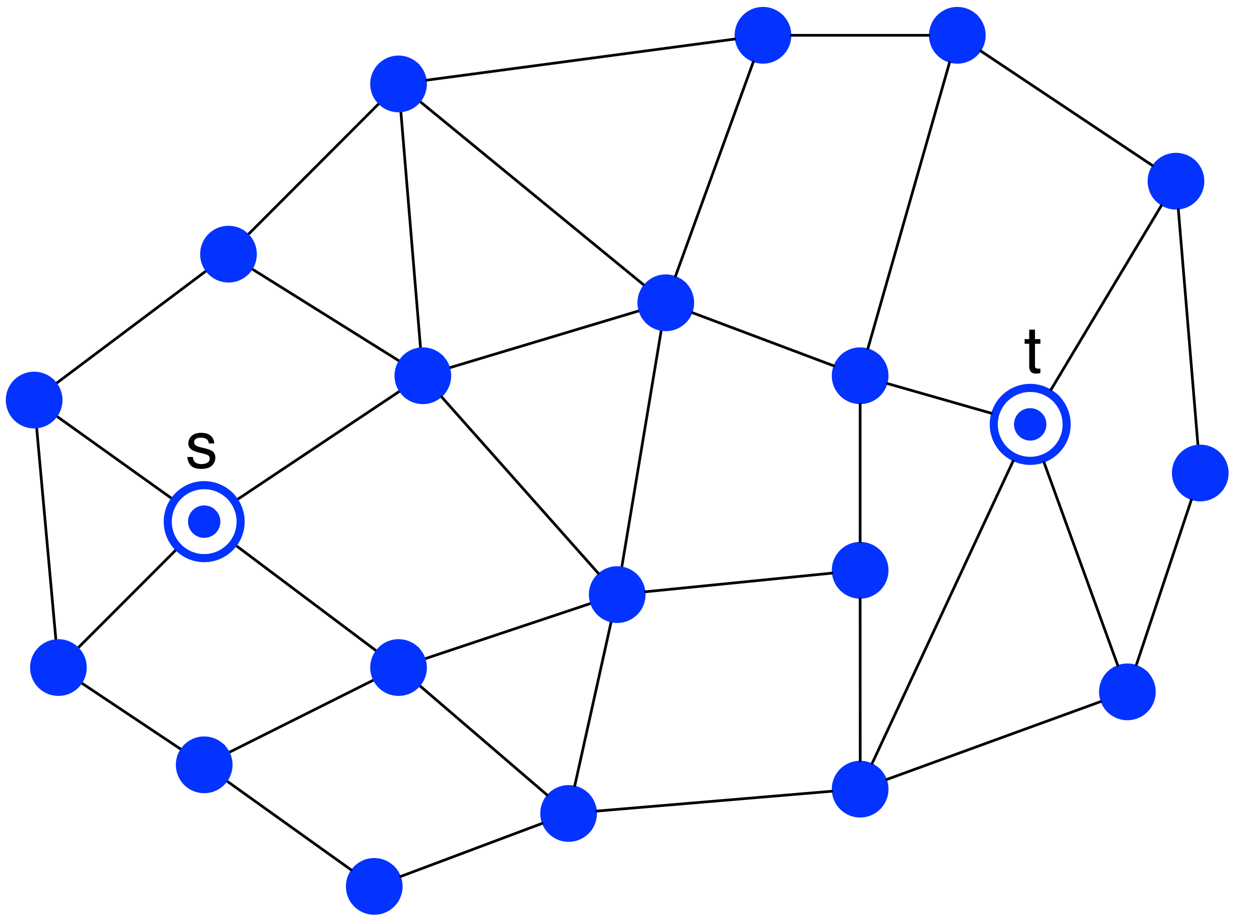
Counting and Sampling Minimum Cuts in Genus g Graphs

