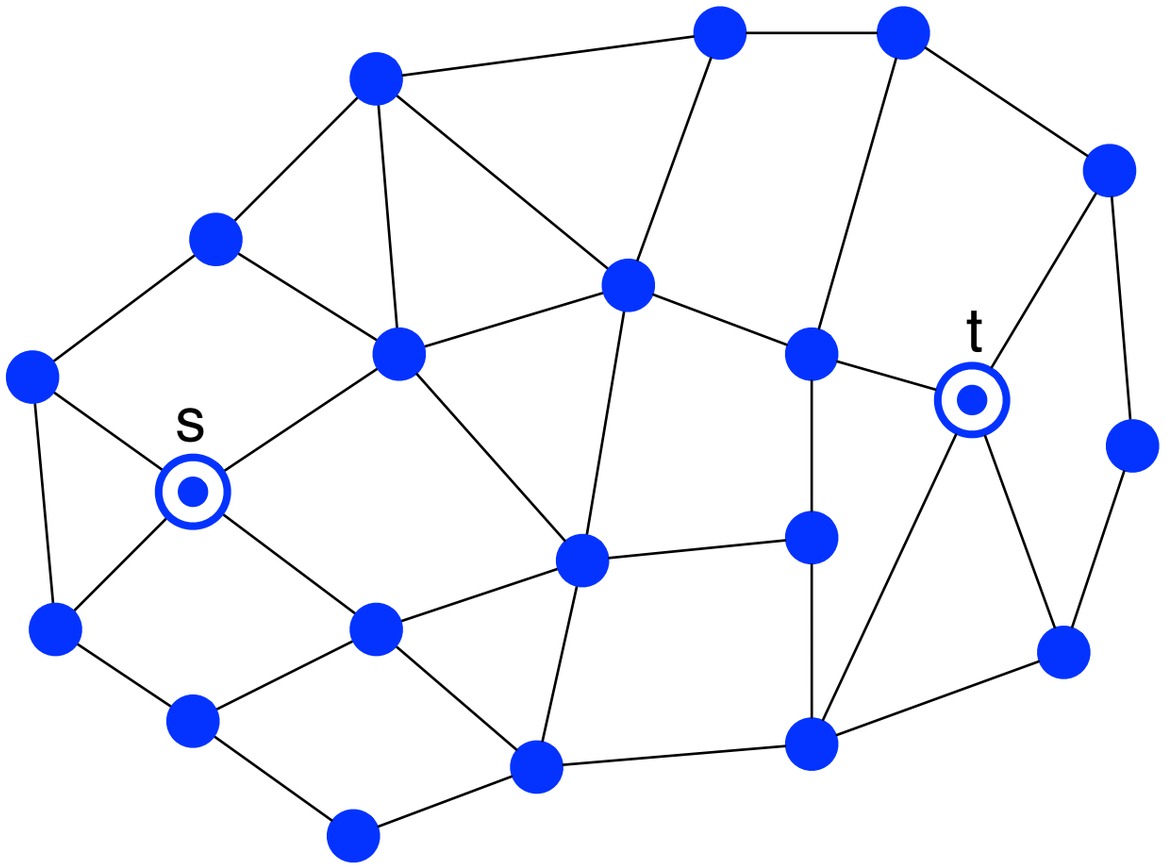
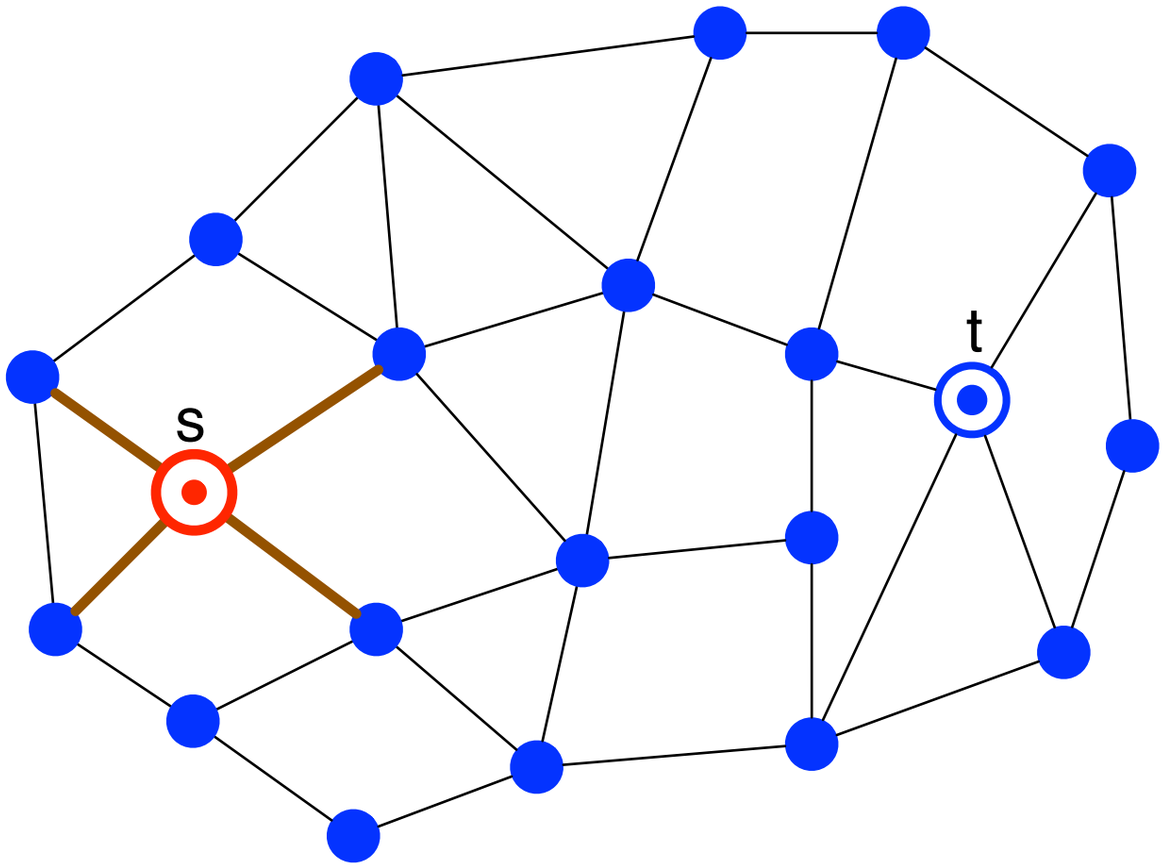
Counting and Sampling Minimum Cuts in Genus g Graphs

Kyle Fox
with Erin W. Chambers and Amir Nayyeri

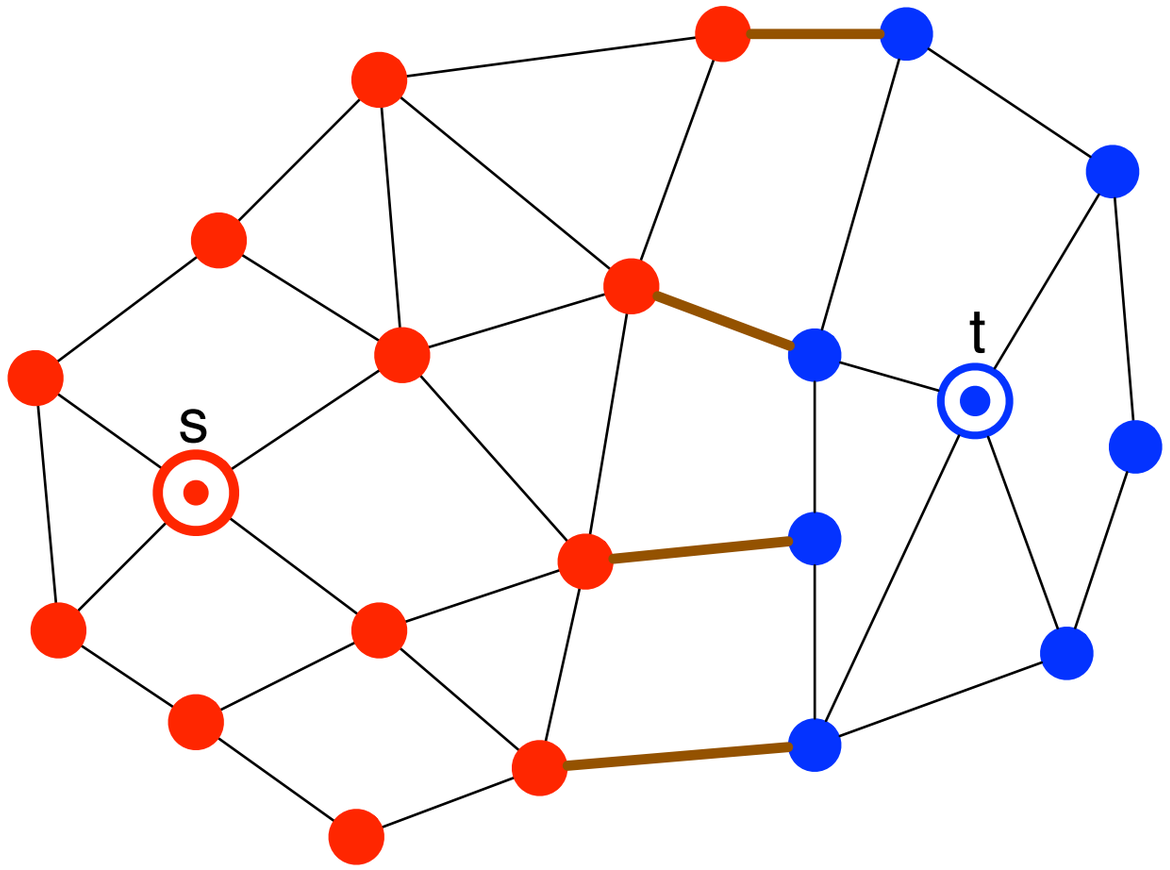
Counting Minimum Cuts



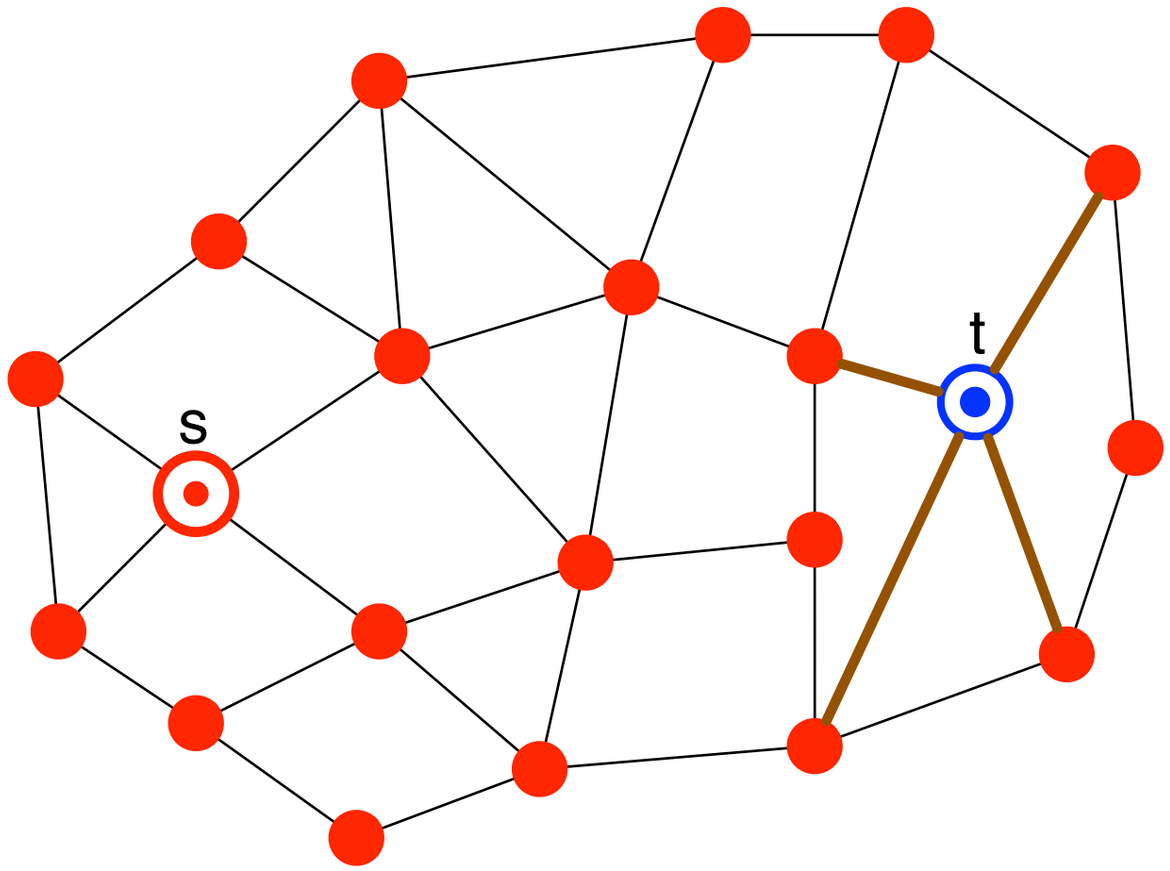
Counting Minimum Cuts



Counting Minimum Cuts



Counting Minimum Cuts



Counting Minimum Cuts

- Fundamental question to ask; easy to find, hard to count
- Applications to stochastic network reliability and image segmentation
- Counting algorithms can be used for sampling

