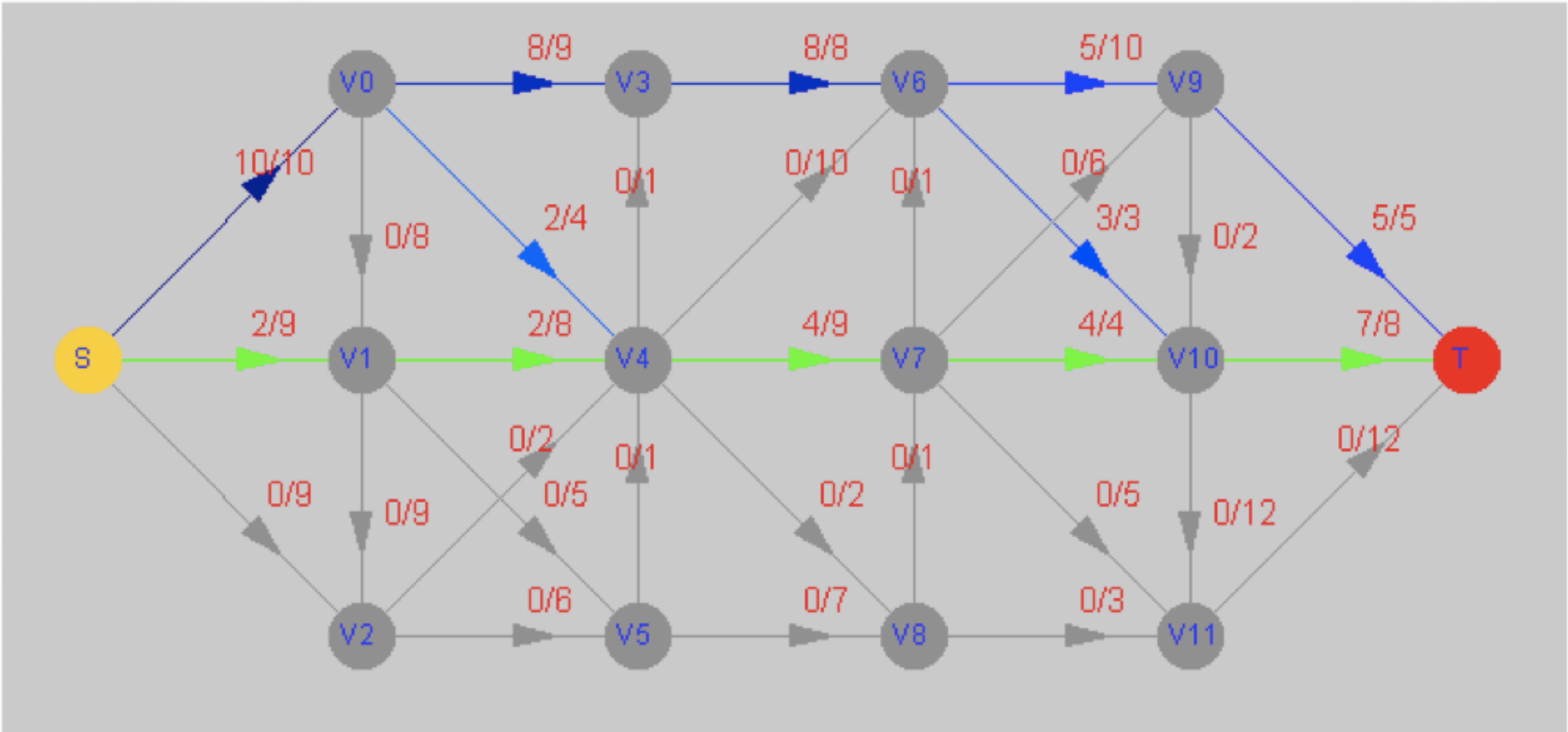
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



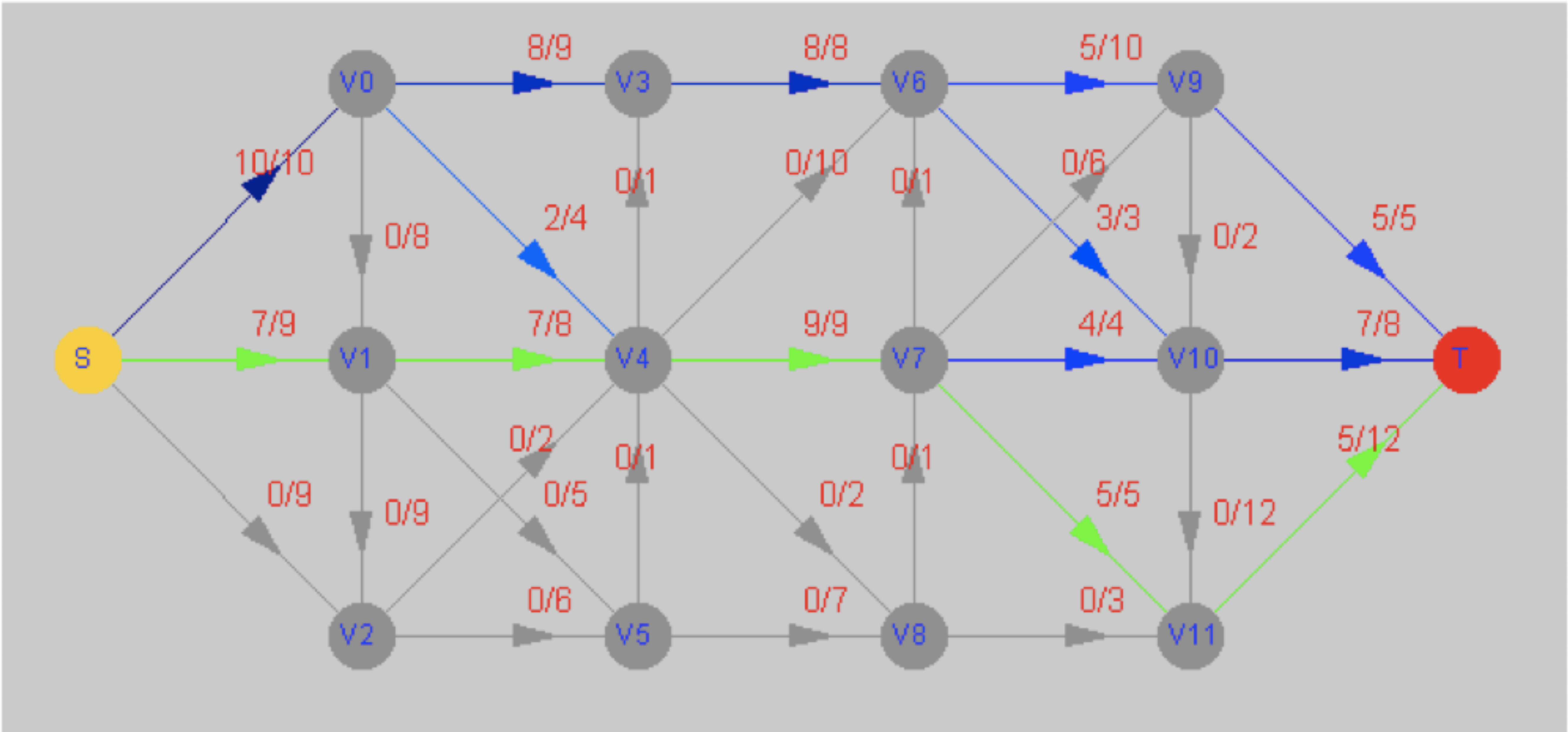
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



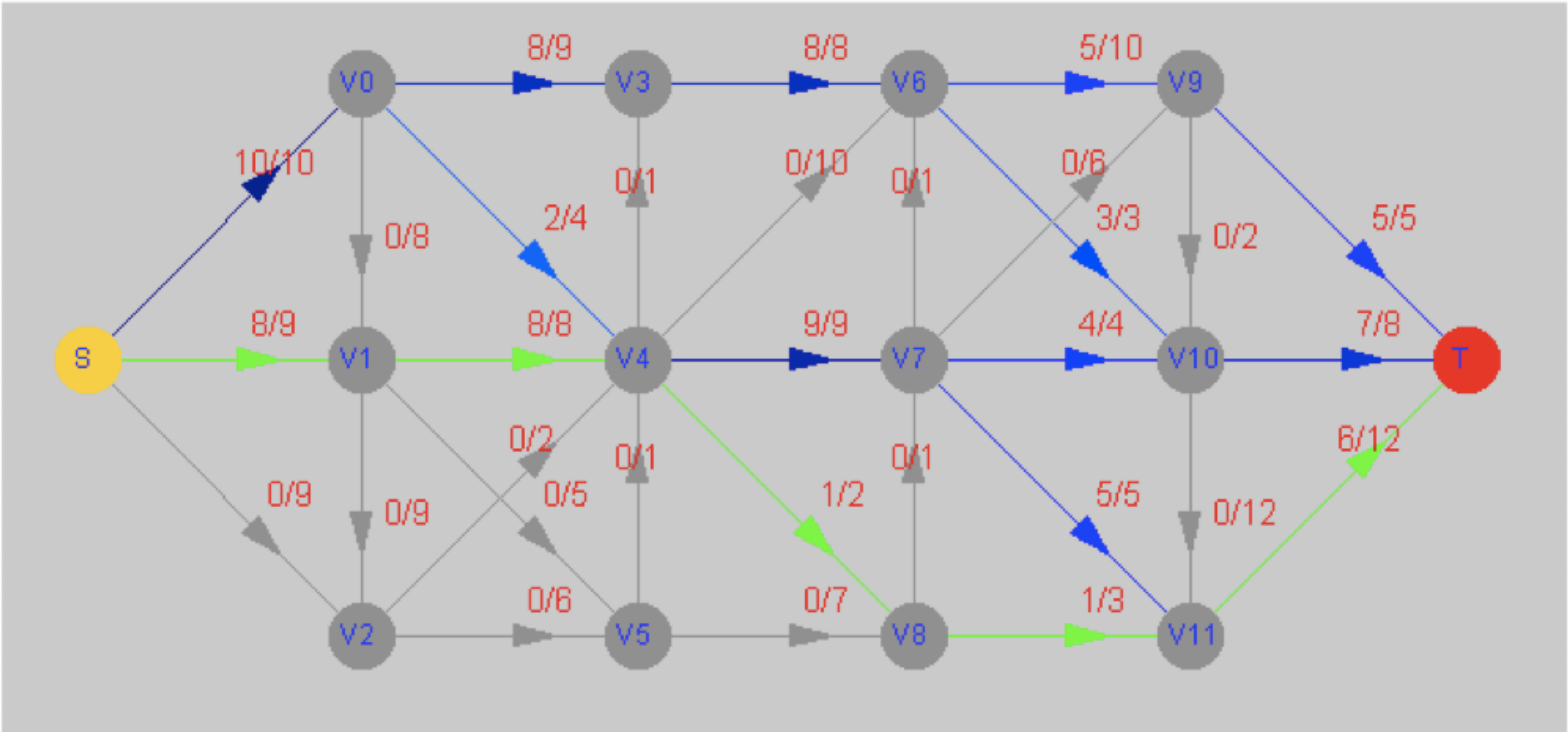
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