Erin W. Chambers, **Kyle Fox**, and Amir Nayyeri

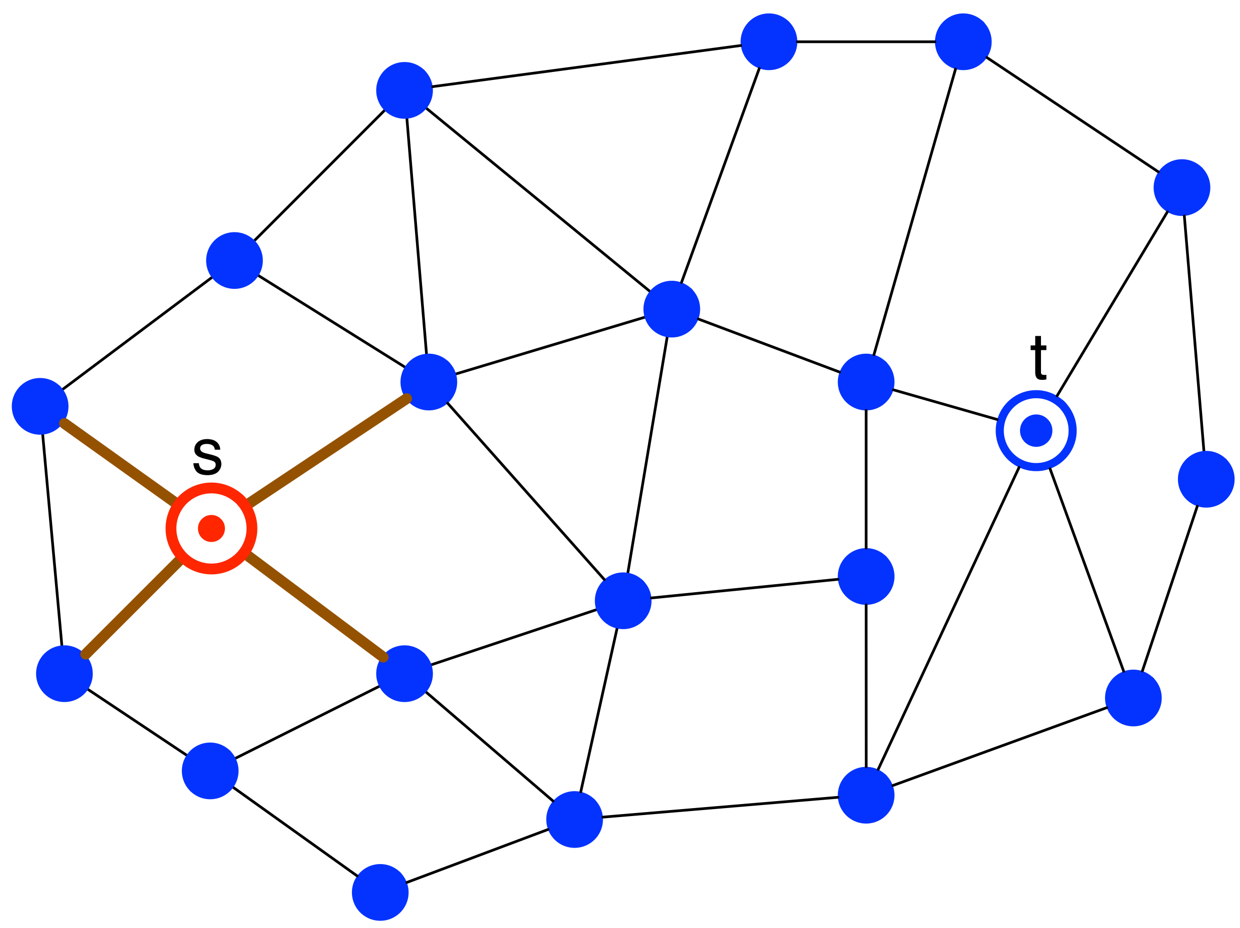
Counting Minimum Cuts

- Input designates vertices s and t and positive weights on edges
- Want to count vertex subsets S with $s \in S$, $t \notin S$ that minimize total edge weight leaving S