Complexity of Counting

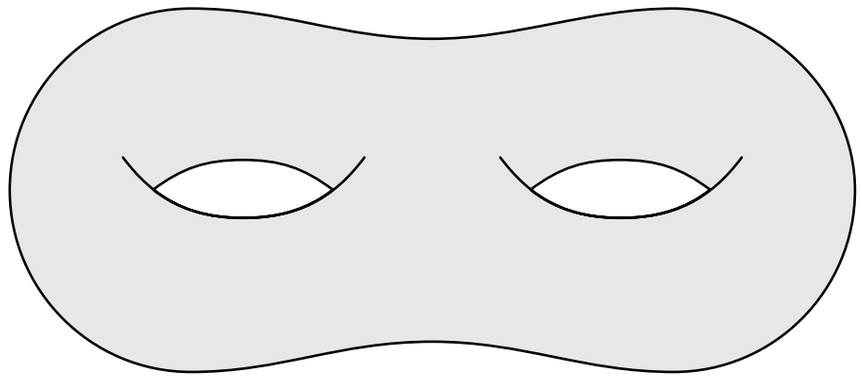
- Counting minimum cardinality s,t-cuts is #P-complete [Provan, Ball '83]
- Can count minimum cardinality s,t-cuts in $O(n^2)$ time in s,t-planar graphs [Ball, Provan '83]
- Can count or sample minimum weight s,t-cuts $O(n^2)$ time in general planar graphs [Bezáková, Friedlander '12]

Planarity Helps

- Many problems can be solved more quickly in planar graphs
- Minimum spanning trees, single-source shortest paths, graph isomorphism, maximum flows and minimum cuts
- Results usually generalize

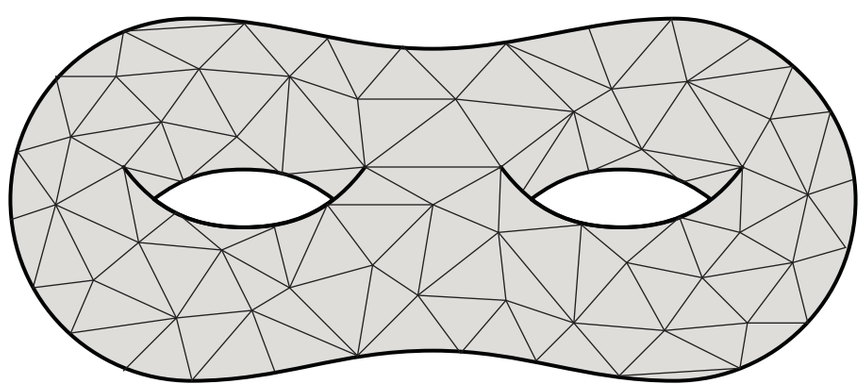
Surfaces

- 2-manifolds (with b boundary components)
- ***genus g***: max # of disjoint simple cycles whose compliment is connected
  = number of holes
  = number of handles attached to sphere



Surface Graphs

- Generalizes planar graphs
- *Most* planar results generalize easily
- Cuts and flows only recently [Chambers, Erickson, Nayyeri STOC/SOCG '09; Italiano et al. '11, Erickson, F, Nayyeri '12]

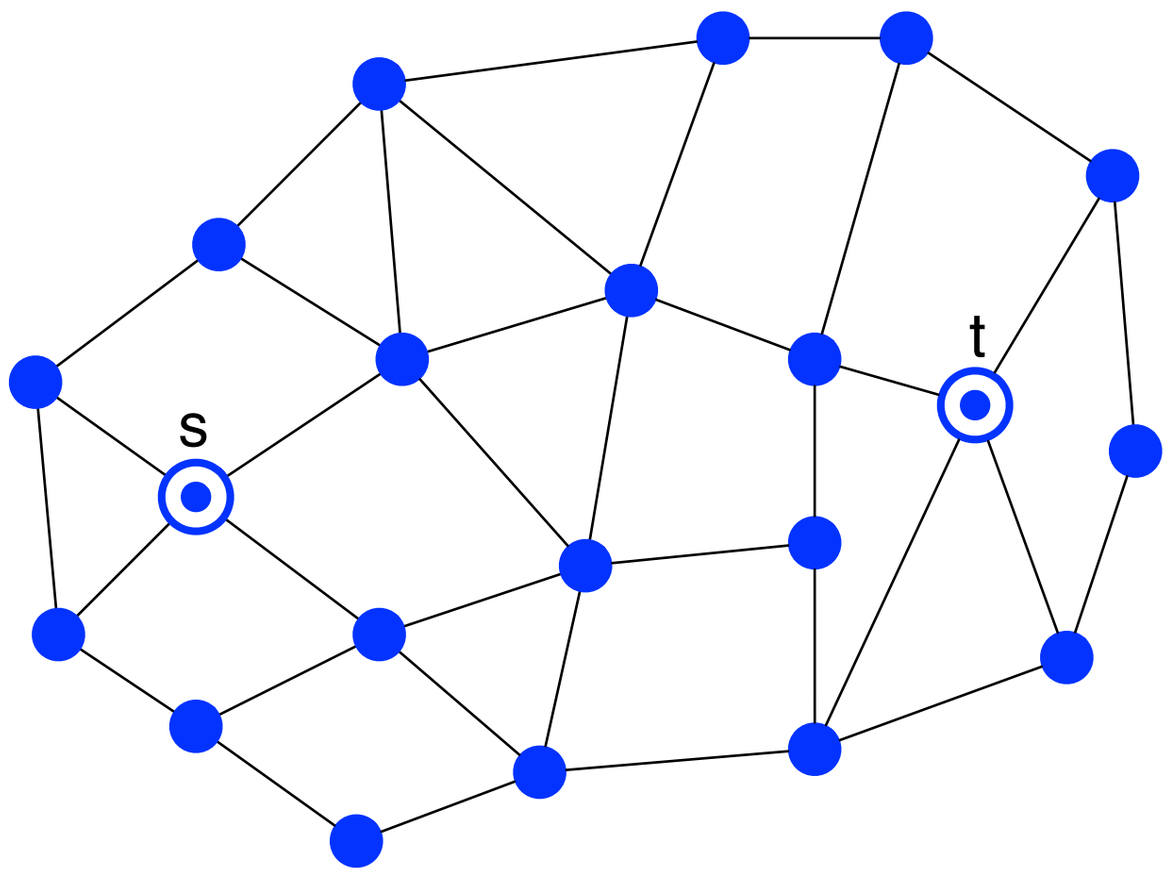


Our Results

- An algorithm to count minimum weight s,t-cuts in directed graphs in $2^{O(g)}\, n^2$ time
- After counting, can sample in $O(g^2\, n)$ time each
- Only assumes positive edge lengths and every point lying on an s to t walk

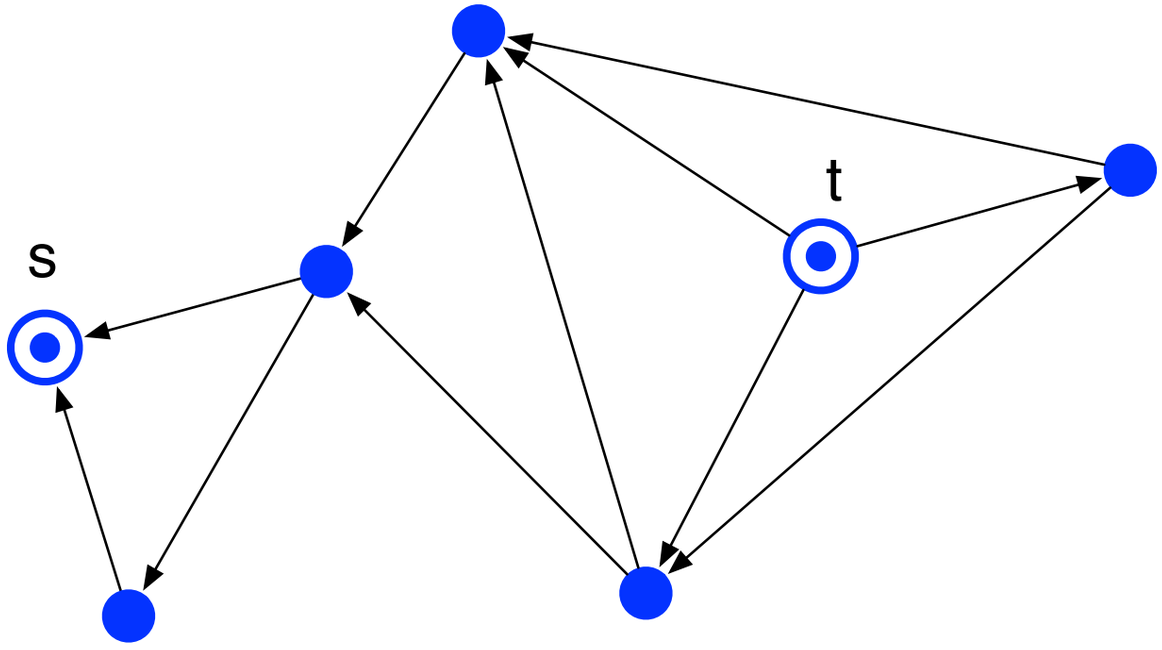
General Counting Idea

- Begin by computing a maximum *s*,*t*-flow
- Contract all cycles in the residual graph