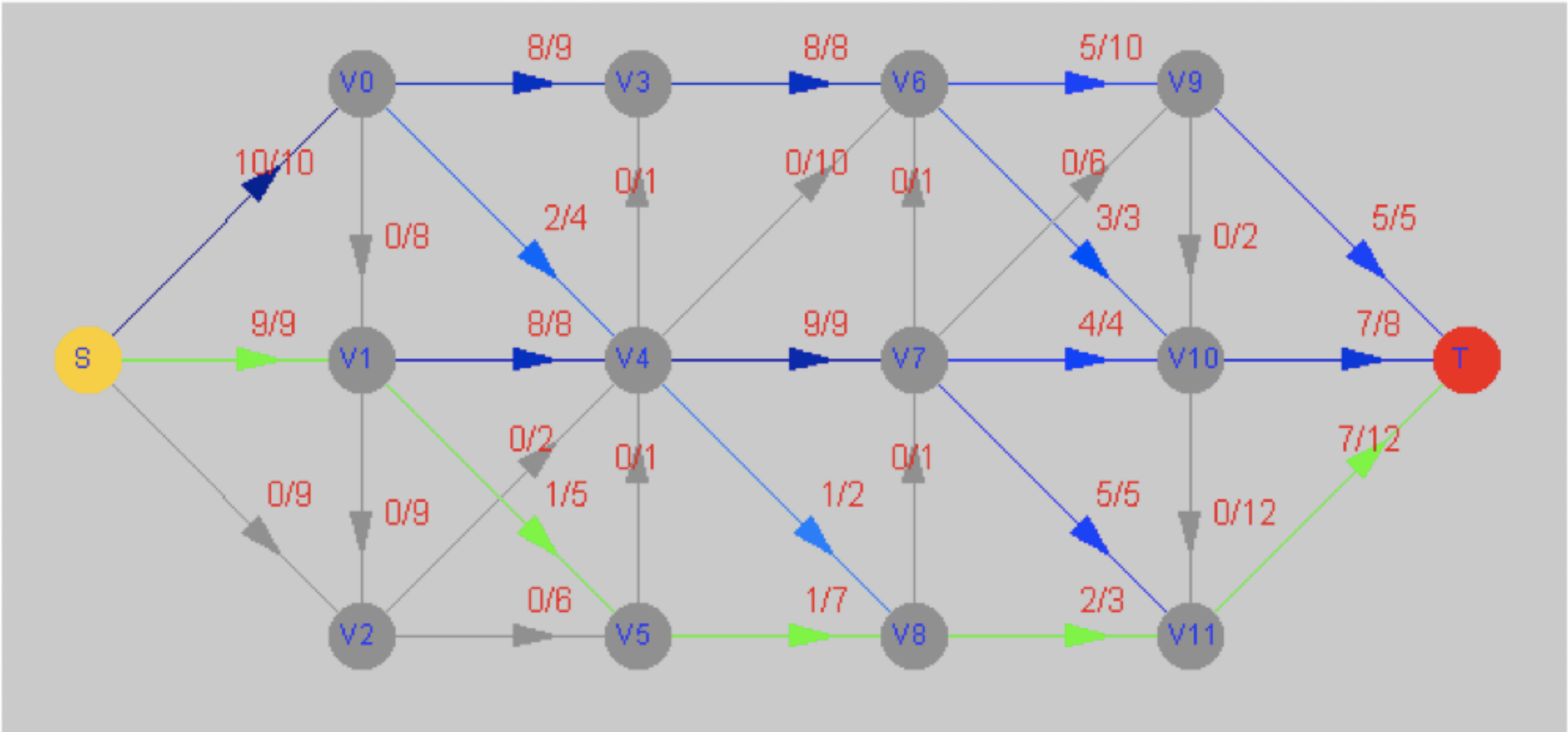
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



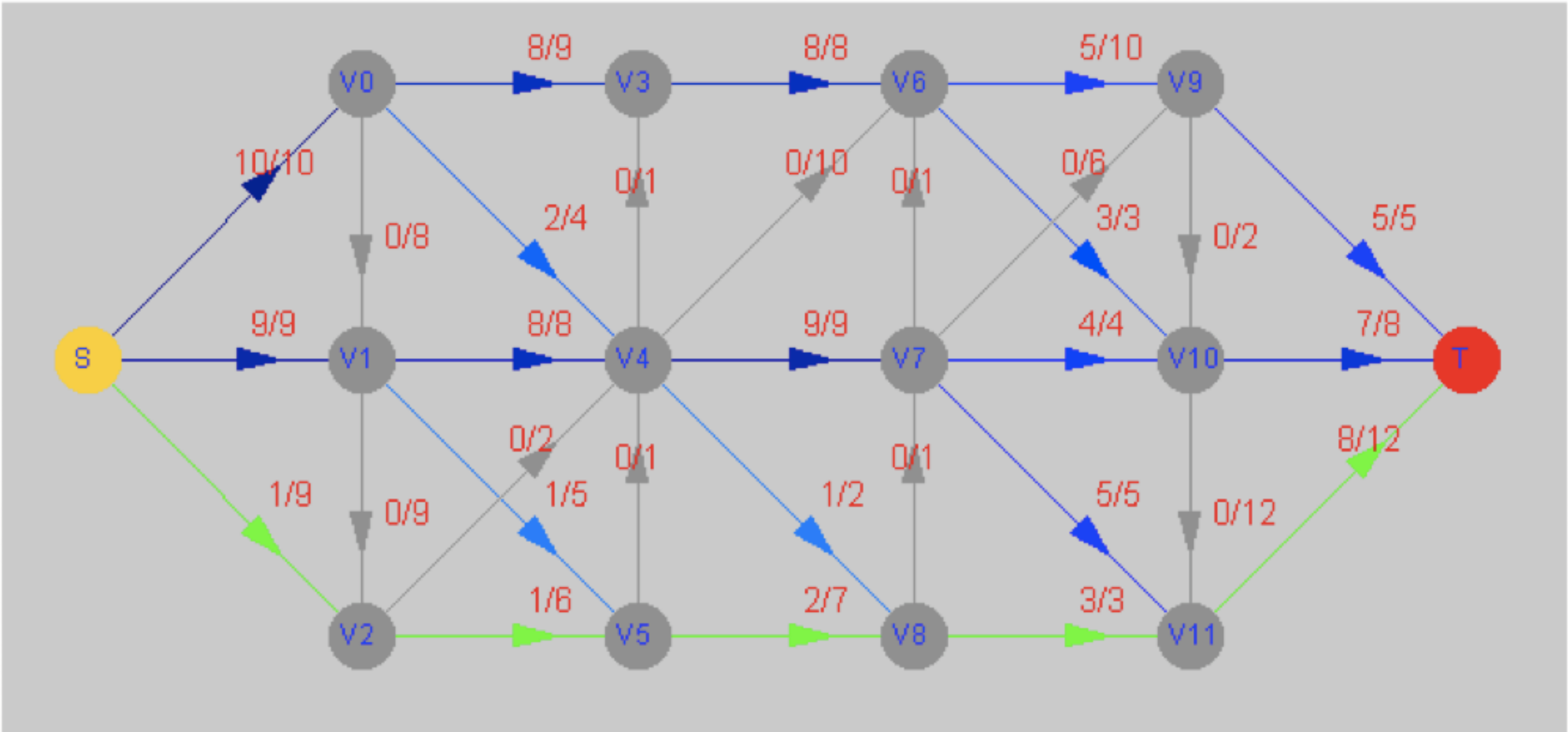
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



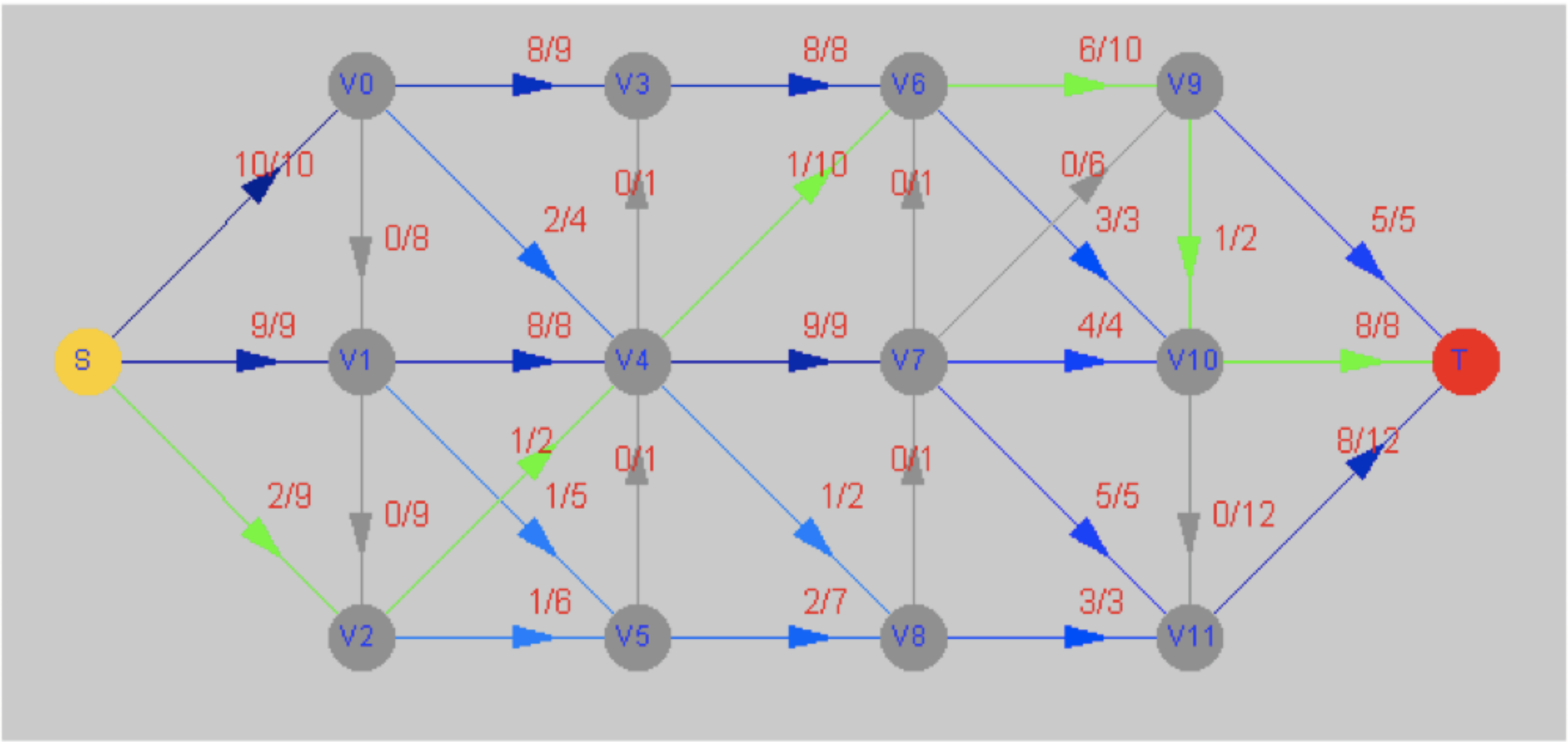
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