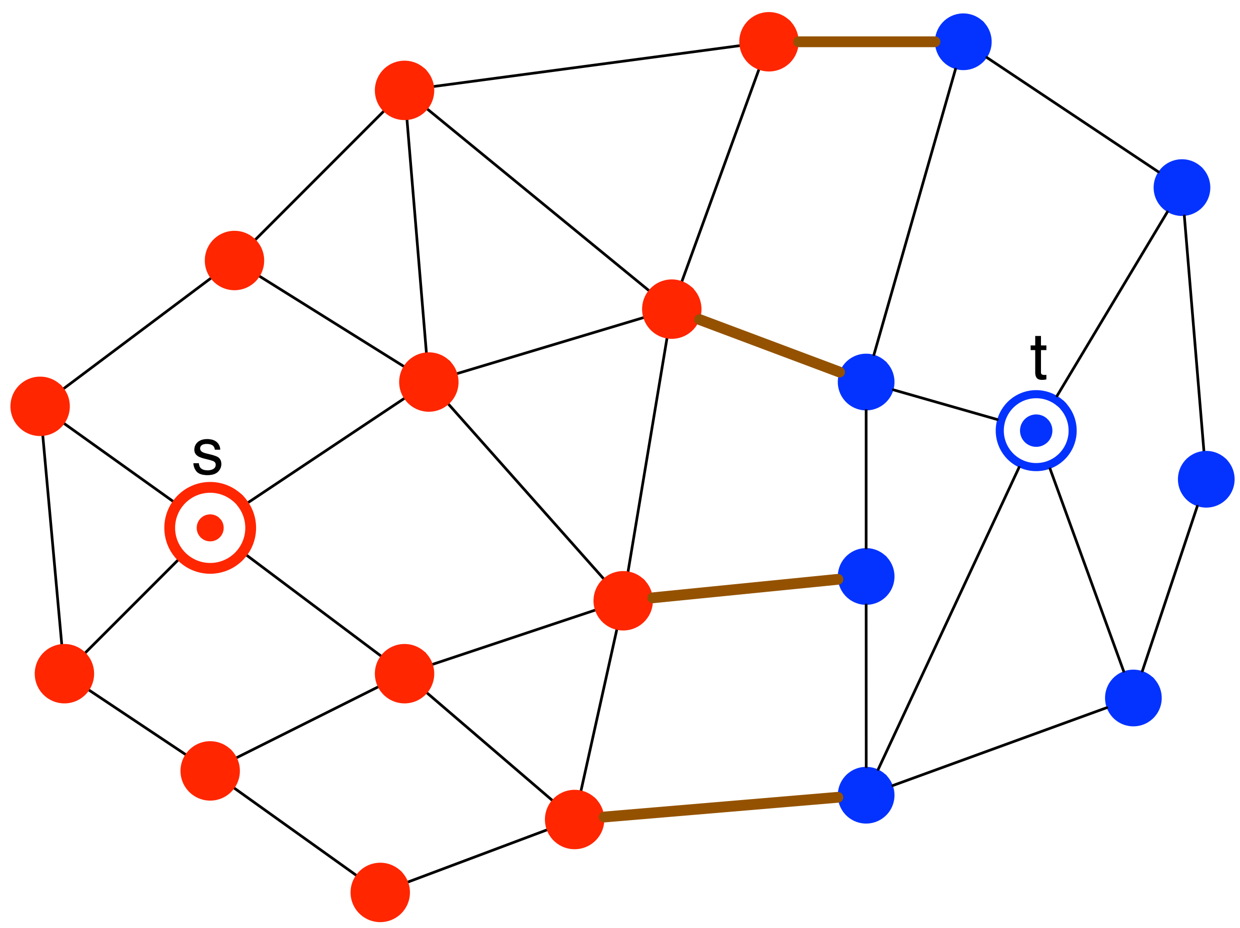
Counting Minimum Cuts



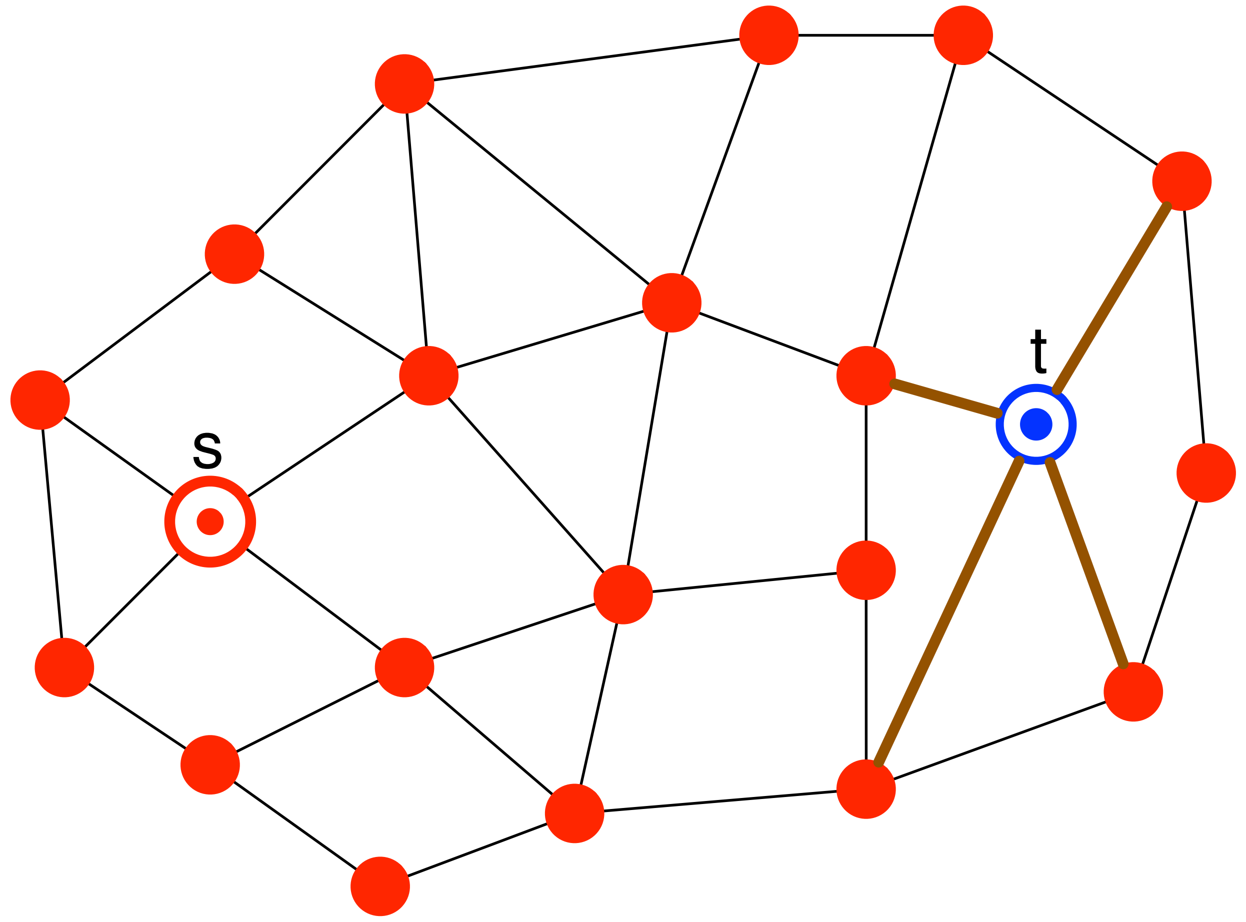
Counting Minimum Cuts



Counting Minimum Cuts



Counting Minimum Cuts



Counting Minimum Cuts

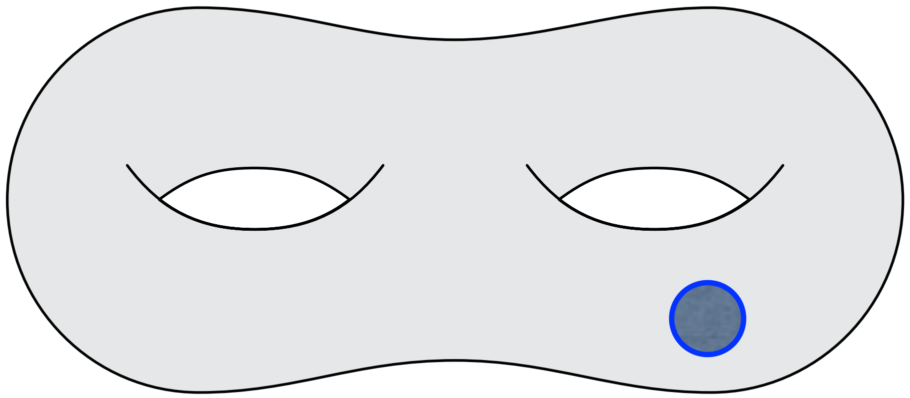
- Counting algorithms can be used for sampling
- Applications to stochastic network reliability and image segmentation
- Fundamental question to ask; easy to find, hard to count

Complexity of Counting

- Exactly 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]