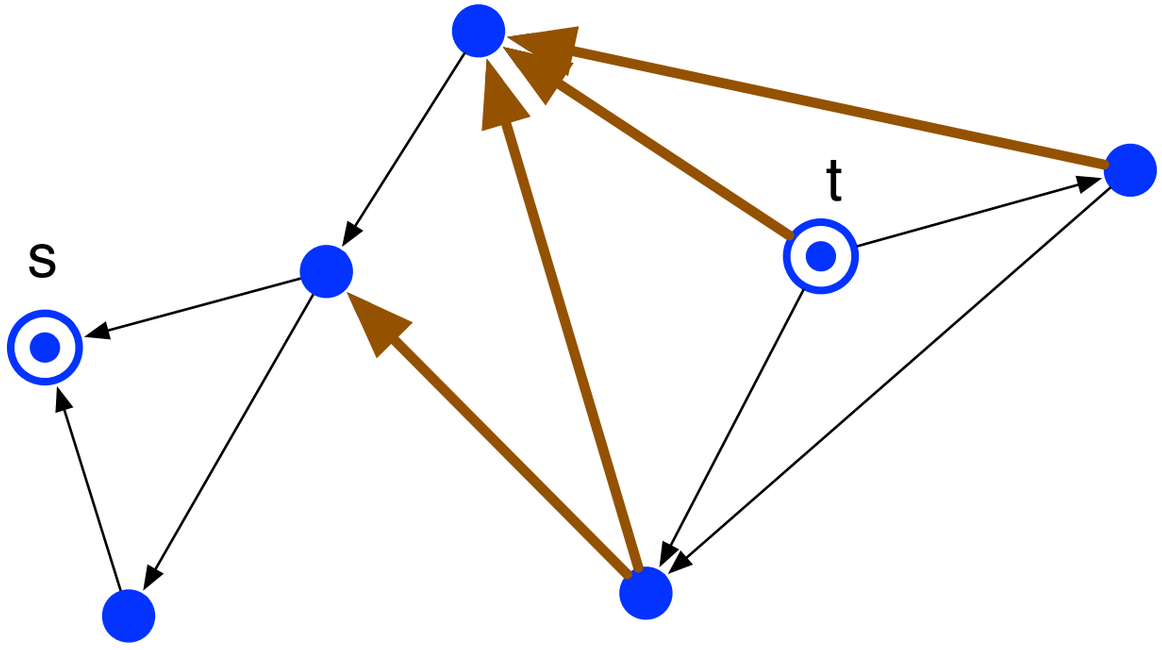
General Counting Idea

- Results in a directed acyclic graph with source *t* and sink *s*



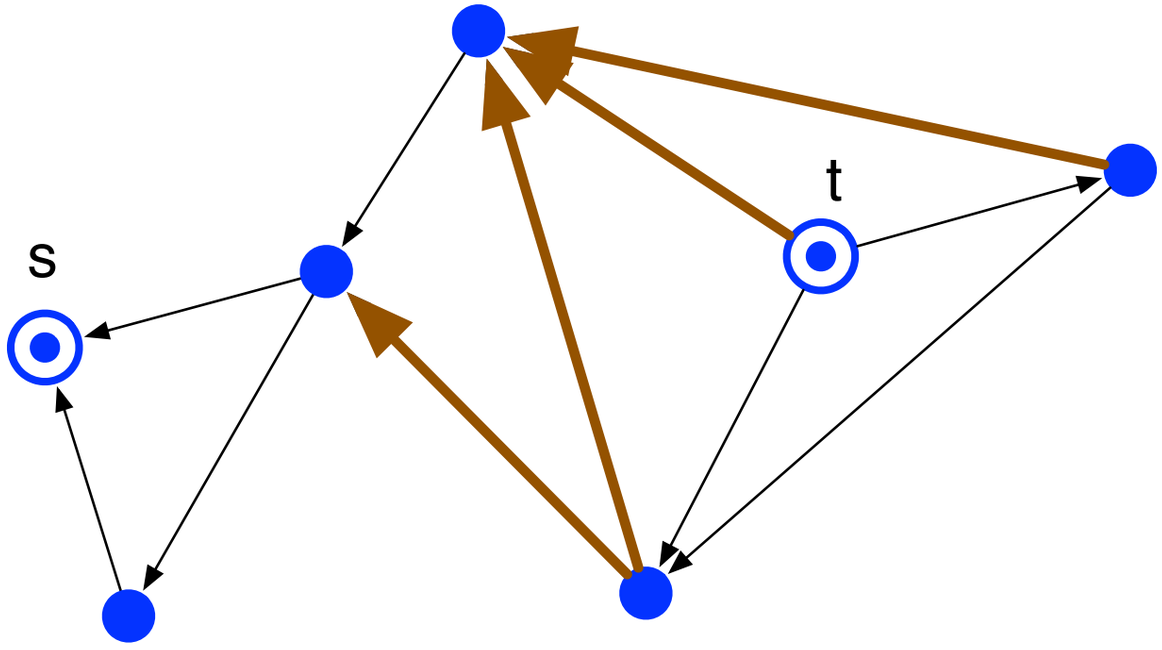
Forward Cuts

- A **forward *t*,*s*-cut** partitions vertices into sets T and S with $t \in T$, $s \in S$, and no edges going from T to S



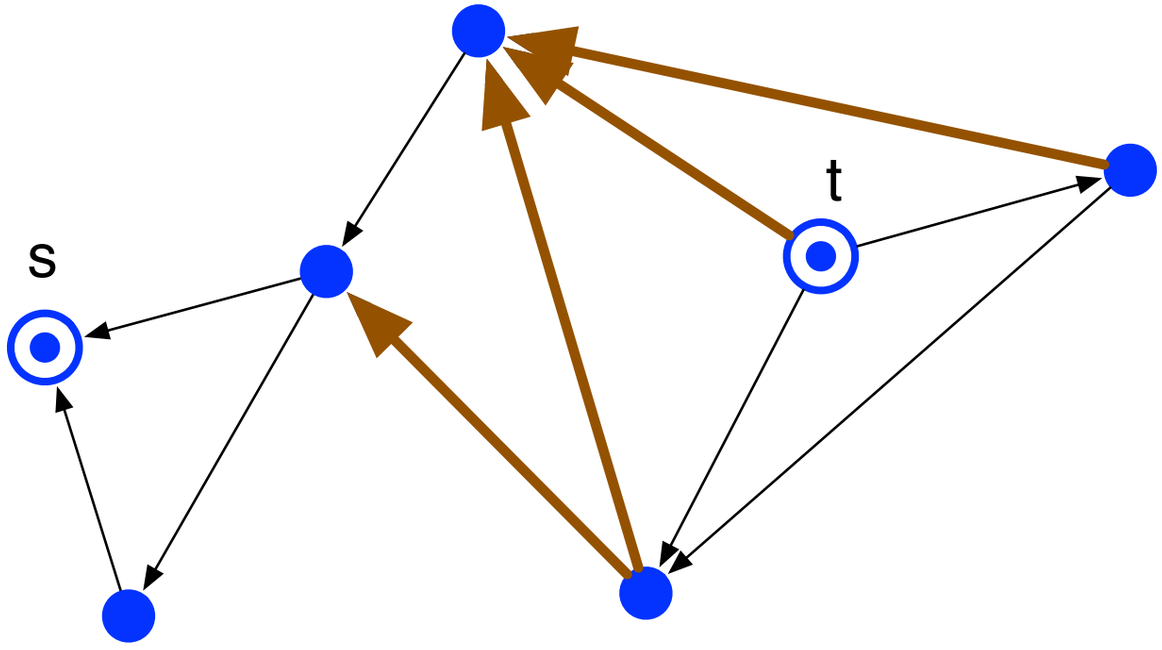
Forward Cuts

- Exists a bijection between minimum *s*,*t*-cuts in original graph and forward *t*,*s*-cuts in the DAG