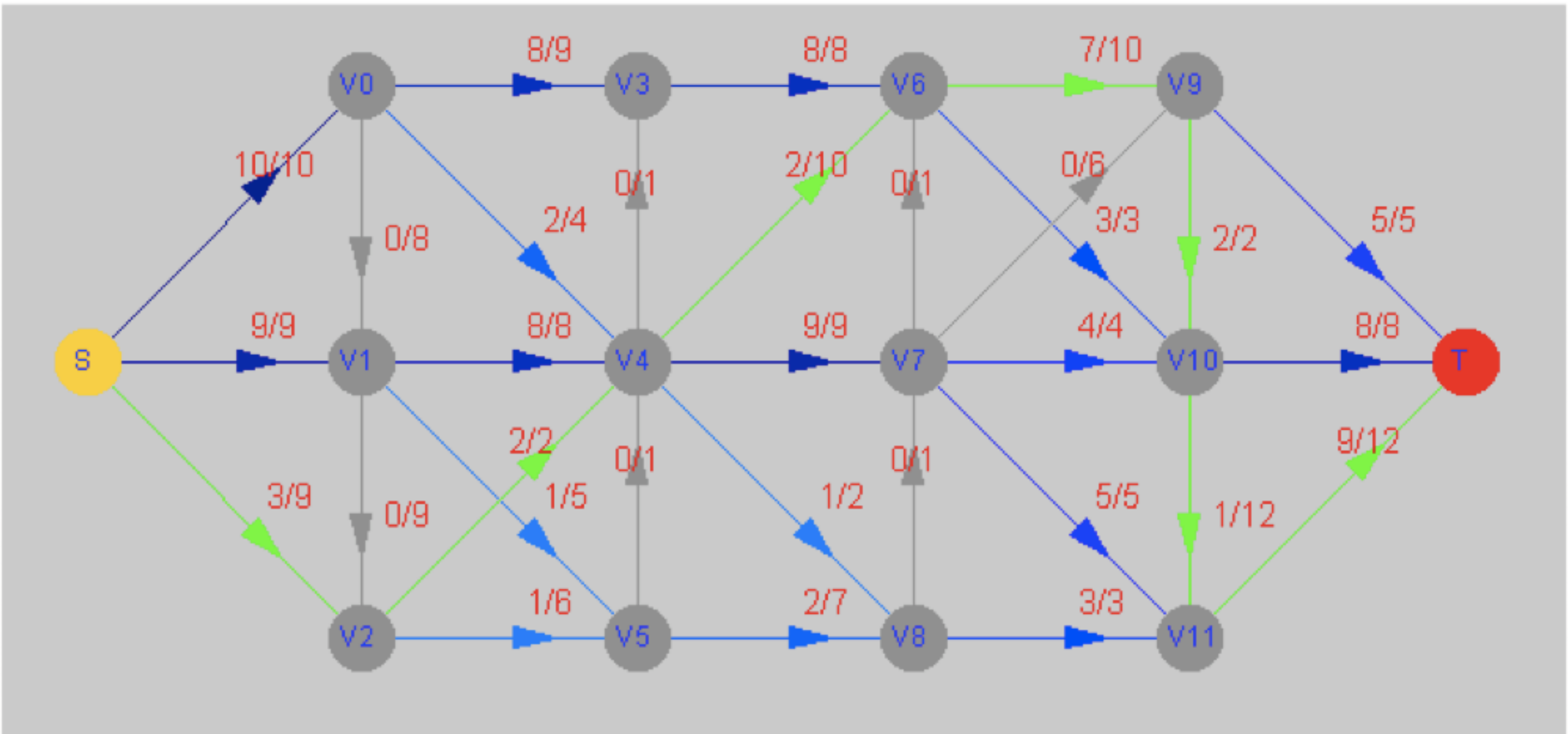
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



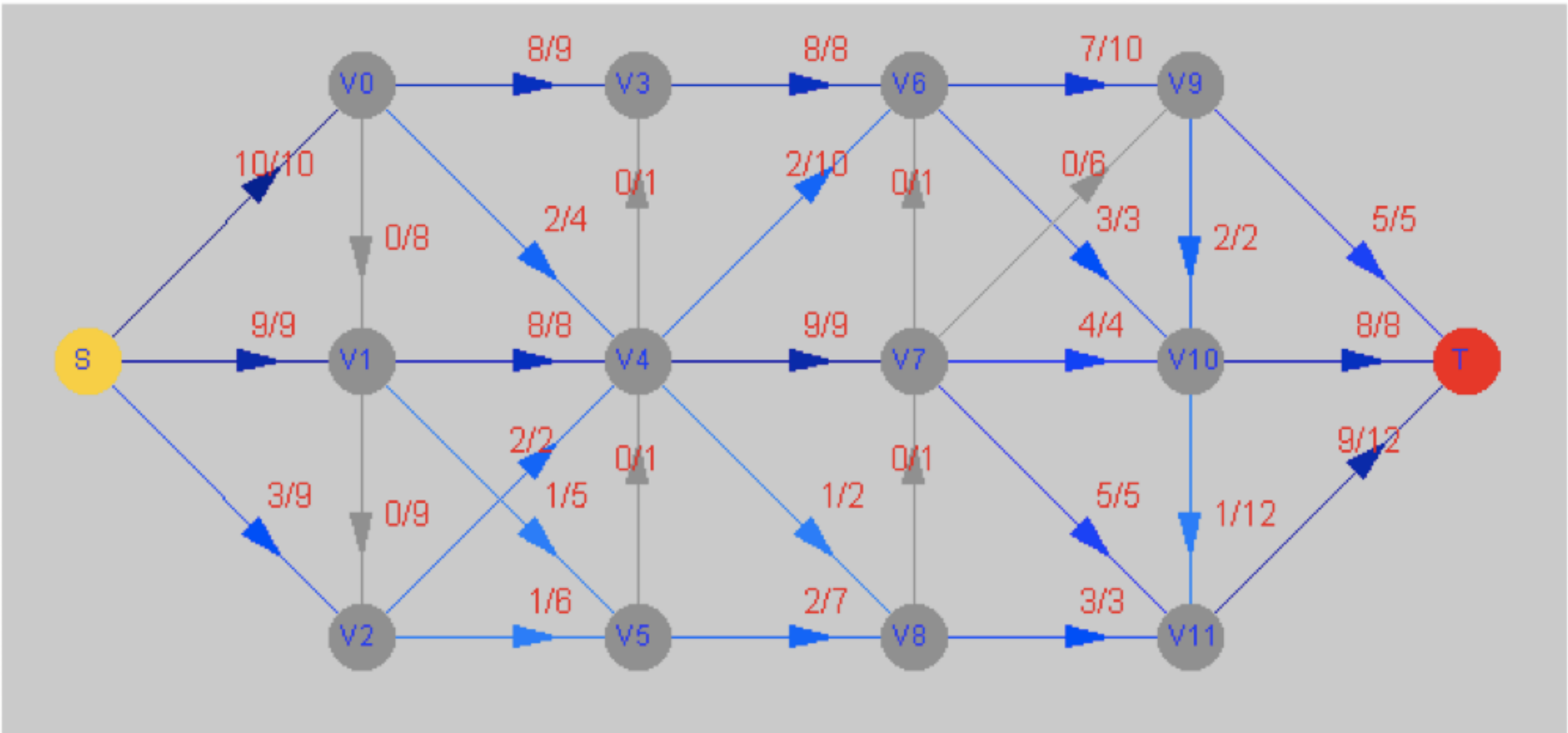
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

PATH AUGMENTATION



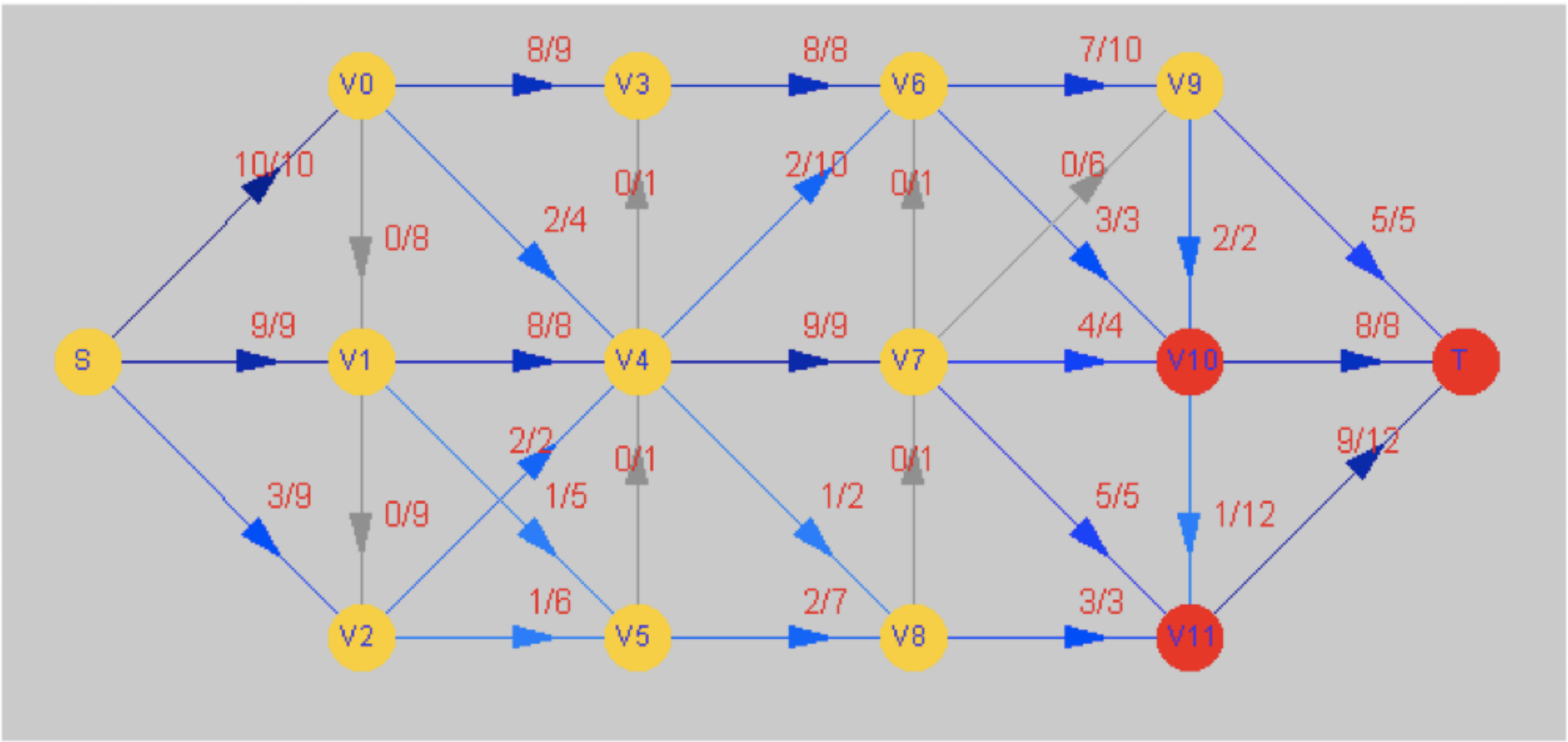
Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Path augmentation



Path augmentation was the first algorithm proposed. It works by finding paths from s to t and saturating the flow along.

Continuous Maximum flows

- Discrete maxflows can find L^1 minimal surfaces efficiently, but are biased by the grid. The result can show metrication artifacts.
- To avoid this, we seek a solution that is an L^2 minimal surface (a “proper”, Euclidean minimal surface)
- Contribution: algorithm.

Surface/flow duality

- Iri (1979) and Strang (1982) studied this problem in the continuous domain.

1. The graph is replaced by a field
2. The flow becomes a vector $\vec{F}$
3. Flow conservation is expressed by $\nabla.\vec{F} = 0$
4. Capacity constraints are expressed $|\vec{F}| \leq g$, with g a scalar field (tensor field formulation is possible).
5. Same concept of source and sink.