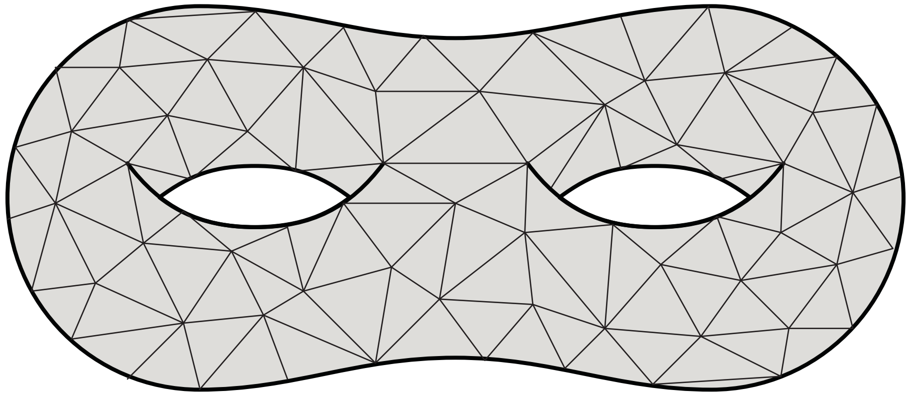
Surfaces

- 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

- With n vertices and $O(n)$ edges
- *Most* planar results generalize easily
- Cuts and flows only recently [Chambers, Erickson, Nayyeri STOC/SOCG '09; Italiano et al. '11; Erickson, F, Nayyeri '12; Chambers, F, Nayyeri, '12]