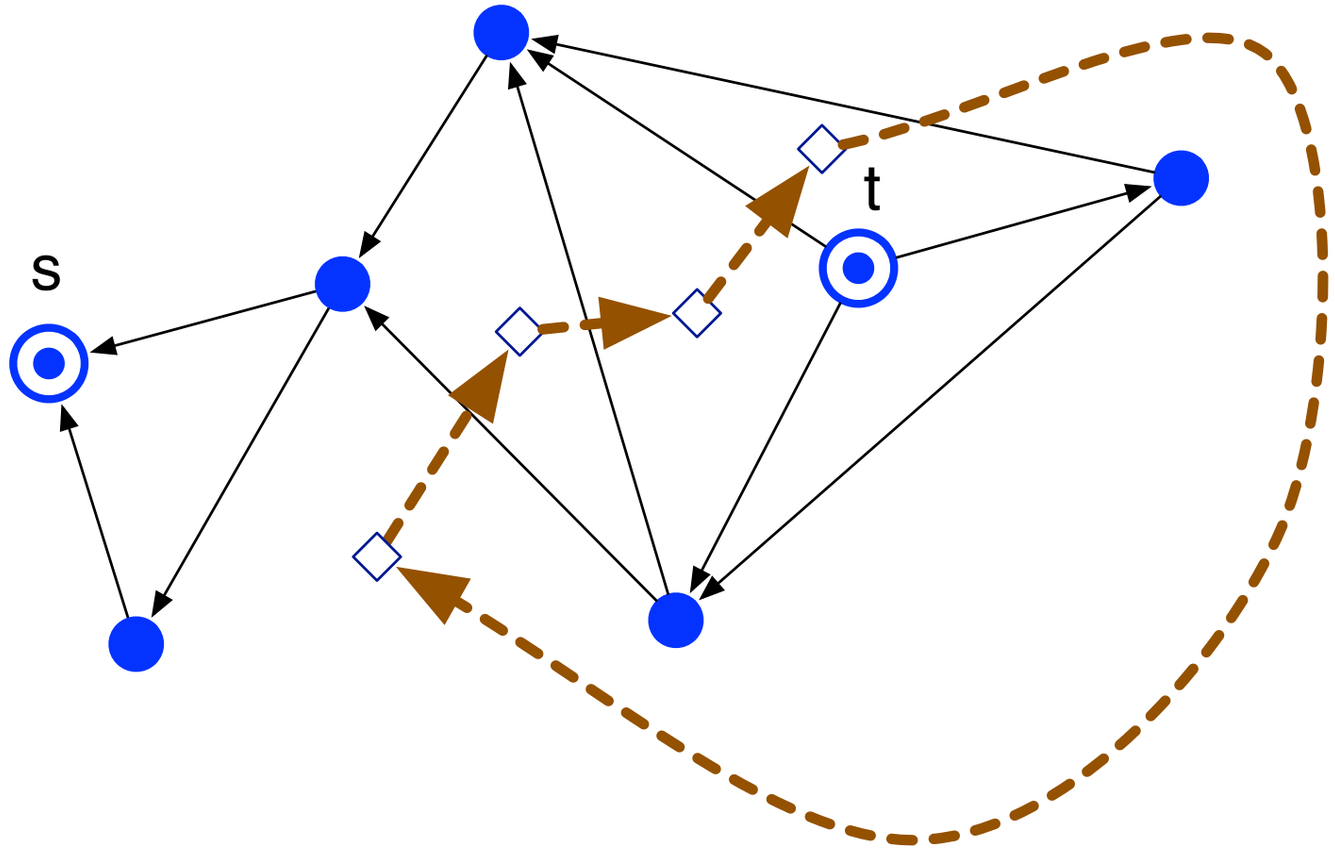
Forward Cuts

- Concentrate on counting forward *t*,*s*-cuts in surface embedded DAGs



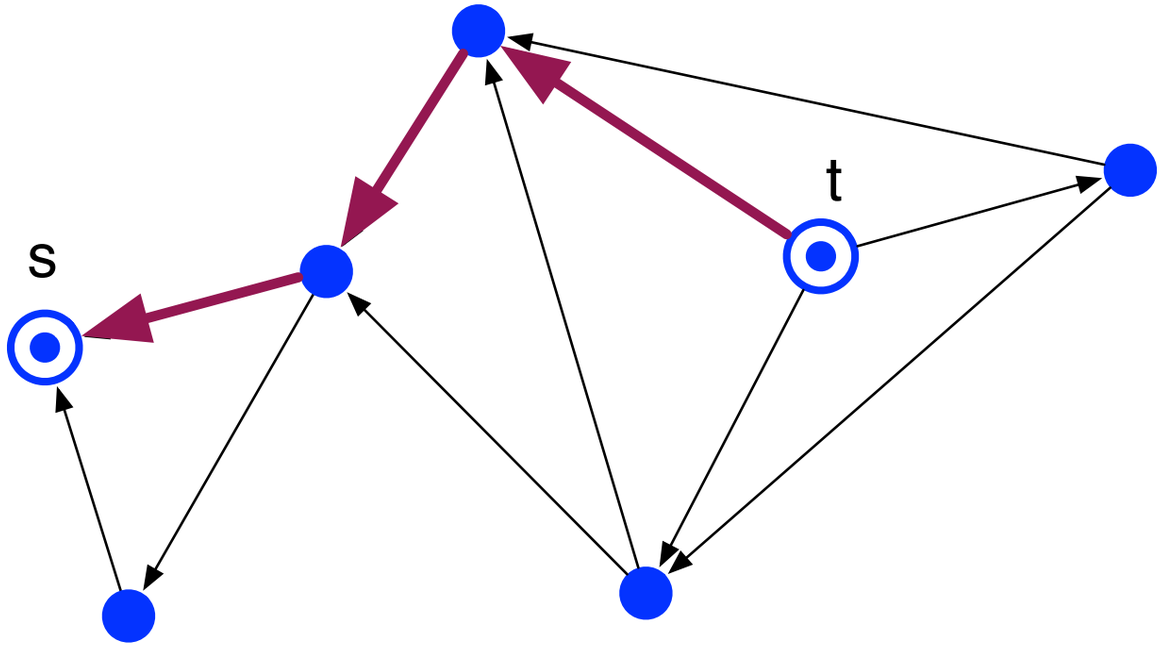
Counting in the Plane

- Forward cuts are dual to cycles



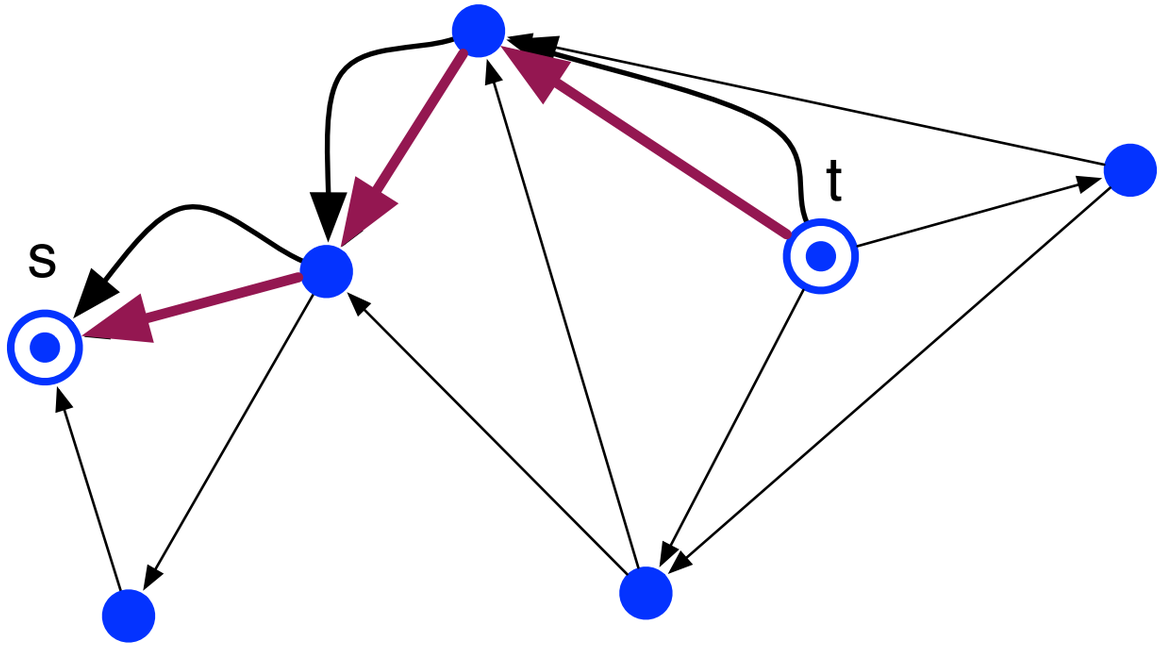
Counting in the Plane

- Pick a directed *t*,*s*-path



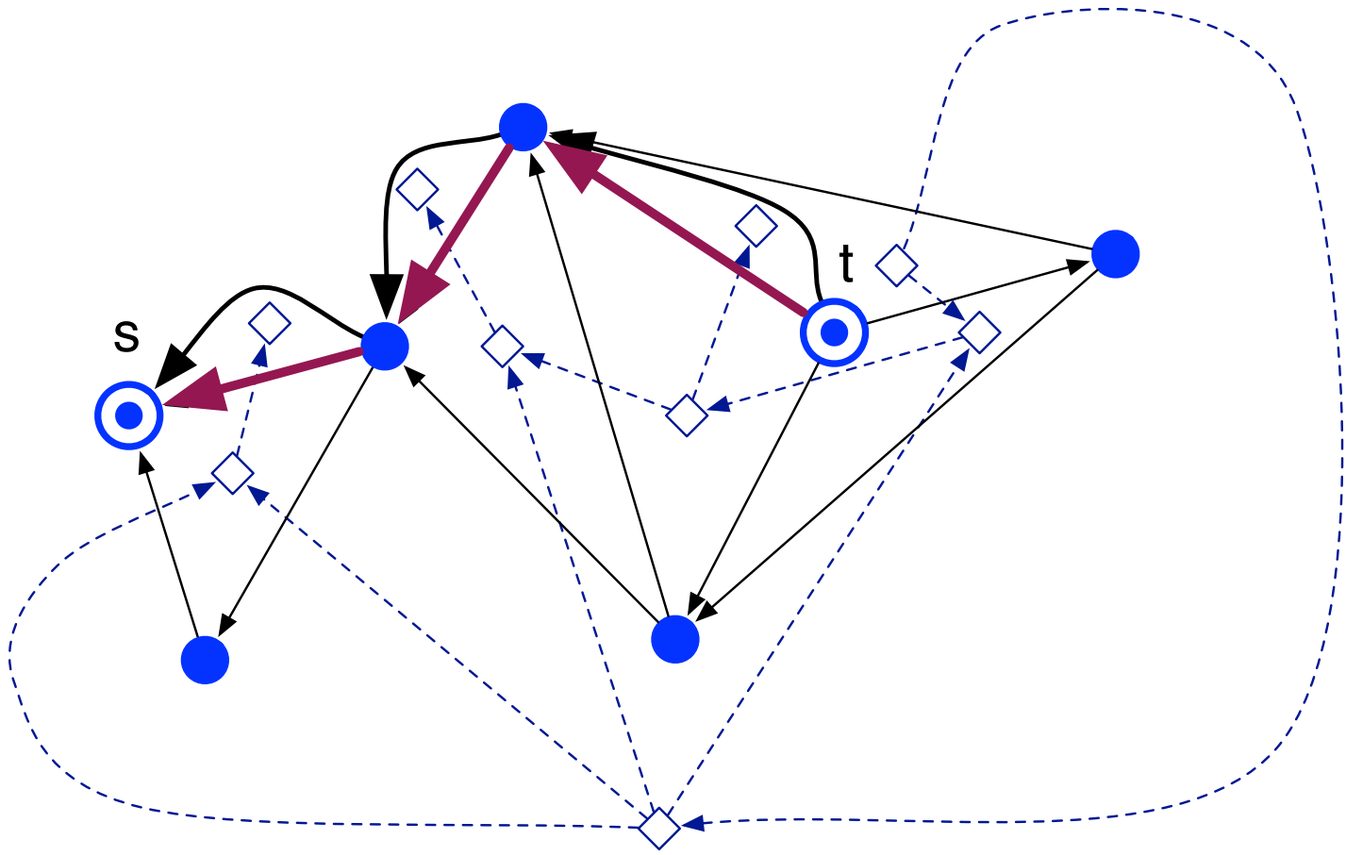
Counting in the Plane

- Add some edges to the path



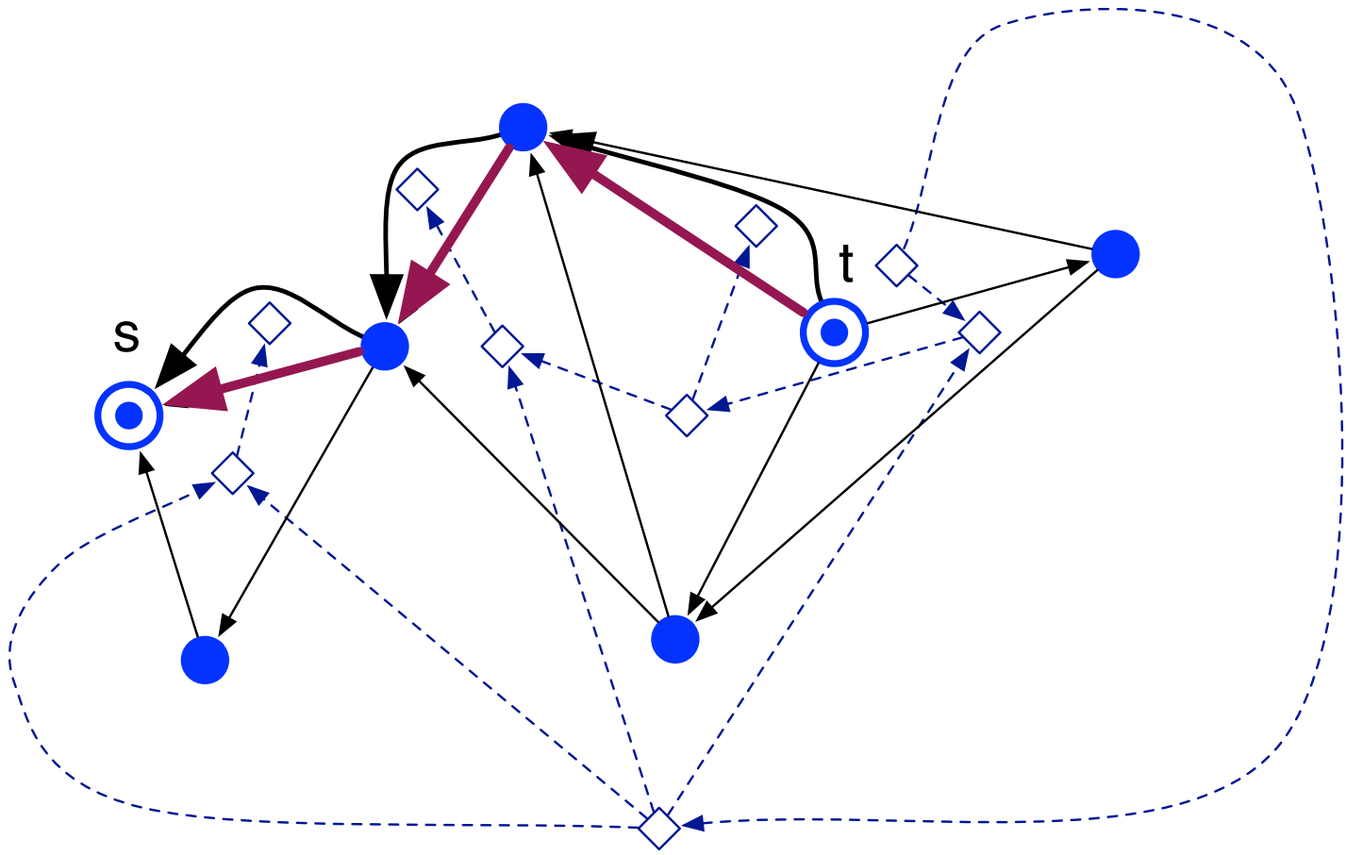
Counting in the Plane

- Create the "dual" graph without extra edges



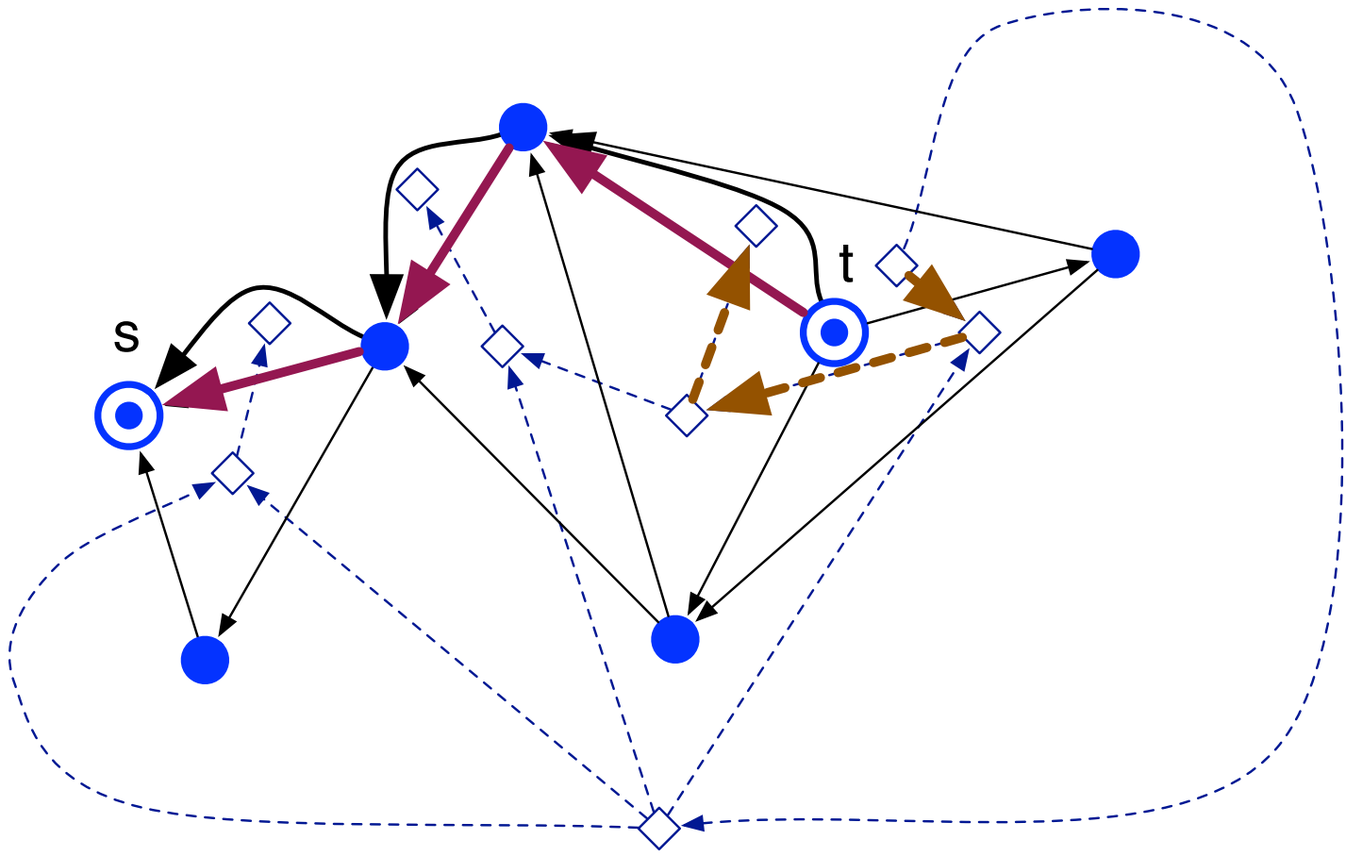
Counting in the Plane

- Dual is a DAG, so we can count paths between a pair of faces in linear time



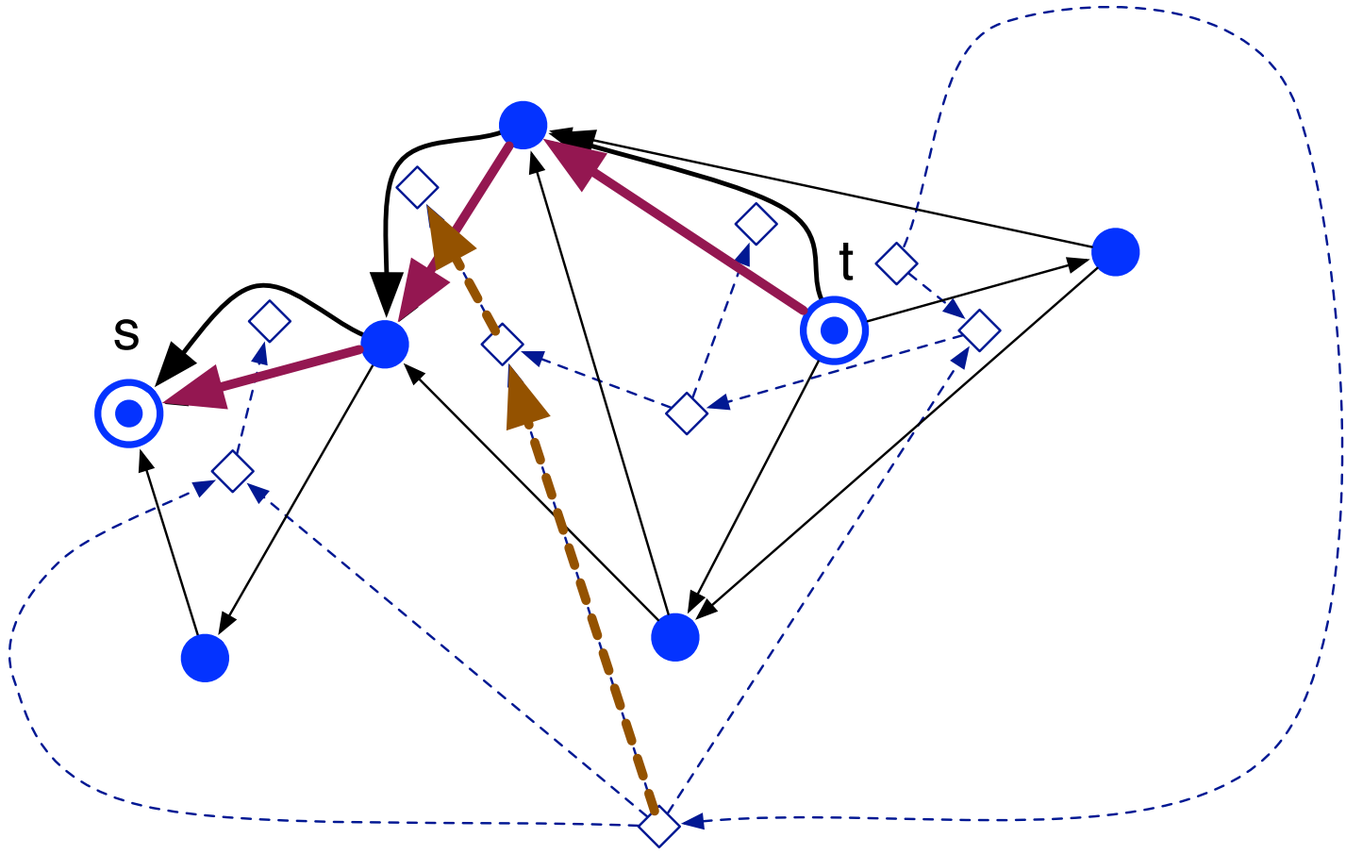
Counting in the Plane

- Dual is a DAG, so we can count paths between a pair of faces in linear time



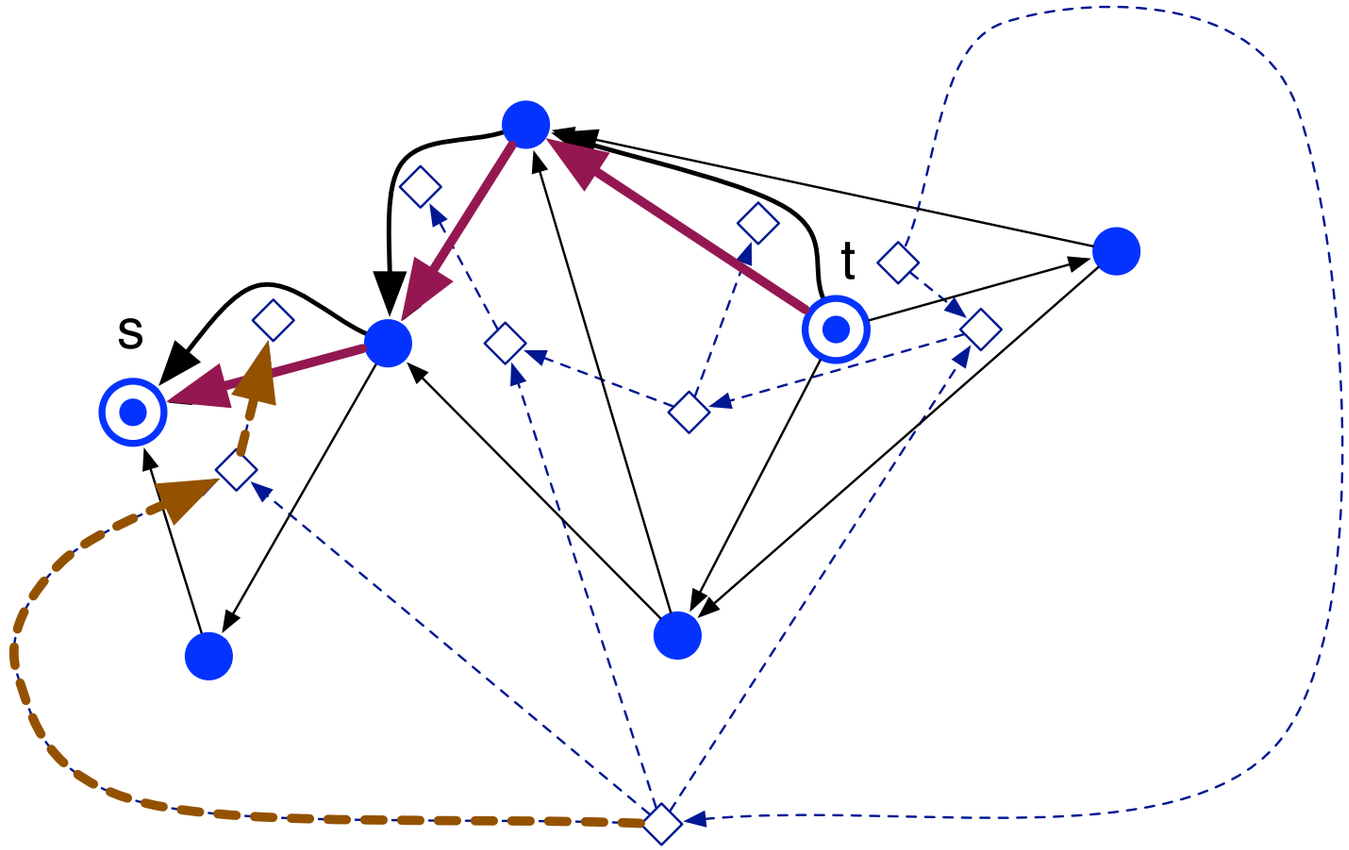
Counting in the Plane

- Dual is a DAG, so we can count paths between a pair of faces in linear time



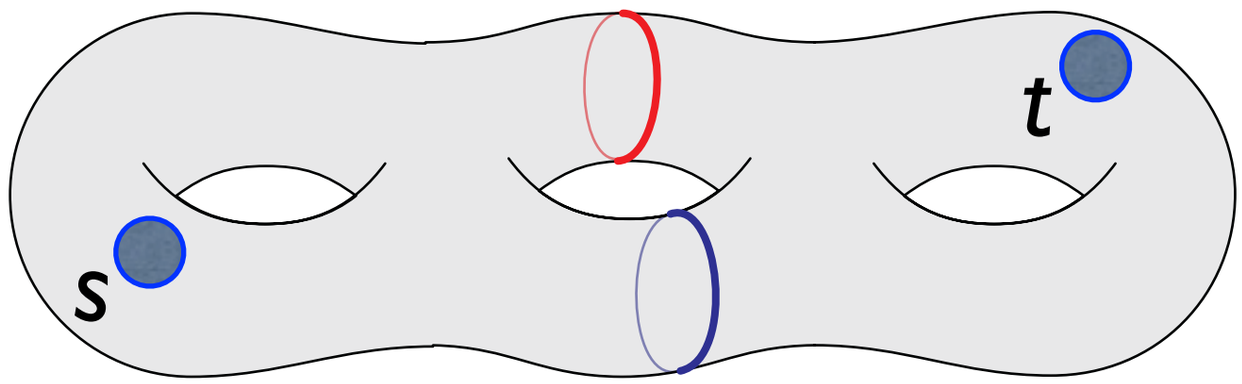
Counting in the Plane

- Dual is a DAG, so we can count paths between a pair of faces in linear time



Difficulties in the Surface

- Minimum cuts may be dual to multiple cycles



- Can handle finding these cycles in the undirected case [CEN '09; INSW '11, EFN '12]

Difficulties in the Surface

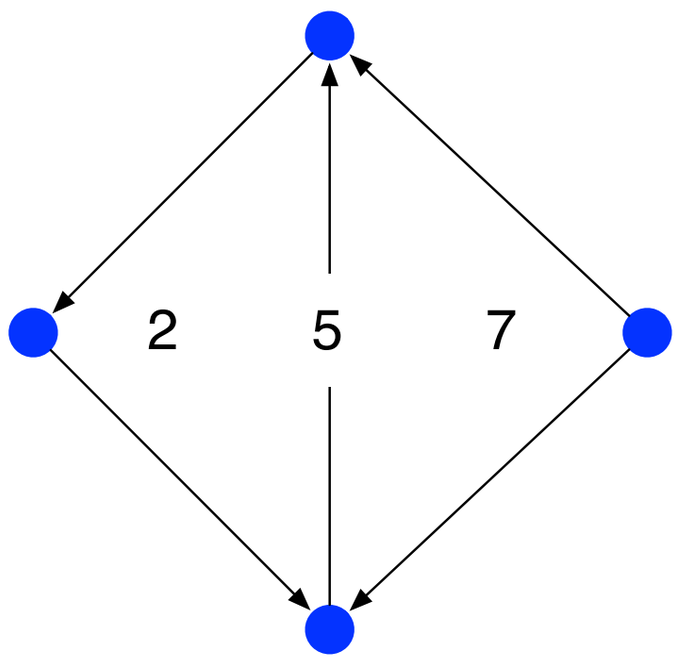
- Need to work directly with directed dual cycles