Flow through a surface

- Let S be as simple, regular, closed surface, with $\vec{N}$ the normal to the surface at each point. We write $\nabla.\vec{F}_s$ the total flow exiting the source. If S contains s, then

$$\nabla.\vec{F}_s = \int_S \vec{F}.\vec{N}\mathrm{d}s \leq \int_S g\mathrm{d}s$$

- All the surfaces S limit the flow from above.

- Equivalently, the g-weighted area for all surfaces S are limited from below by the flow exiting the source.

Continuous flows properties

- Every surface S is a bottleneck for the source flow, which must be less than the capacity, or weighted area of S.
- The maximum flow exiting s is limited by all possible surfaces, which must be less than the flow corresponding to the surface with minimal weighted area.
- Iri showed the existence of a maximal flow under very general conditions, that corresponds to the saturation of this minimal weighted surface.

Computing the flow

- Neither Strang nor Iri (nor others AFAWK) proposed a method for computing this maximum flow.

- We have proposed to compute the flow numerically. However in the continuous domain, it is not possible to consider numerically the set of all paths leading from s to t. Therefore simulating F&F or any path augmentation algorithm seems impossible.

- However, Push-Relabel (Golberg & Tarjan, 1987) algorithms are interesting to consider.

Push-relabel (G&T)

- Preflow-push propagation. A pre-flow is a relaxed form of a flow *F*, with the following characteristics :
    1. The flow entering an edge can be greater or equal to the exiting flow.
    2. An edge with non-zero difference is called *active*. The difference is called *excess*.
    3. The algorithm propagates the excess toward the sink, helped by a height function *H*, which depends on the shortest path from the edge to the sink via unsaturated edges.
    4. Source and sink have fixed height, with $H(t) = 0$. The sink is never active (it can receive an infinite flow).
- The algorithm is purely local in its operations.

Continuous propagation

- In continuous terms, one can assimilate the excess to a pressure P.
- To simulate a propagation, one an use an elementary fluid model, linking the pressure P and the speed F:

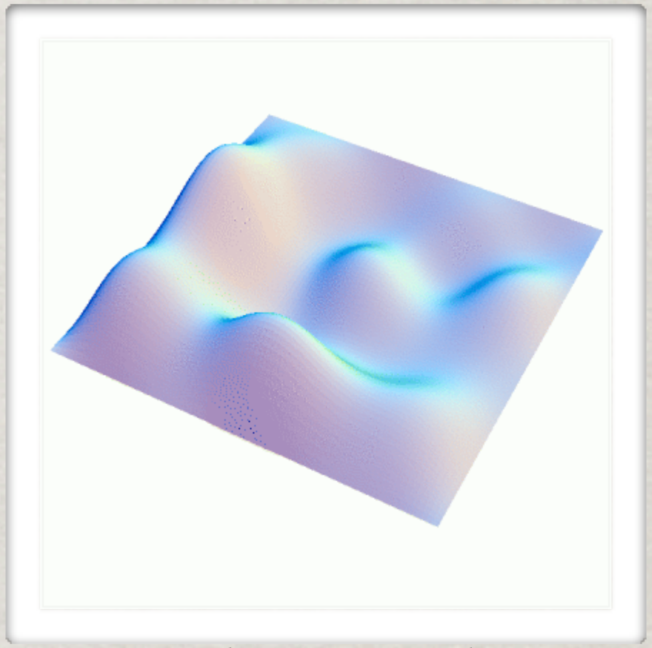
$$\frac{\partial P}{\partial t} = -\nabla.\vec{F}$$

$$\frac{\partial \vec{F}}{\partial t} = -\nabla P$$

Constraint and propagation

- By derivation one gets a standard wave equation :

$$\nabla^2 \vec{F} - \frac{\partial^2 \vec{F}}{\partial t^2} = 0$$



- To simulate constraints, we introduce the field g, such that :

$$\|\vec{F}\|_2 \leq g$$

Limit conditions are : $s \equiv 1$, $t \equiv 0$

System properties

- As in Push-Relabel, F is no longer incompressible: $\nabla.\vec{F} \neq 0$
- The field P represents the excess.
- We can show that P is conserved in any region that does not contain s or t.
- We can show that the system is dissipative : oscillations must vanish and the system converges to a steady state.

Conservation of potential

- Let A be a sourceless region (neither including s nor t).

$$
\begin{aligned}
\frac{\partial P_A}{\partial t} &= \int_A \frac{\partial P}{\partial t} \mathrm{d}A \\
&= -\int_A \mathrm{div}\vec{F} \mathrm{d}A \\
&= -\oint_{\partial A} \vec{F} \cdot \vec{N}_{\partial A} \mathrm{d}\left(\partial A\right). \\
&= 0
\end{aligned}
$$

- P is conserved in any such regions.

System at convergence

- At convergence, temporal derivatives vanish, so the system becomes :

$$\nabla.\vec{F} = 0,$$

$$\nabla P = \begin{cases} 0 & \text{if} \|\vec{F}\|_2 < g \\ -\lambda \vec{F} \qquad \text{with } \lambda \geq 0 & \text{if} \|\vec{F}\|_2 = g \end{cases}$$

- The simulated fluid become incompressible. The flow lines are aligned with the pressure. P is constant wherever F is not saturated. P decreases monotonically from s to t.

System at convergence

- P is monotonically decreasing because, wherever P is defined,
$$\nabla P \cdot \vec{F} \leq 0.$$

- In an incompressible fluid, flow lines can only originate at the source and terminate at the sink, therefore P has no local extremum.

- We can consider any region A defined by thresholding P.

At convergence (II)

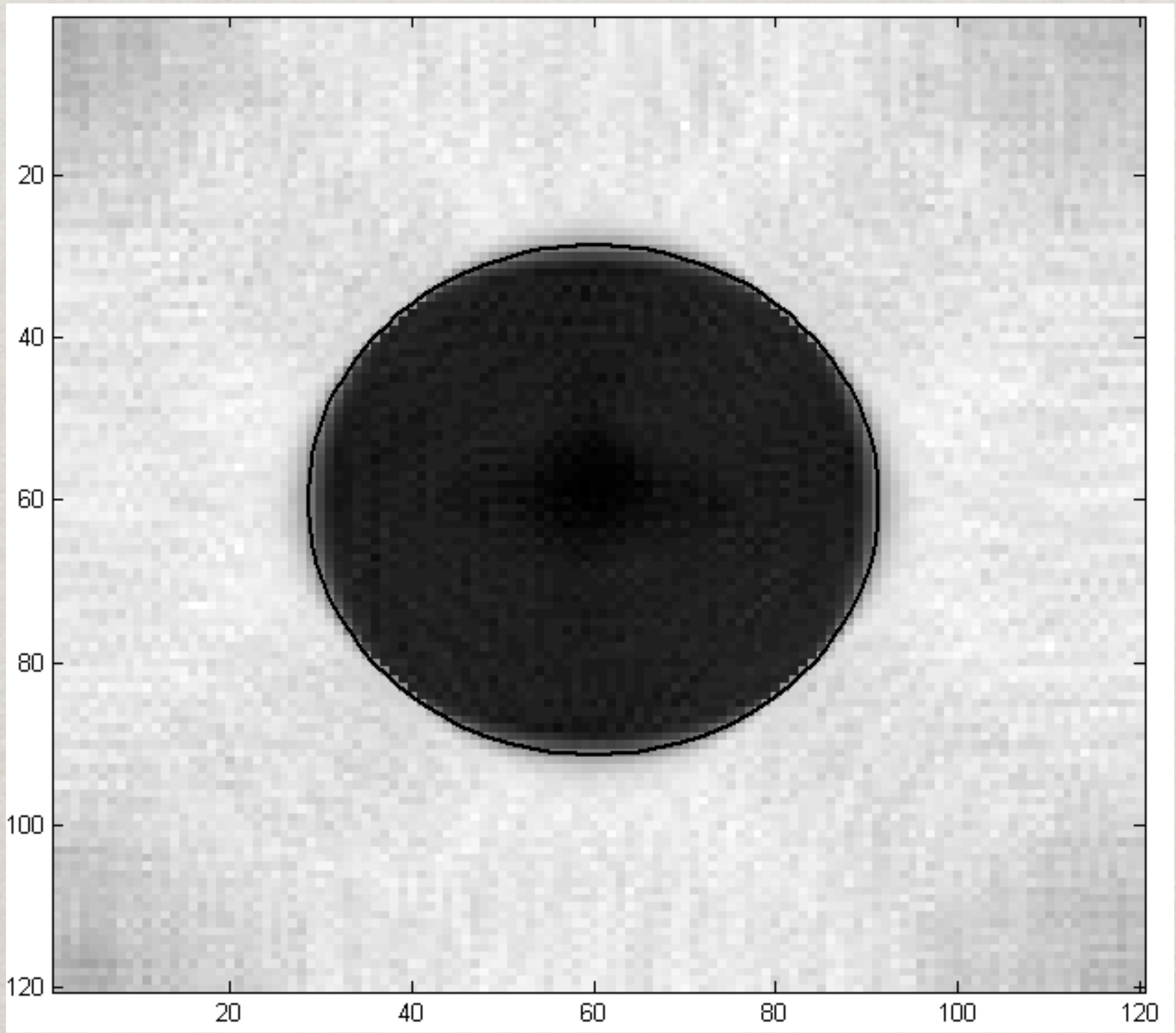
- At convergence, we can consider any region where $P \geq p > 0$, (p constant). Due to monotonicity, this region contains s, and the border of S is saturated because P is non-zero.

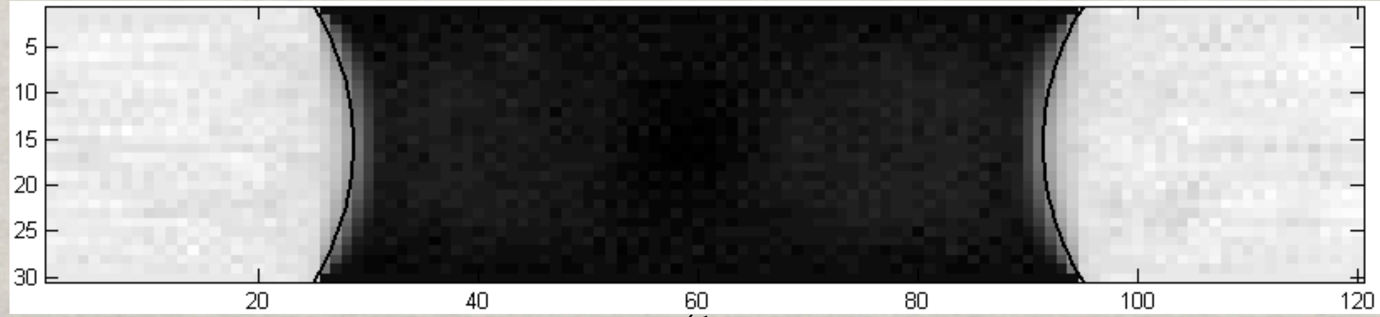
- On the border of S, we have:

$$\nabla.\vec{F}_s = \int_S \vec{F}.\vec{N}\mathrm{d}s = \int_S g\mathrm{d}s$$

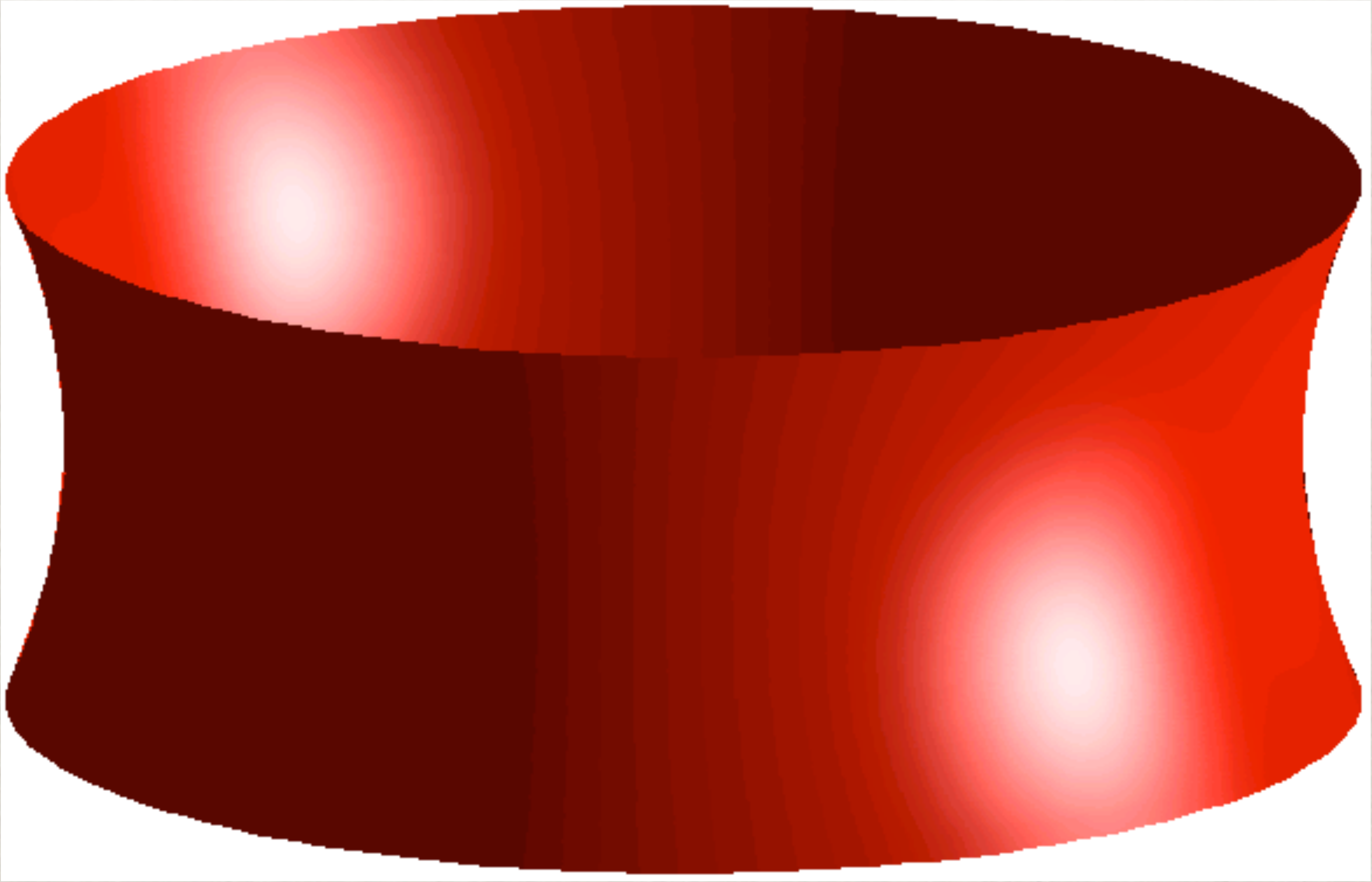
- We have found both the maxflow and the minimal surface.