- Requires similar but more computationally difficult techniques
- Integer vs. $\mathbb{Z}_2$-homology
- Further complications when primal graph is not triangulated

Chains and Circulations

- A **[0,1,2]-chain** assigns integer values to each [vertex, edge, face] of the graph
- A 1-chain ϕ is a **circulation** if $\Sigma_{(u \to v)} \phi(u \to v) - \Sigma_{(v \to u)} \phi(v \to u) = 0$ for each vertex u
- A set of directed cycles C **trivially generates** a circulation ϕ where $\phi(u \to v)$ equals the number of times $u \to v$ appears

Chains and Circulations

- The boundary of a 2-chain α is the 1-chain $\partial\alpha$ where $\partial\alpha(e) = \mathit{right}(e) - \mathit{left}(e)$
- **Boundary circulations** are the boundaries of 2-chains



Integer Homology

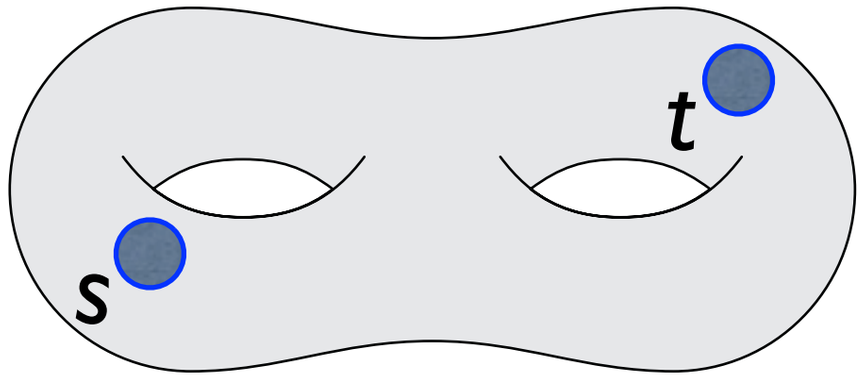
- Not all circulations are boundary circulations
- Circulations ϕ and Ψ are **homologous** if ϕ - Ψ is a boundary circulation
- Homology is an equivalence relation between circulations
- The vector space of homology classes is isomorphic to $\mathbb{Z}^{2g + \max\{0,b-1\}}$

Integer vs. $\mathbb{Z}_2$-homology

- Both give a sense of how cycles wrap around features in the surface
- Integer homology needed to understand directed cycles
- More efficient algorithms dealing directly with $\mathbb{Z}_2$-homology
- $n^{O(g2)}$ algorithm for sparsest cut uses integer homology [Patel '10]

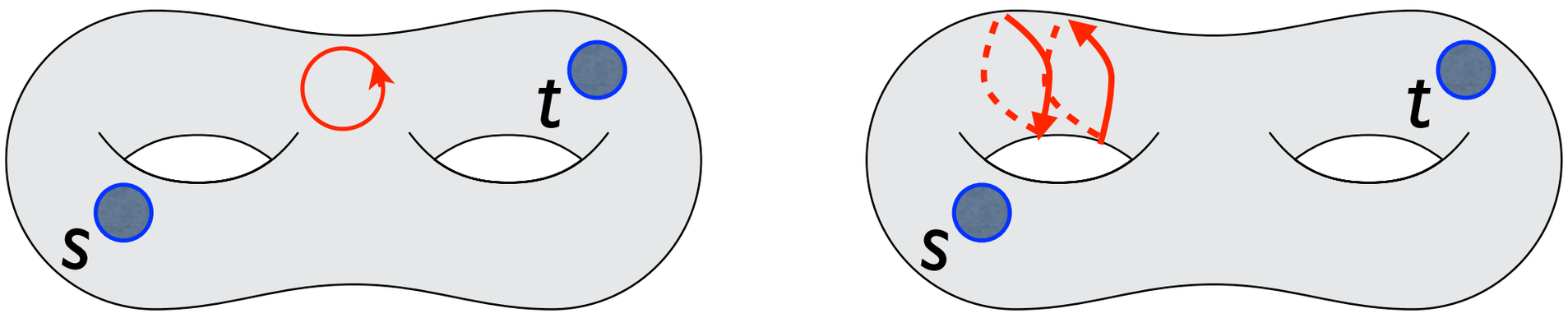
Dual of the DAG

- Want to describe forward t,s-cuts in terms of dual cycles
- For dual graph, remove duals of t and s from surface to create boundary cycles ∂t^* and ∂s^*
- Need a useful fact for our punctured dual



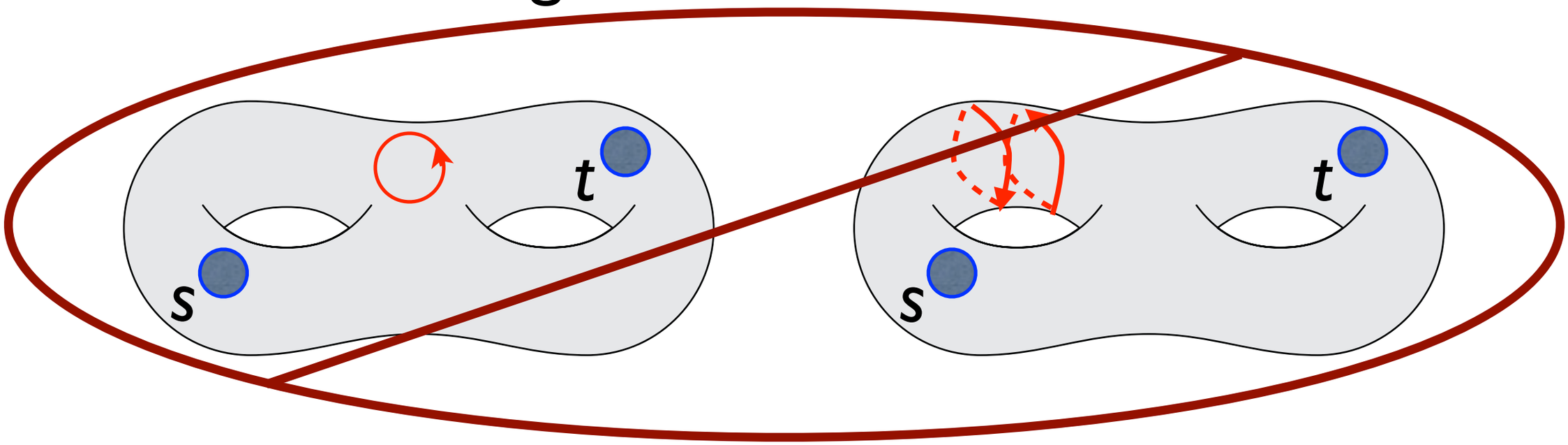
Dual of the DAG

- **Lemma:** No non-trivial boundary circulations in the dual assigning non-negative values to each directed edge
- Generalization of the planar "dual" from before being a DAG



Dual of the DAG

- **Lemma:** No non-trivial boundary circulations in the dual assigning non-negative values to each directed edge
- Generalization of the planar "dual" from before being a DAG

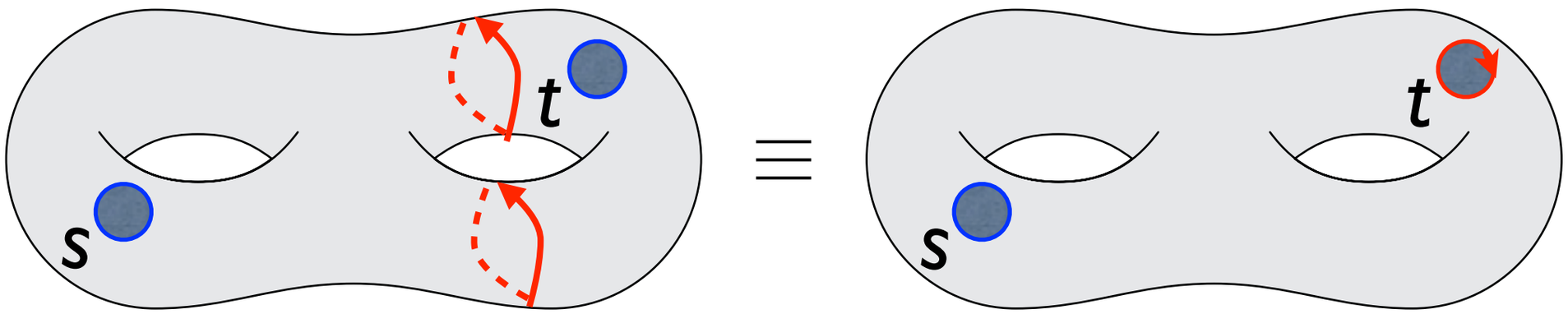


Dual of a Cut

- Characterization has some surprising consequences

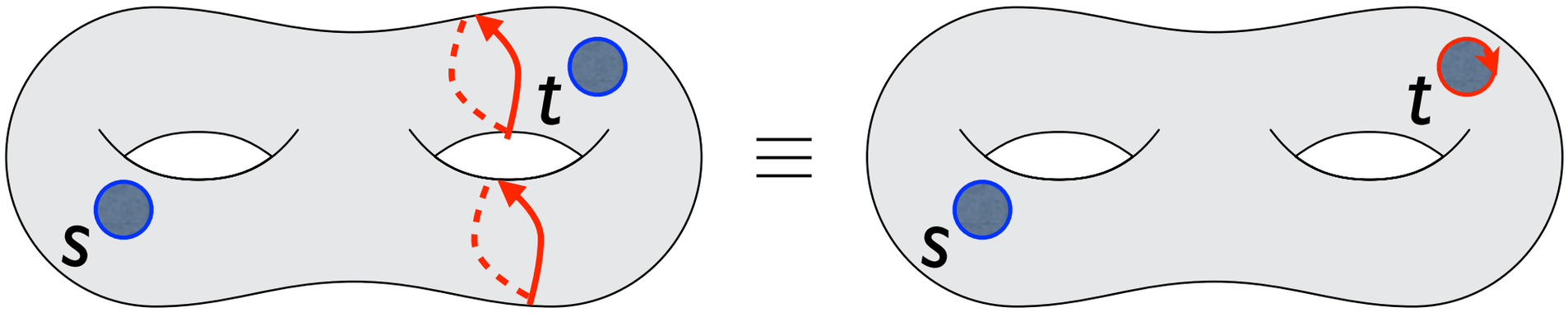
Dual of a Cut

- **Lemma:** The dual of every forward *t*,*s*-cut trivially generates a boundary circulation ϕ homologous to ∂t^*



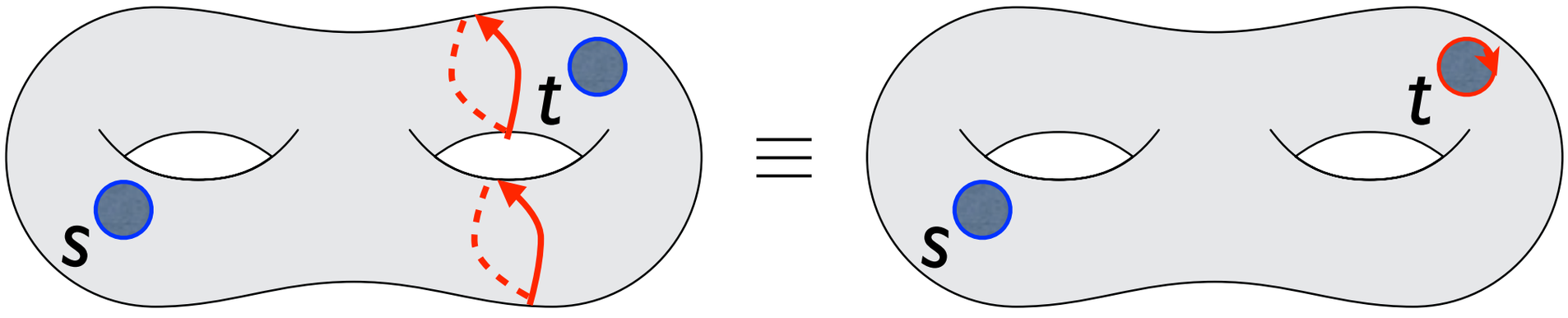
Dual of a Cut

- **Lemma:** For every non-negative circulation ϕ in the dual homologous to ∂t^*, there exists a forward t,s-cut containing only edges e such that $\phi(e) > 0$



Dual of a Cut

- **Lemma:** For every non-negative circulation ϕ in the dual homologous to ∂t^*, there exists a forward t,s-cut containing only edges e such that $\phi(e) > 0$
- In fact, its homology alone guarantees ϕ assigns 0 or 1 to every directed edge



Why 0 or 1

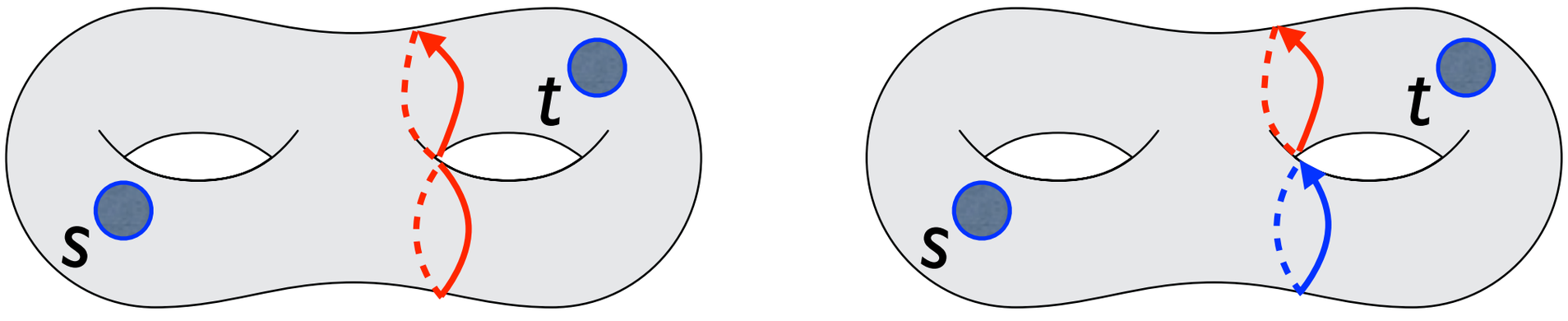
- Let T be the forward t,s-cut with edges e such that $\phi(e) > 0$
- Let ϕ_T be the circulation trivially generated by the edges of T
- If $\phi(e) > 1$ for any edge e, then $\phi - \phi_T$ is non-trivial and non-negative
- But $\phi - \phi_T$ is a boundary circulation

Cuts as Cycles

- There exists a bijection between for forward *t*,*s*-cuts and **(0,1)-circulations** in the dual in a particular homology class
- Can represent (0,1)-circulations as collections of edge disjoint cycles

Overcounting

- Looking for edge disjoint cycles may lead to overcounting
- Distinct collections of edge disjoint cycles may contain the same set of edges and trivially generate the same circulation



Triangulations Help

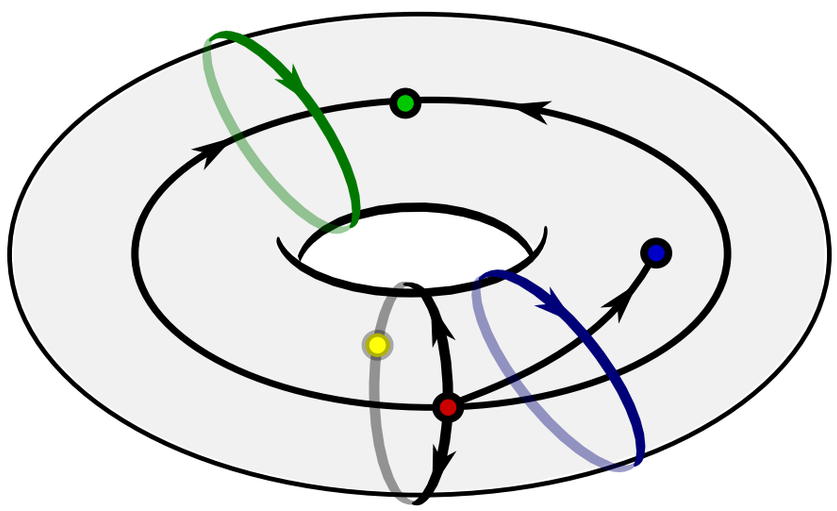
- Primal triangulations imply 3-regular duals
- 3-regularity implies edges disjoint cycles are also vertex disjoint
- Here there exists a bijection between collections of edge disjoint cycles and forward t,s-cuts