Back to the soap bubbles

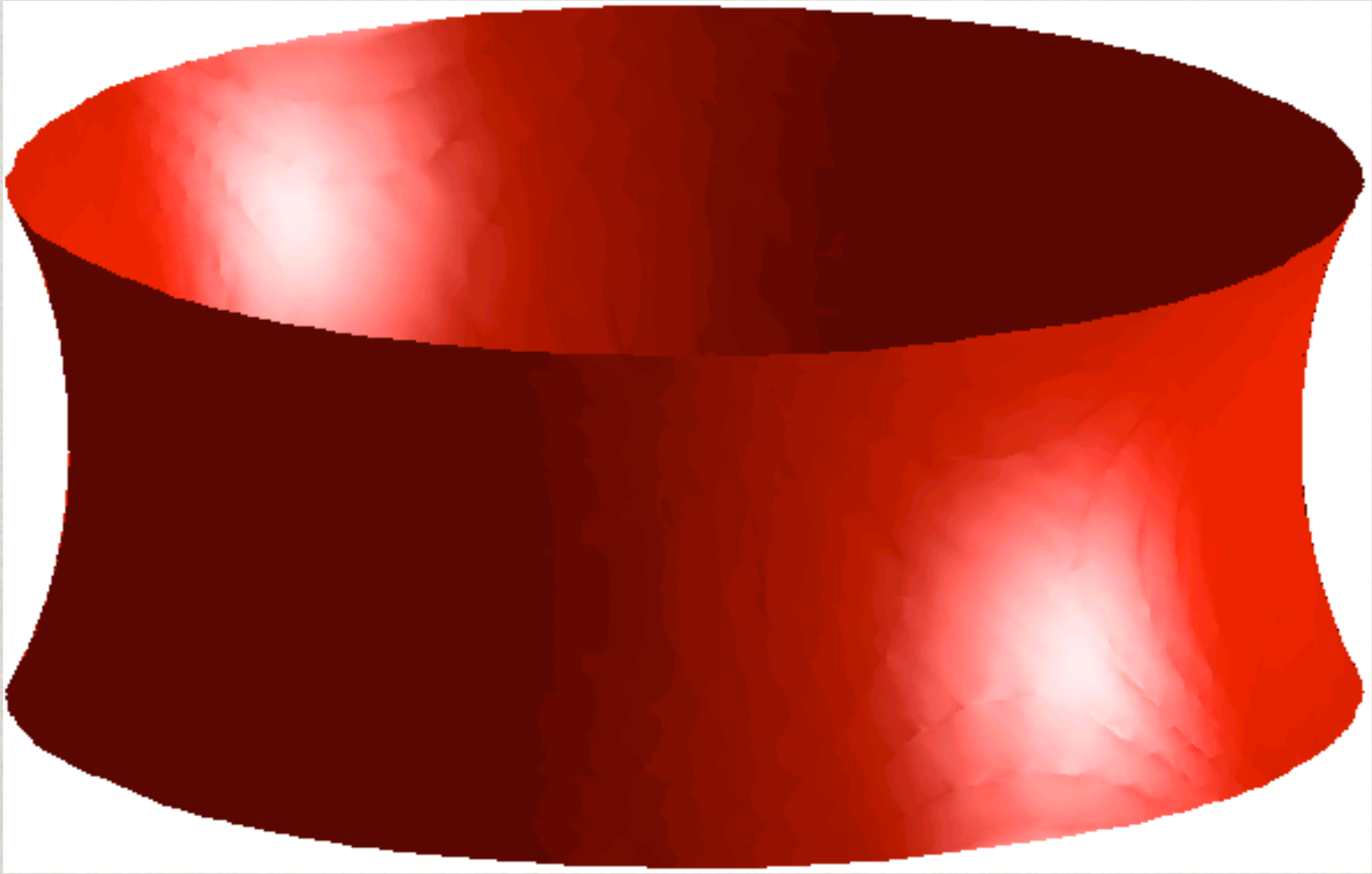




Analytic solution

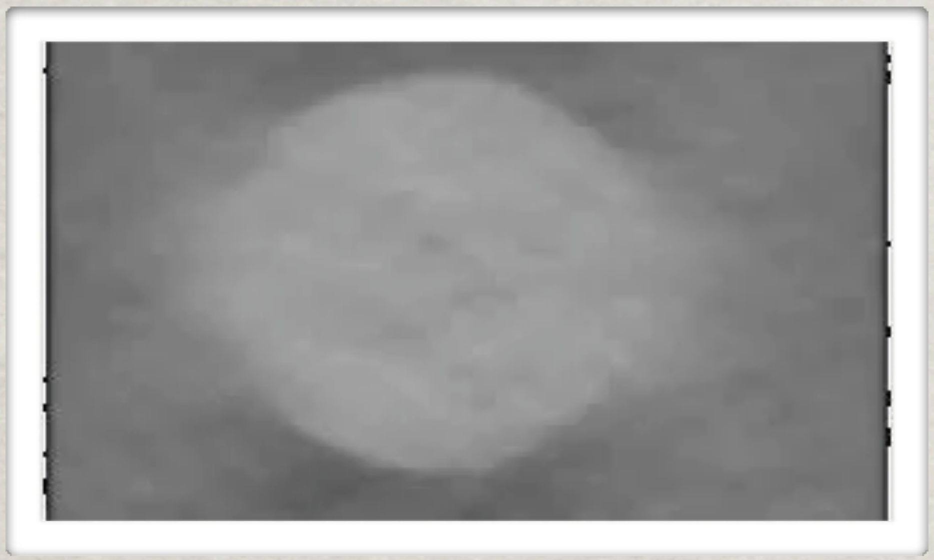


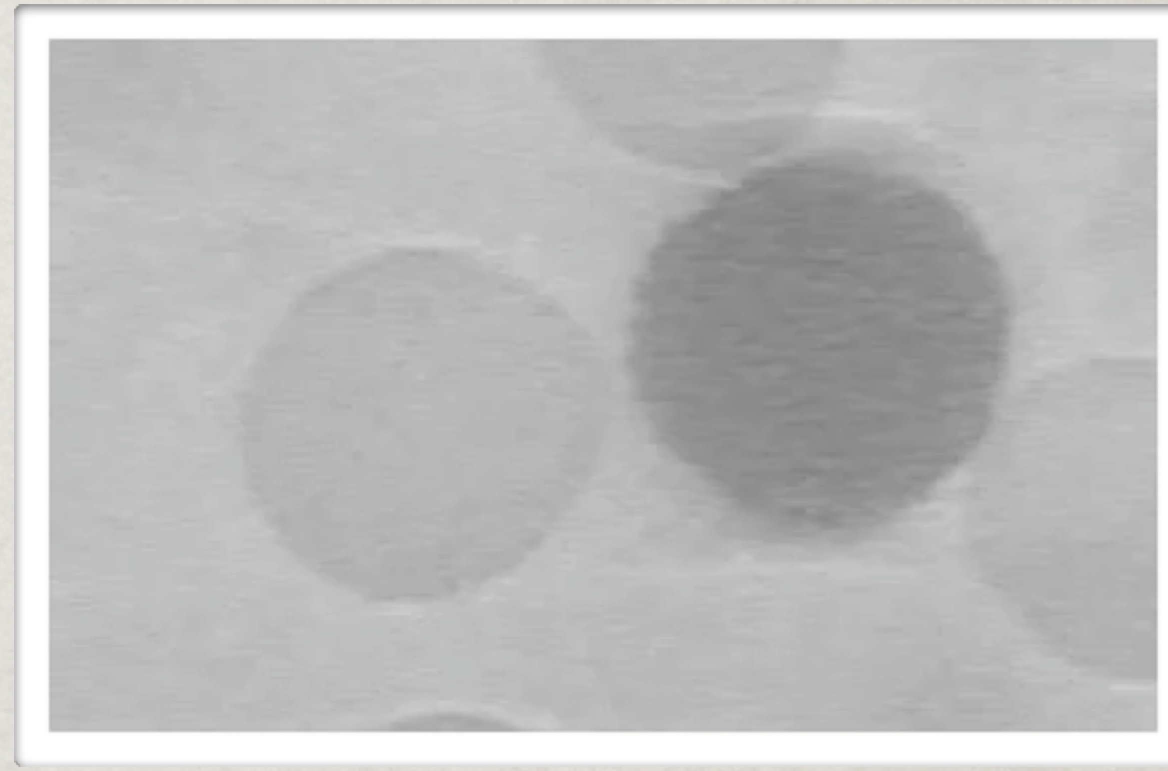
CMF solution

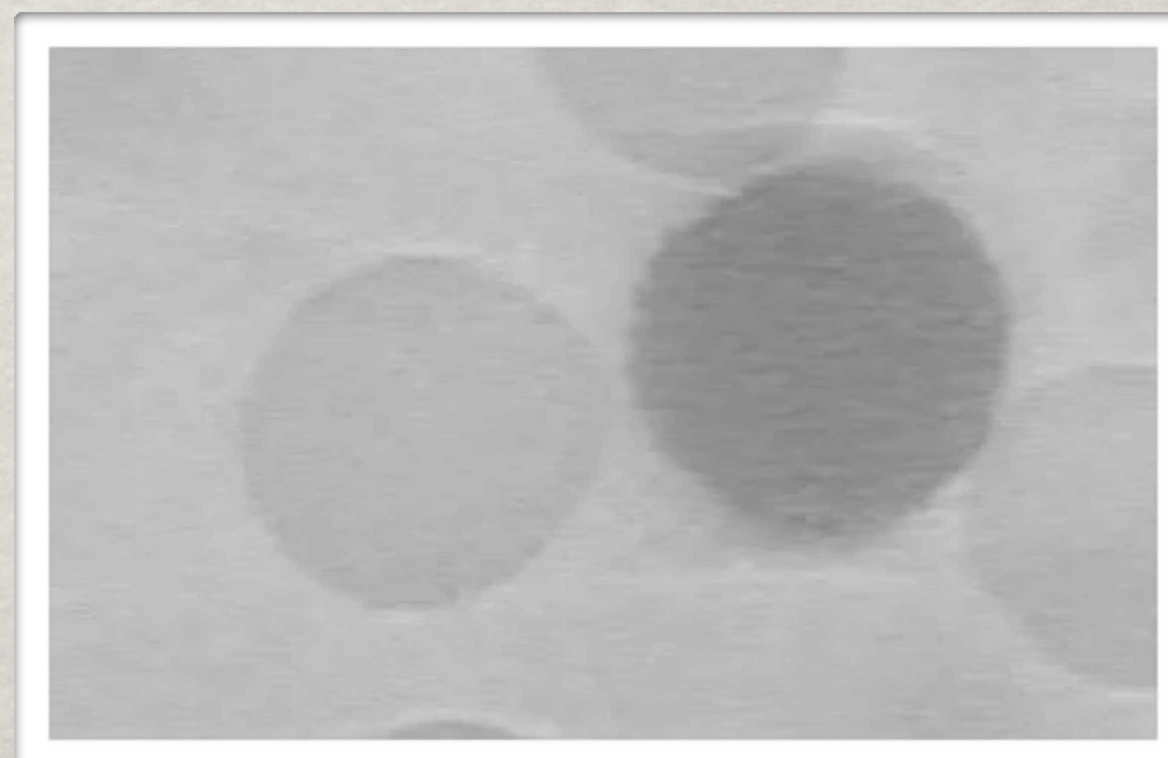


Mean error ~ 0.1 pixel.

Illustrations







Bias toward small regions

- Due to the nature of the equation we minimize, small regions are favoured.
- A solution is to weigh the metric

Issues

- Convergence
- Complexity
- Difficult cases
- Implementations
- Stability

Convergence

- Reformulate (Chambolle)

$$\begin{aligned}\frac{\partial P}{\partial t} &= -\nabla.\vec{F}\\ \frac{\partial \vec{F}}{\partial t} &= -\nabla P - \partial_P G^*(x,P)\end{aligned}$$

- Energy :

$$\epsilon(t) = \frac{1}{2}\int_\Omega |\frac{\partial P}{\partial t}|^2 + |\mathrm{div} P|^2 \mathrm{d}x$$

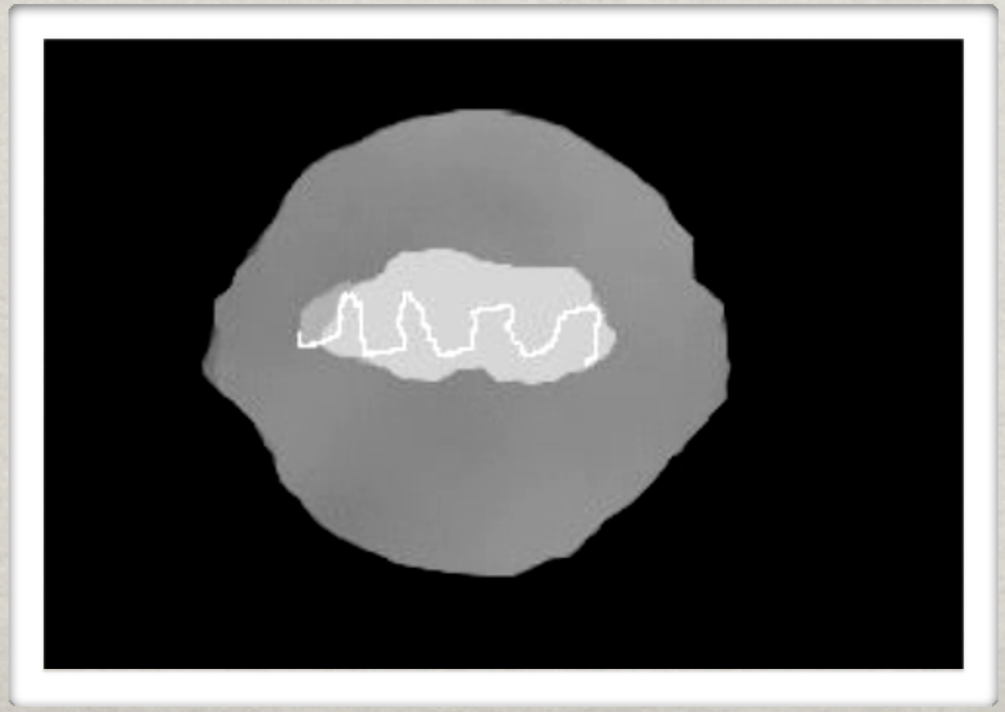
- Lyapunov :

$$\epsilon'(t) = -\int_\Omega \left(\partial^2_{PP} G^*(x,P)\frac{\partial P}{\partial t}, \frac{\partial P}{\partial t}\right)\mathrm{d}x$$

- This term is always negative until oscillation dampen.

Complexity

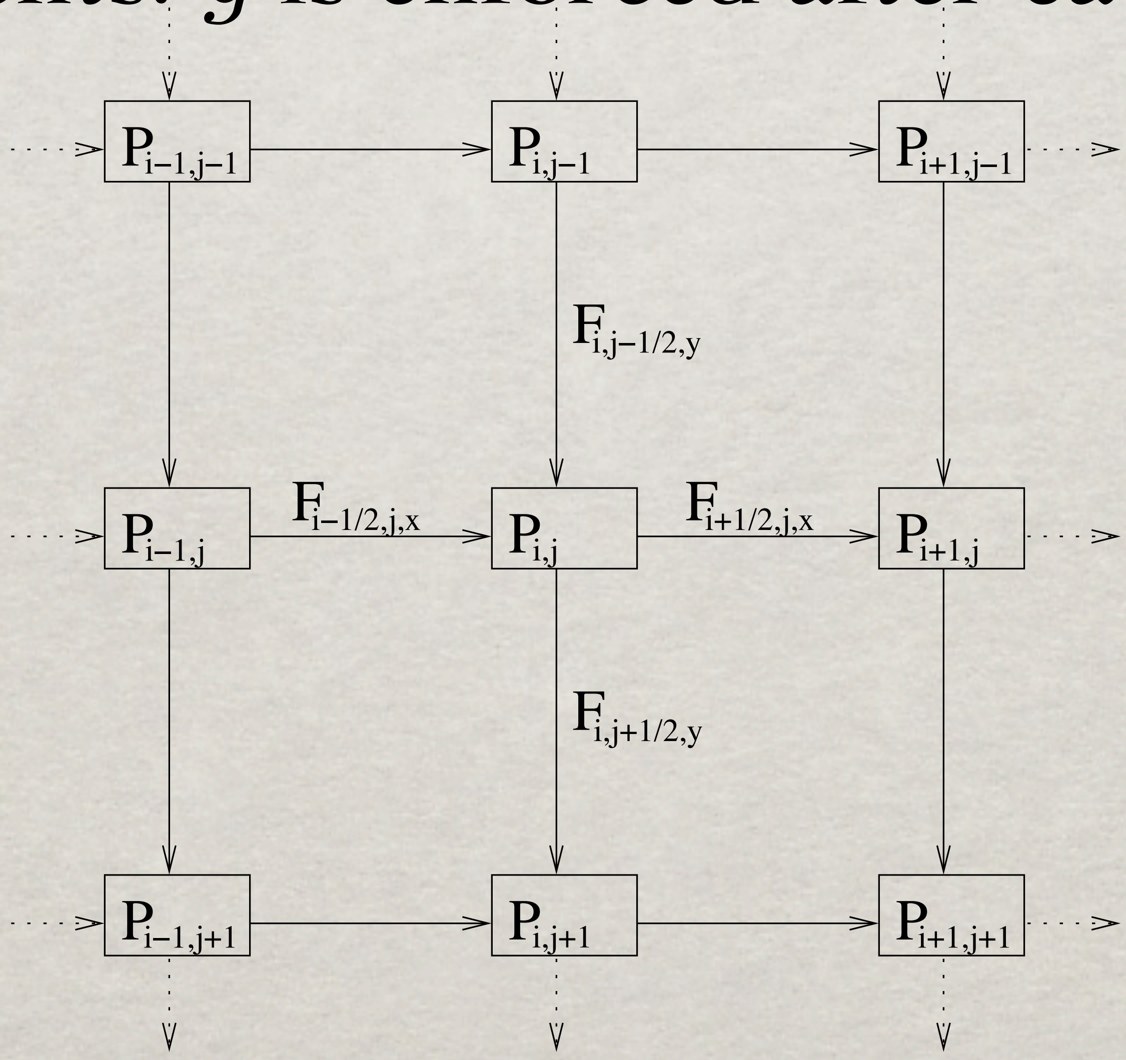
- Difficult to express. Oscillation dampen reasonably quickly in a small number of passes ~ $5 \times \max(nx, ny, nz)$
- This works out to $O(n^{3/2})$ in 2D, however convergence may not necessarily be achieved after dampening.



Implementation

- Finite differences implemented on a staggered grid, explicit in time and space.

- P is stored on nodes and F on edges, by components. g is enforced after each timestep.