Triangulations Help

- Triangulating the primal is an interesting problem in itself
- Assume the DAG is triangulated for now
- Apply some ideas for finding interesting cycles in undirected graphs [Chambers *et al.* '08; CEN '09]

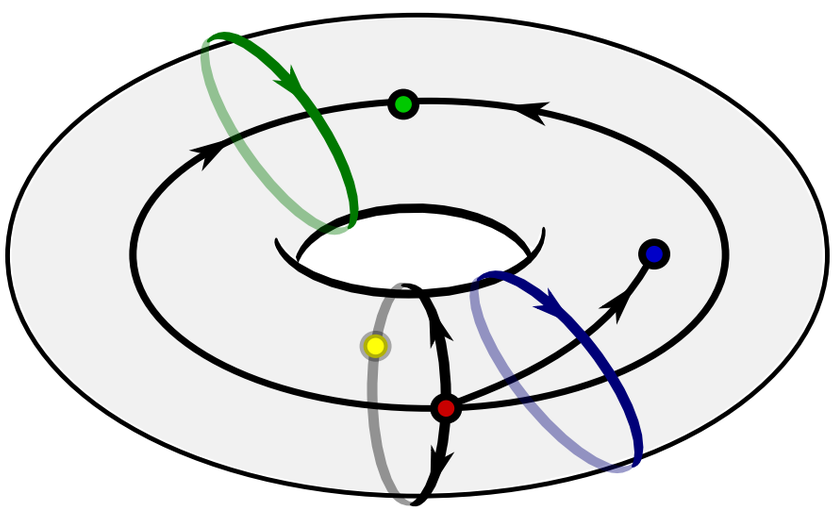
Polygonal Schema

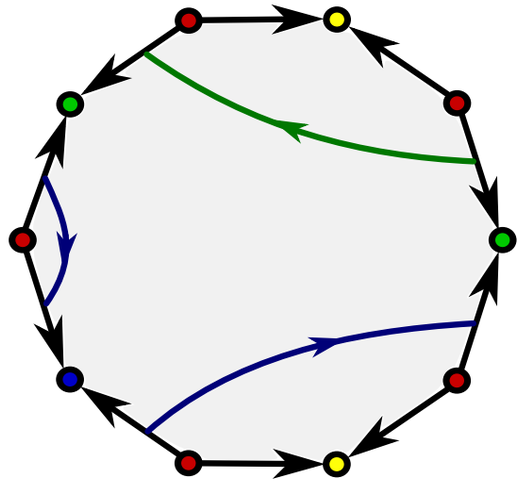
- Compute a system Λ of $2g$ loops and one directed path in the primal graph based at t
- Each loop made from two directed paths
  - Directed paths cross any forward t,s-cut at most once



Polygonal Schema

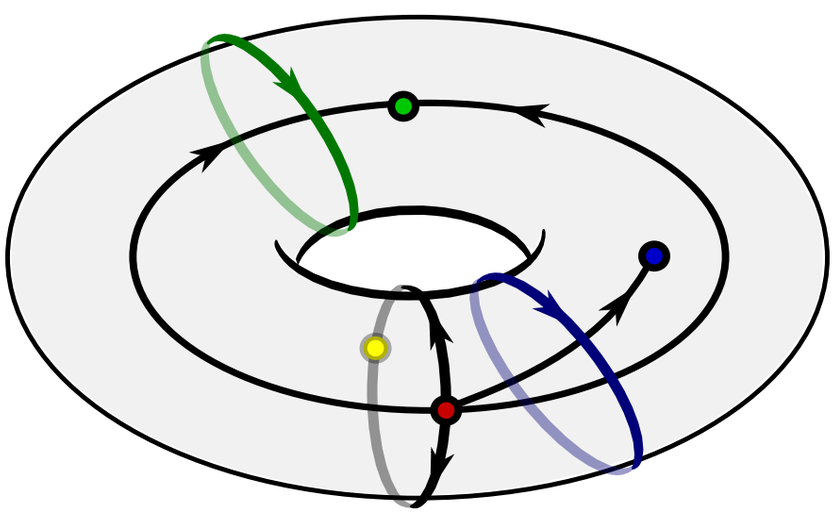
- Cutting along Λ turns the surface into a disk or **polygonal schema** with $8g + 2$ edges
  - Duals of forward t,s-cuts cross paths as arcs in the schema

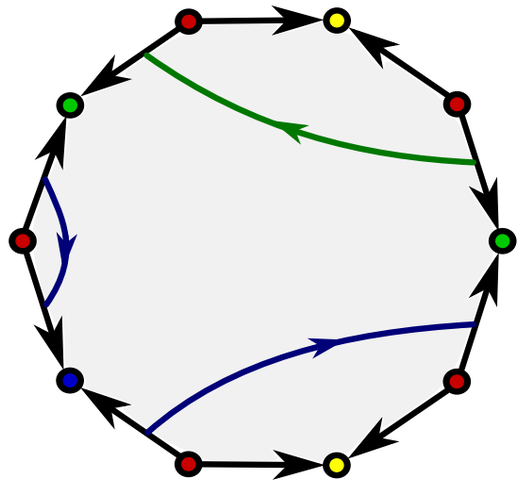


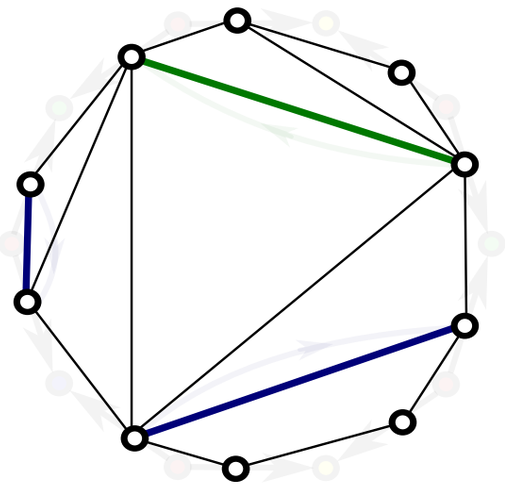


Polygonal Schema

- Swap vertices and edges to create the **dualized polygonal schema**
- Now any forward *t*,*s*-cut looks like a partial triangulation of the dualized schema

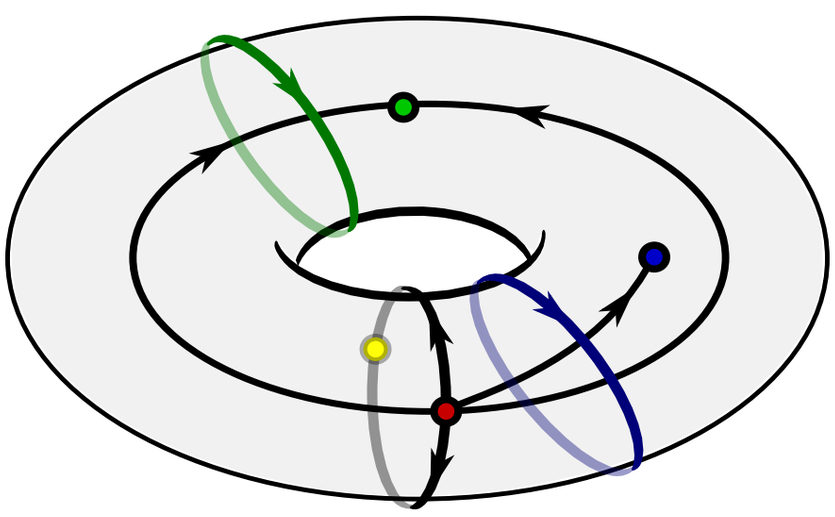


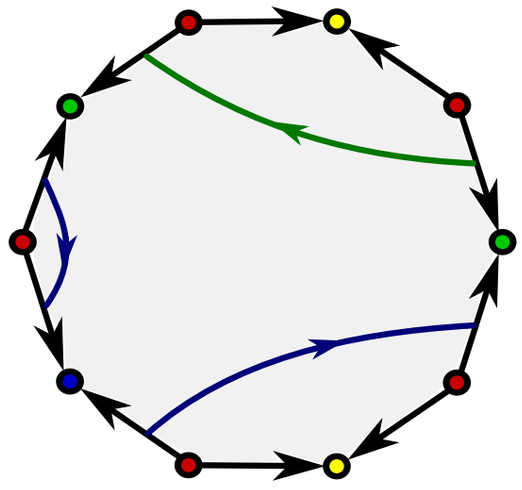


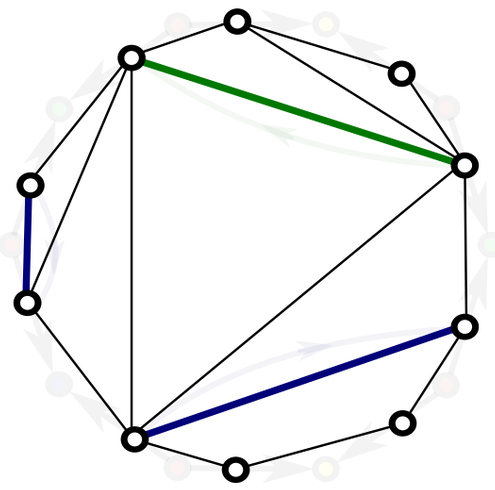


Crossing Sequences

- Exist only $2^{O(g)}$ partial triangulations
- Each describes several **crossing sequences** between dual cycles and the system Λ

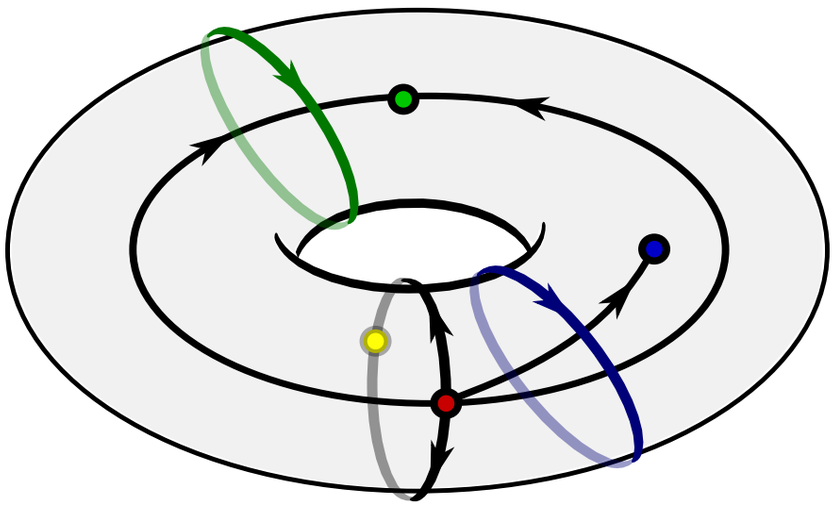


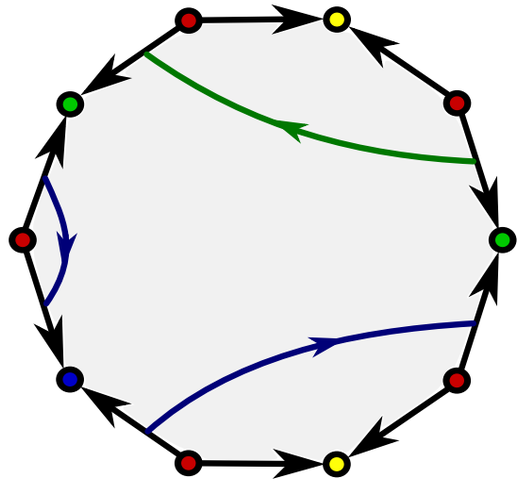


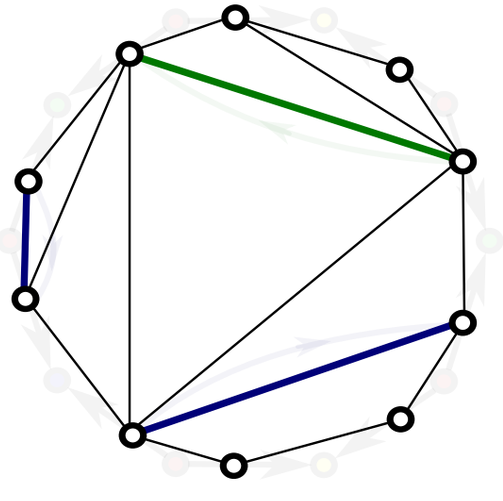


Crossing Sequences

- Can modify prior techniques to determine homology class [CEN '09; Erickson, Nayyeri '11] and count cycles [Kutz '06, BF '12] for a given set of crossing sequences
- Count directed paths in the **universal cover**







Final Counts

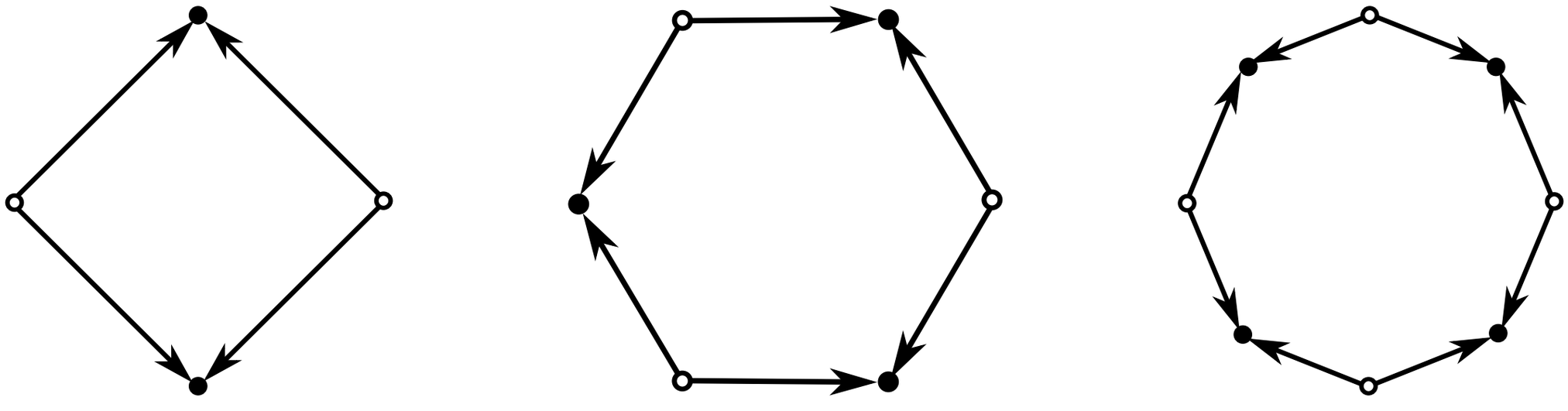
- Can count cycles with a given crossing sequence in $O(g\, n^2)$ time
- Multiply to get number of collections with a given set of crossing sequences
- Sum over counts for all $2^{O(g)}$ partial triangulations
- $2^{O(g)}\, n^2$ time spent in total

Sampling Cuts

- Sample a partial triangulation proportionally to the number of cuts using that triangulation
- For each crossing sequence, sample a cycle using planar techniques [BF '12]
- $O(g^2 n)$ time spent per sample

Triangulation is Hard

- Needed a triangulated DAG to avoid overcounting
- Cannot subdivide a face with new edge $u \to v$ if there exists no path from u to v
- Call these **irreducible** faces



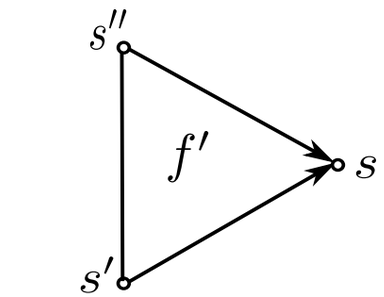
Limited Irreducibility

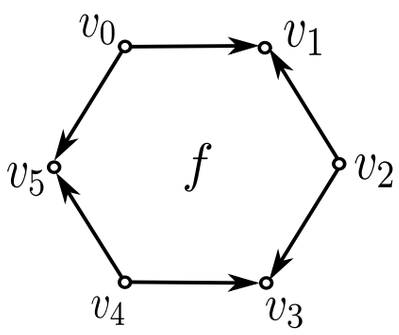
- **Lemma:** The total degree of irreducible faces is at most $12g$
- Bound based on the maximum number of non-crossing loops that do not separate the surface

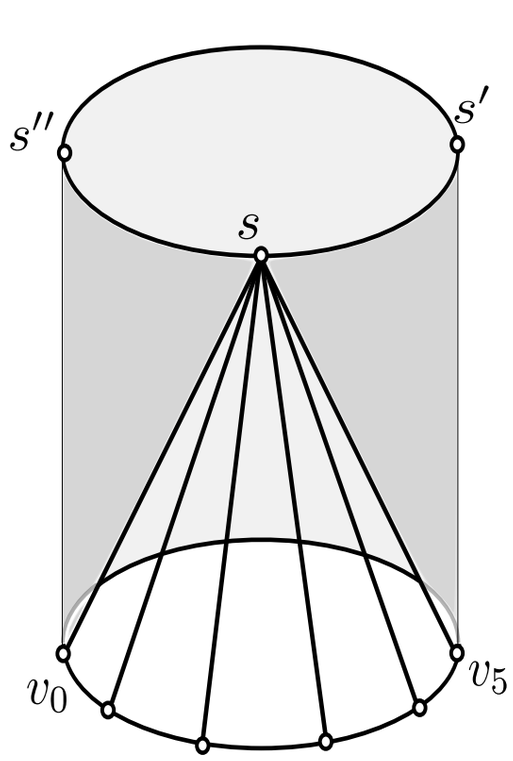
Adding Handles

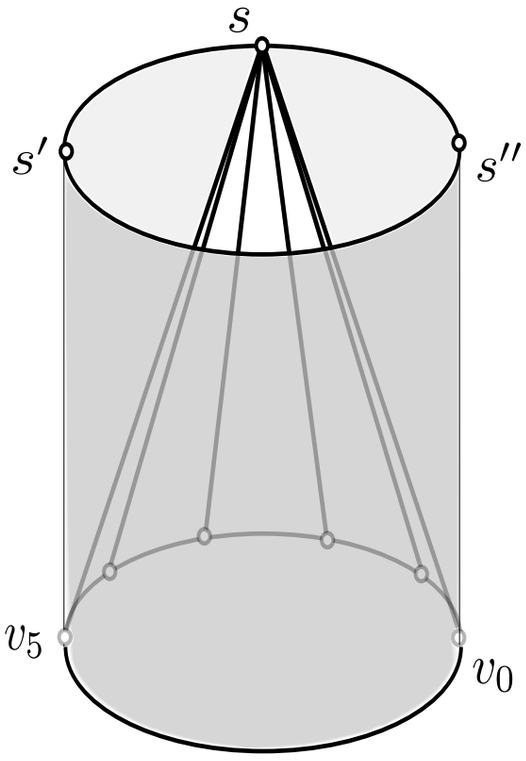
- Iteratively add a handle between each irreducible face *f* and any face *f'* incident to *s*

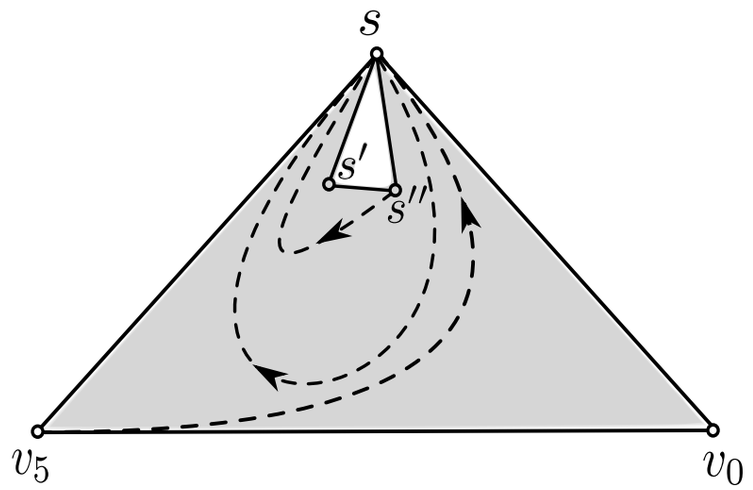
- Add edges to s to triangulate new faces





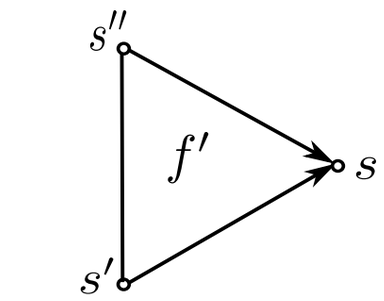


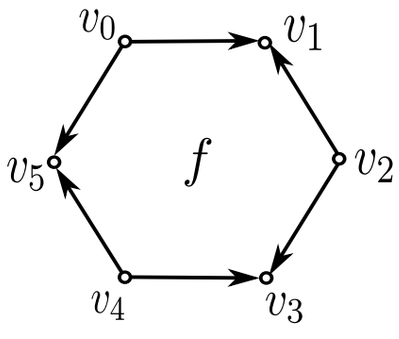


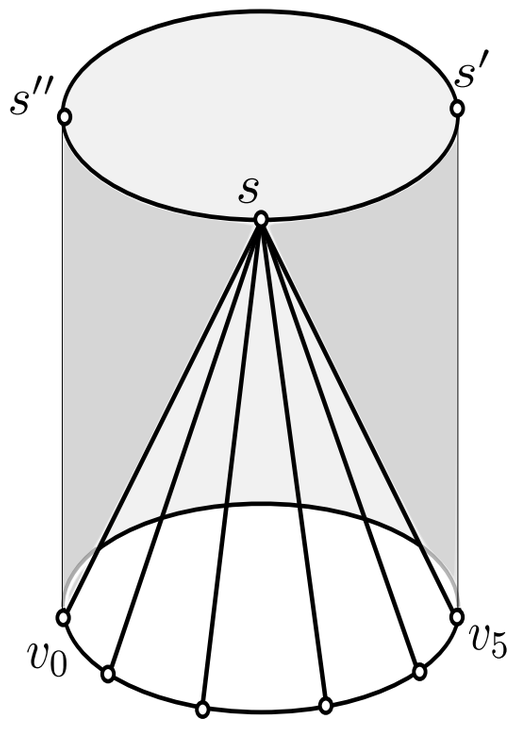


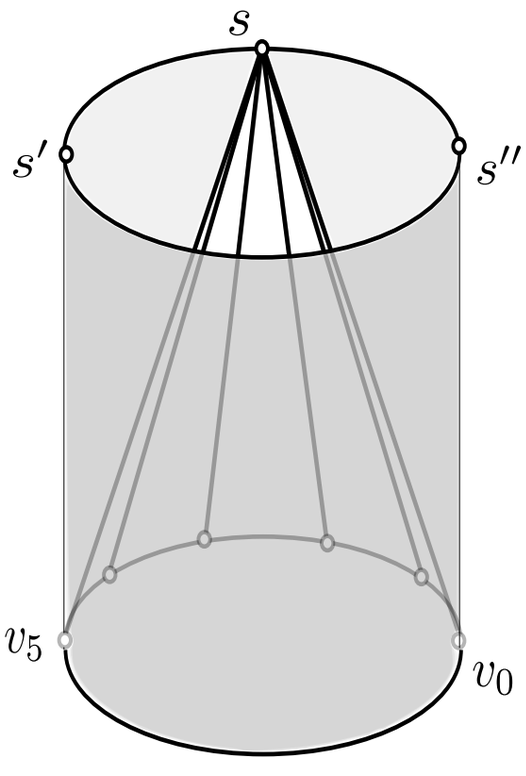
Adding Handles

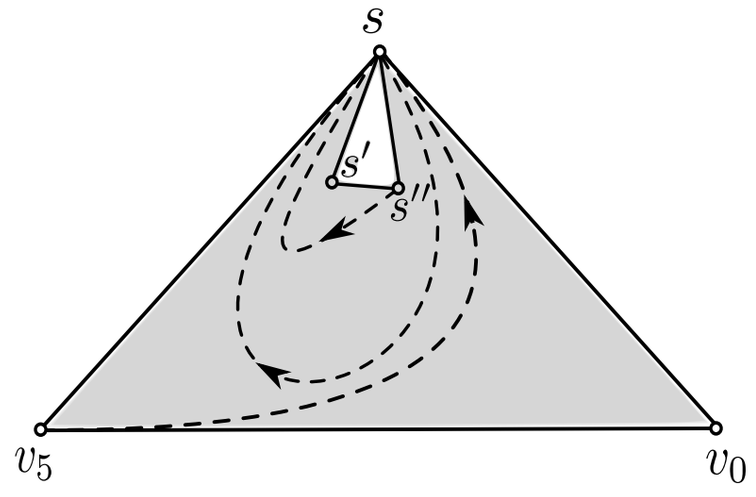
- Genus of surface and number of edges increases by $O(g)$
- Can now apply algorithm for triangulated primal graphs











Conclusions

- Gave an algorithm to efficiently count and sample minimum cuts in surface embedded graphs
- Required some new observations in integer homology
- Required new techniques to triangulate the primal graph

Thank you