FINITE DIFFERENCES SCHEME

- Conservation equation

$$P^{n+1}_{i,j} = P^{n}_{i,j} - \Delta t \left((F^{n}_{i+\frac{1}{2},j,x} - F^{n}_{i-\frac{1}{2},j,x}) + (F^{n}_{i,j+\frac{1}{2},y} - F^{n}_{i,j-\frac{1}{2},y}) \right),$$

- Driving equation

$$\begin{aligned} F'^{n+1}_{i+\frac{1}{2},j,x} &= F^{n}_{i+\frac{1}{2},j,x} - \Delta t(P^{n+1}_{i+1,j} - P^{n+1}_{i,j}) \\ F'^{n+1}_{i,j+\frac{1}{2},y} &= F^{n}_{i,j+\frac{1}{2},y} - \Delta t(P^{n+1}_{i,j+1} - P^{n+1}_{i,j}). \end{aligned}$$

- The magnitude constraint is applied immediately following the update of the flow field

Implementation (II)

- Implementation is simple, but making it fast is hard(ish) :

  1. Keep a tight inner loop : decompose the problem into runlengths (cache reasons)

  2. Parallel implementation is readily feasible.

- We have two implementations (C/C++) with similar performances. One is completely free (contact us).

- Performance similar to GC:
  3D image 256^3 30s on Core Duo 2GHz

Stability

- The scheme is explicit, describes an hyperbolic wave equation with constraints.
- Ignoring the constraints, stability is obtained by a CFL condition:

$$\Delta t \leq \frac{1}{\sqrt{N}}$$

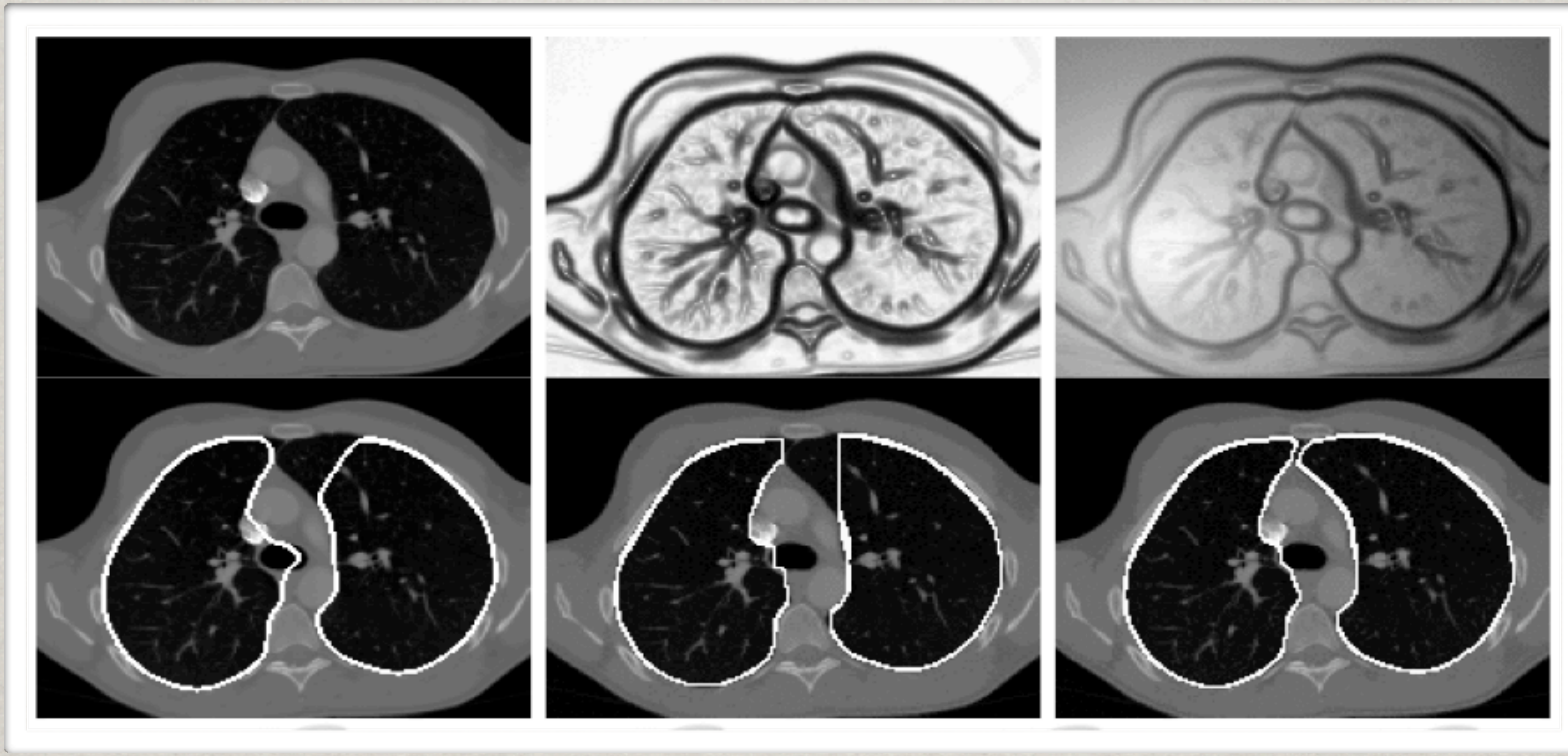
- N is the number of dimension
- The constraint does not perturb the condition as it does not change the wave velocity.

Applications

- Medical imaging
- Materials science

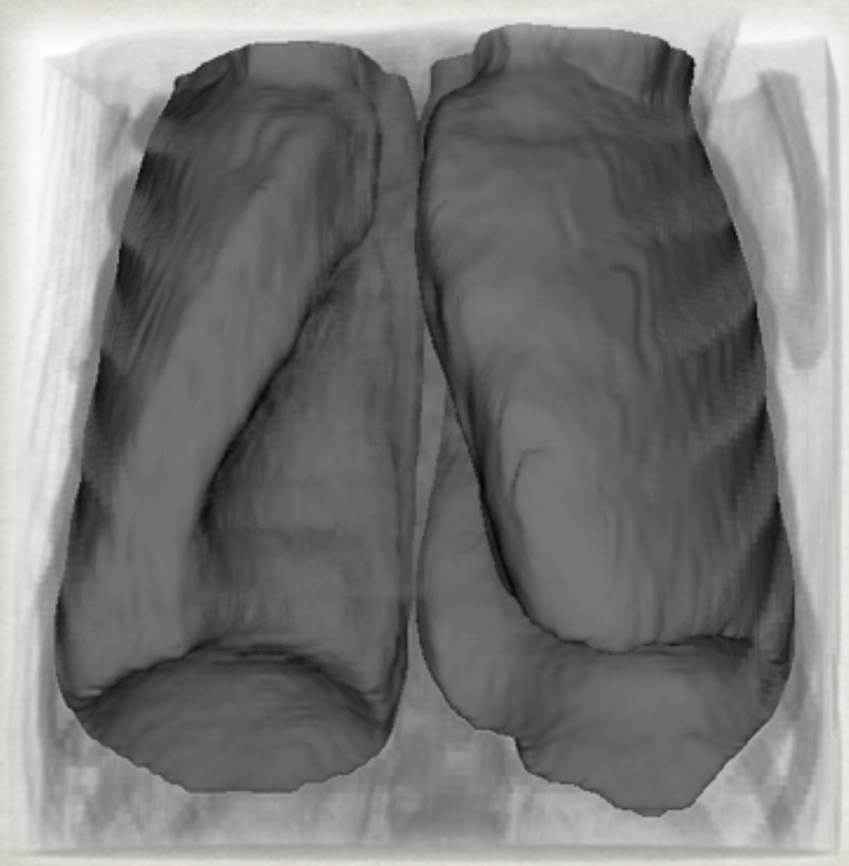
Lung segmentation

Original Gradient Metric

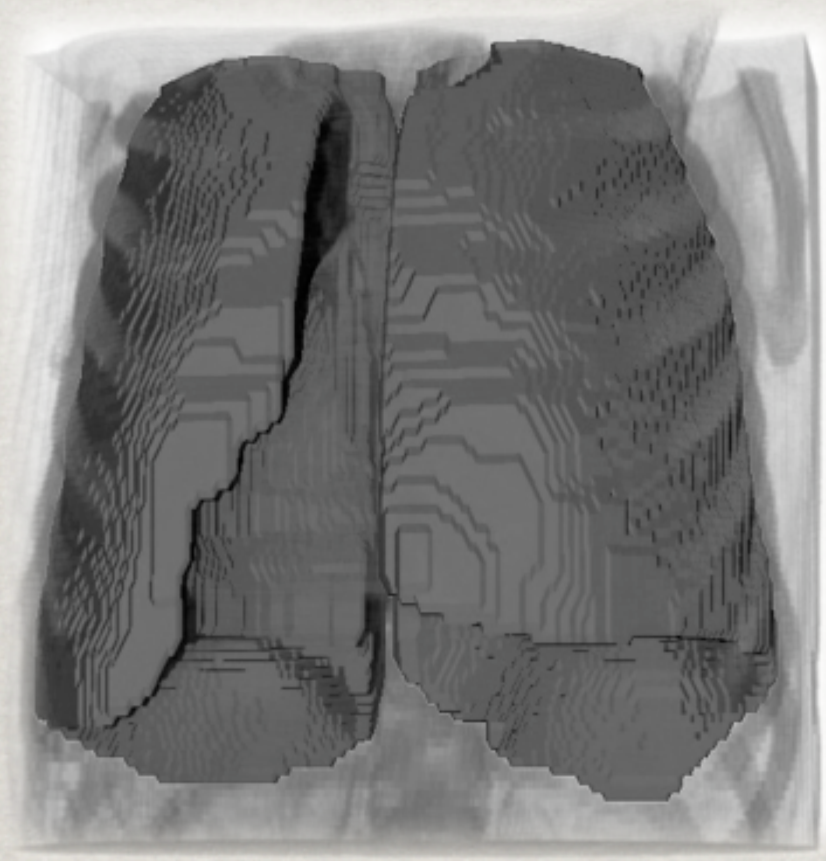


GAC Graph Cut CMF

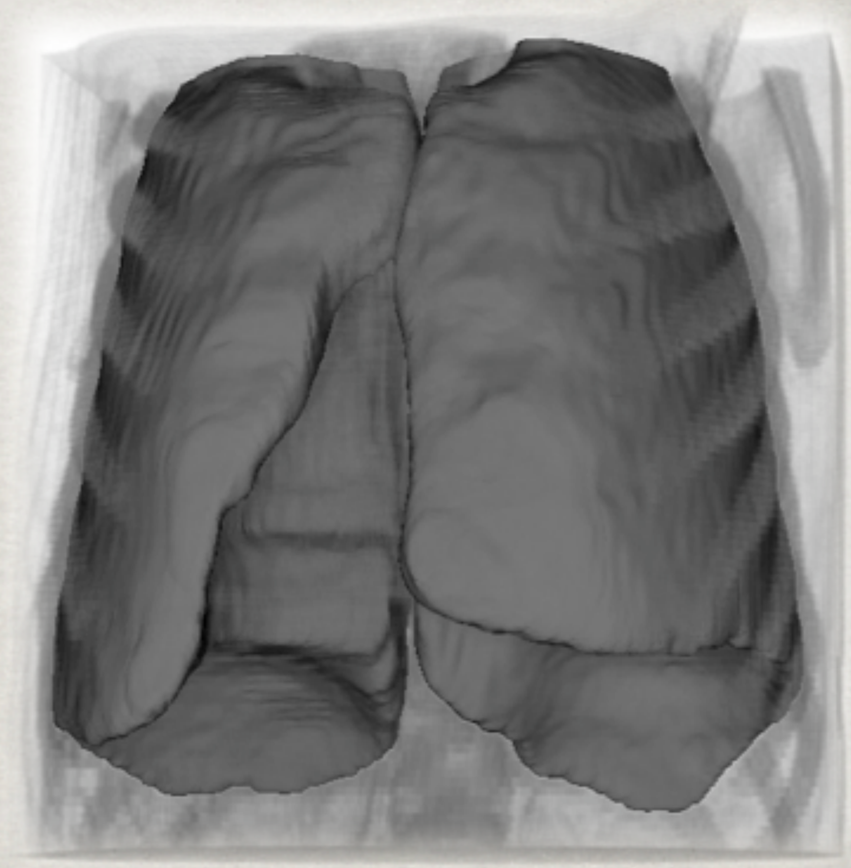
Medical imaging



Iterative variationnal (GAC)

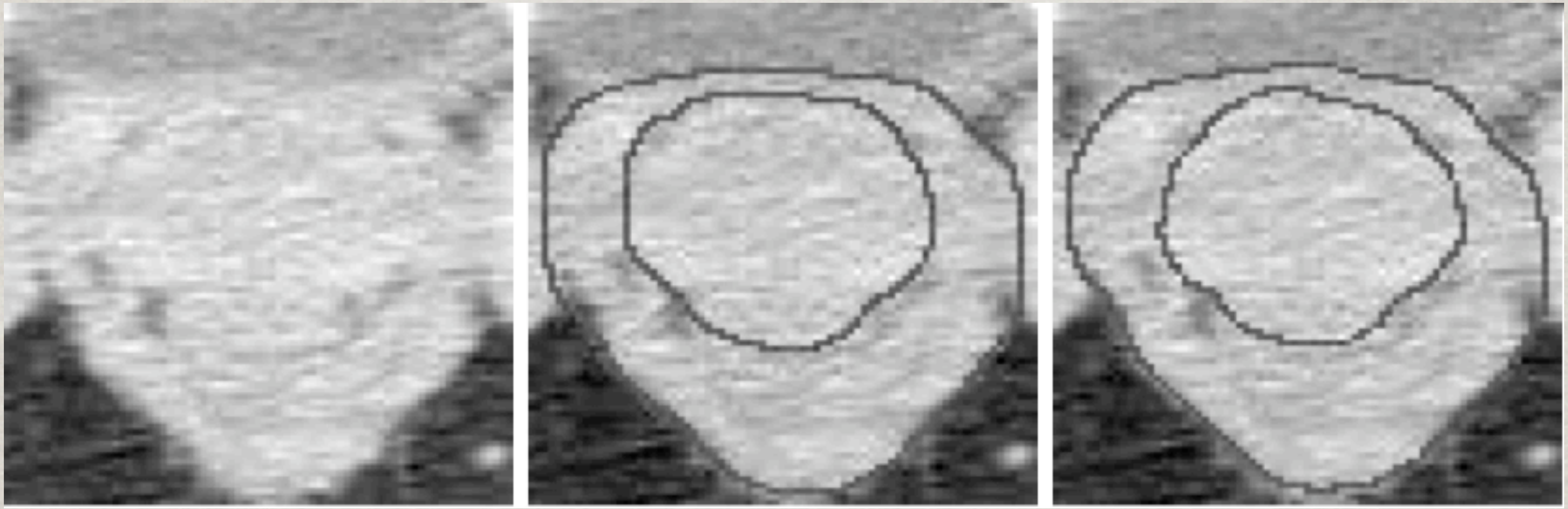


Discrete Graph cut



Continuous max flow

Ultrasound prostate Segmentation

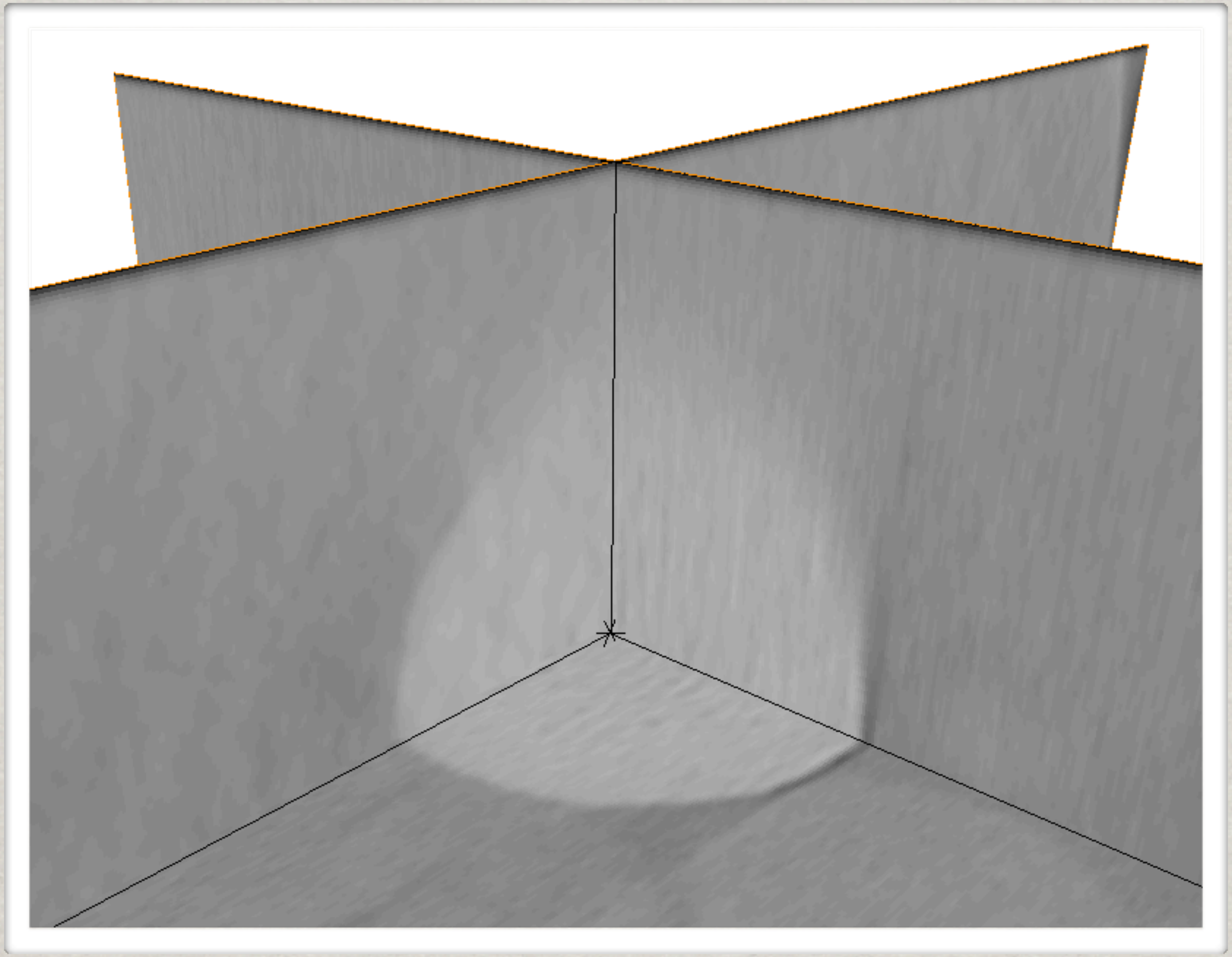


Original

CMF

CMF with tensor constraint

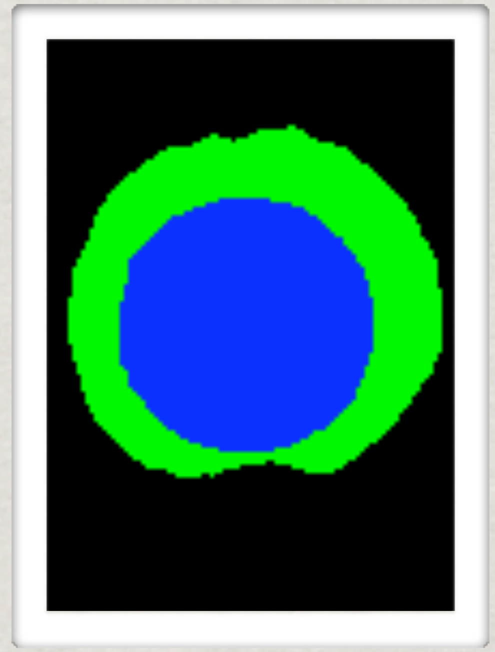
Materials



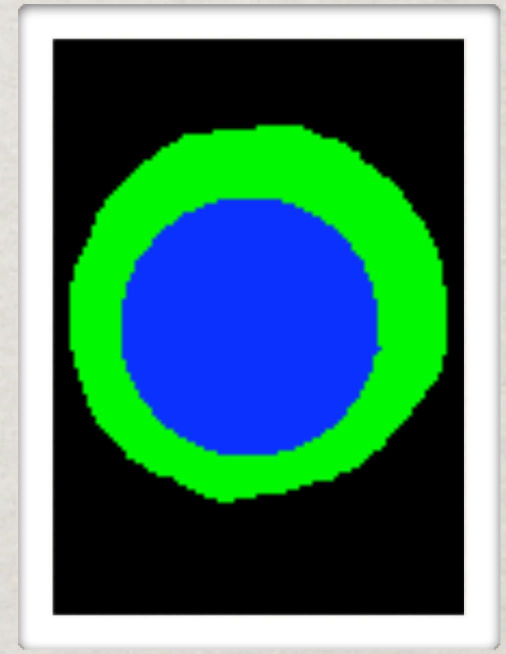
Electron nanotomography, results in anisotropic 3D image acquisition

Materials

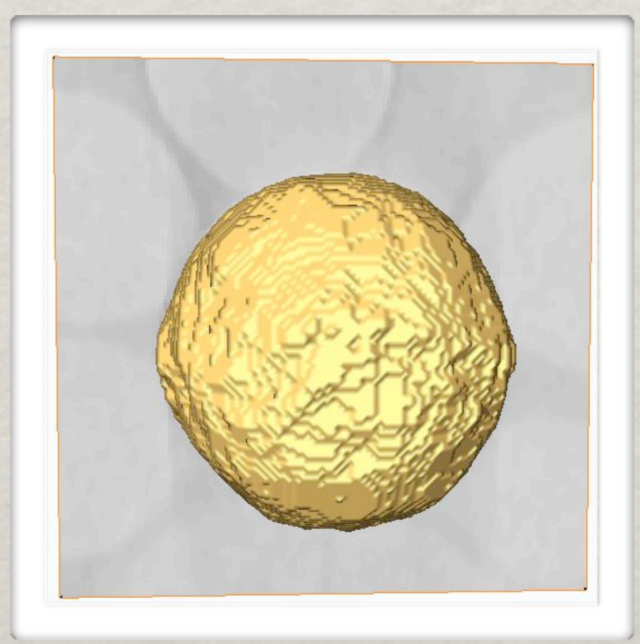
Original



Without shape prior



With shape prior



3D result

Conclusion

- We have proposed a new *continuous* formulation of maxflow/mincut.
- The result is an explicit numerical scheme that computes a binary indicator function, the isolevels of which correspond to minimal surfaces.
- The result is globally optimal. Compared to GCs, metrication errors are limited.
- Optimality does not solve the problem of segmentation.