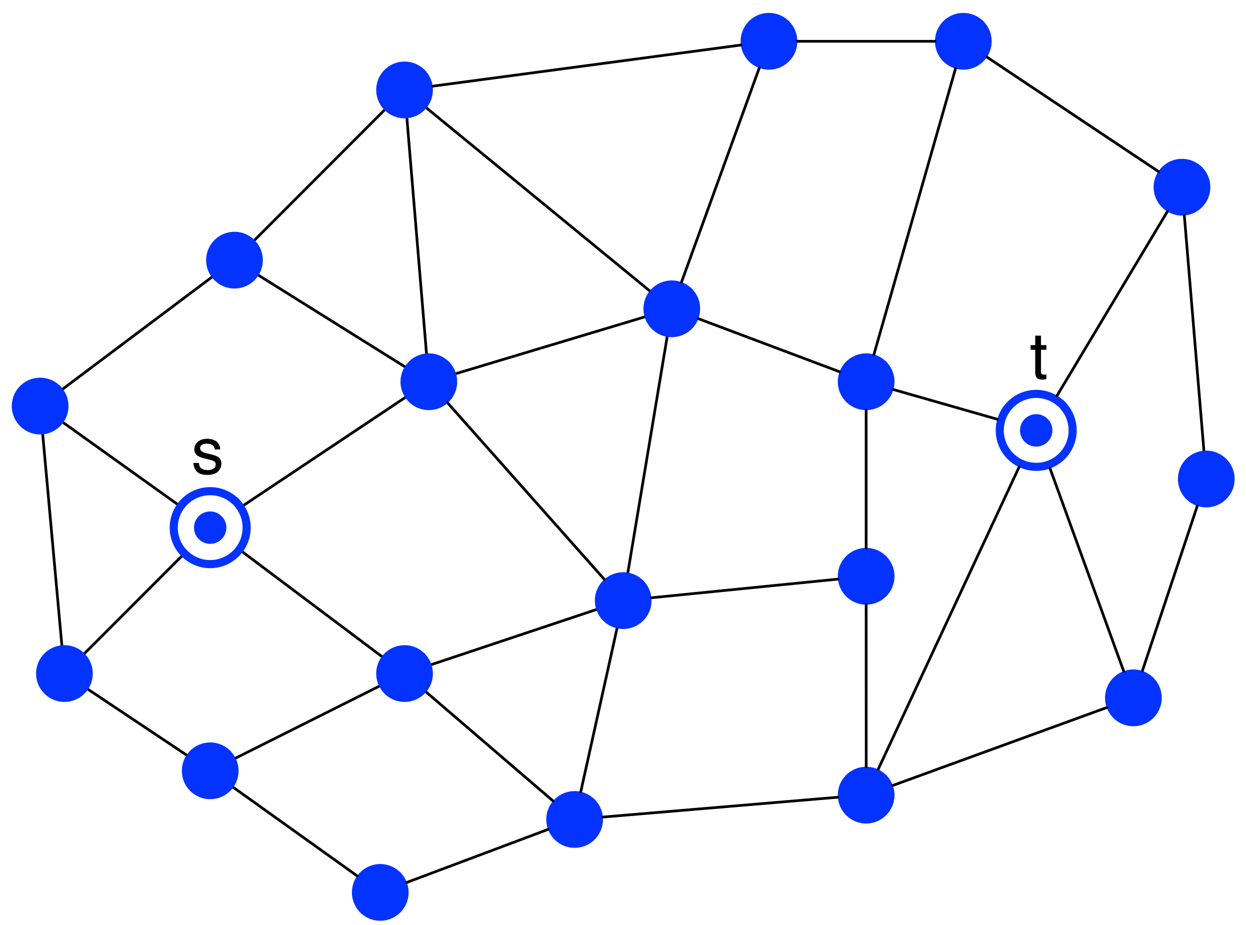


Our Results

- An algorithm to exactly count minimum weight s,t-cuts in directed embedded graphs in $2^{O(g)} n^2$ time
- "Aggregative" as defined on Monday [Alvarez, Seidel '13]
- After counting, can sample minimum cuts in $O(g\ n)$ time each
- Only assumes positive edge lengths and every point lying on an s to t walk

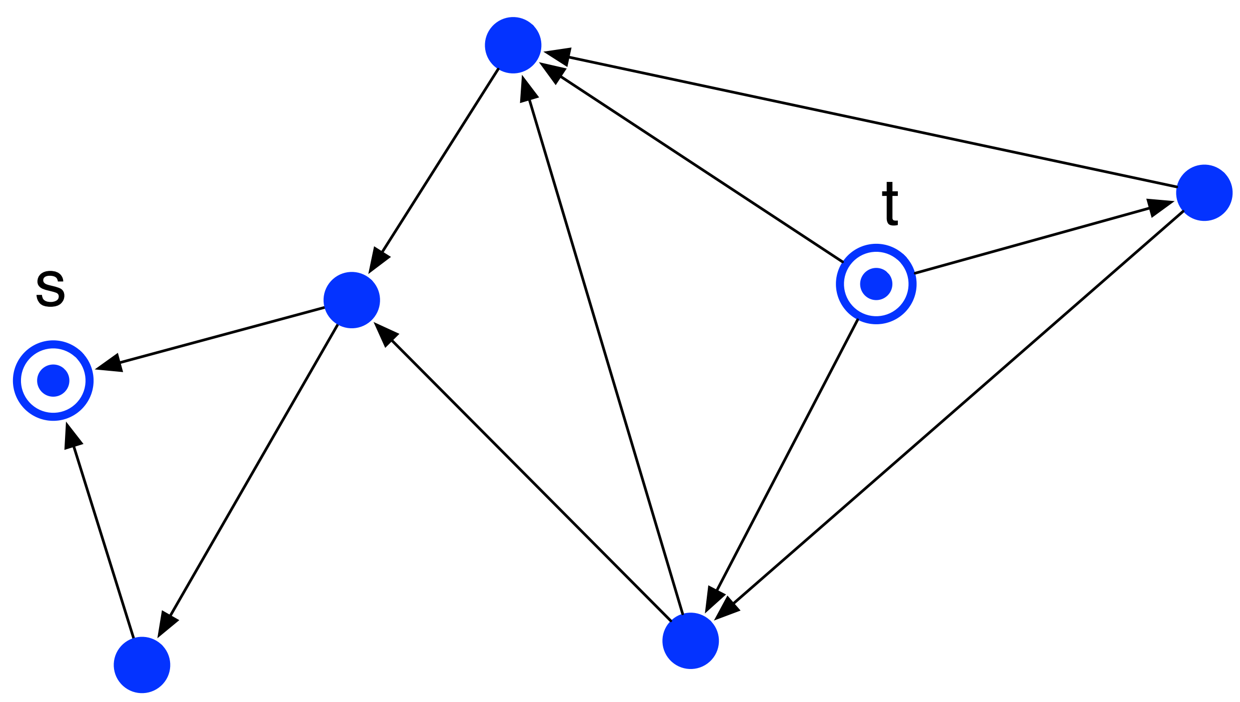
General Counting Idea

- Begin by computing a maximum *s*,*t*-flow
- Contract all cycles which can carry additional flow (are residual)



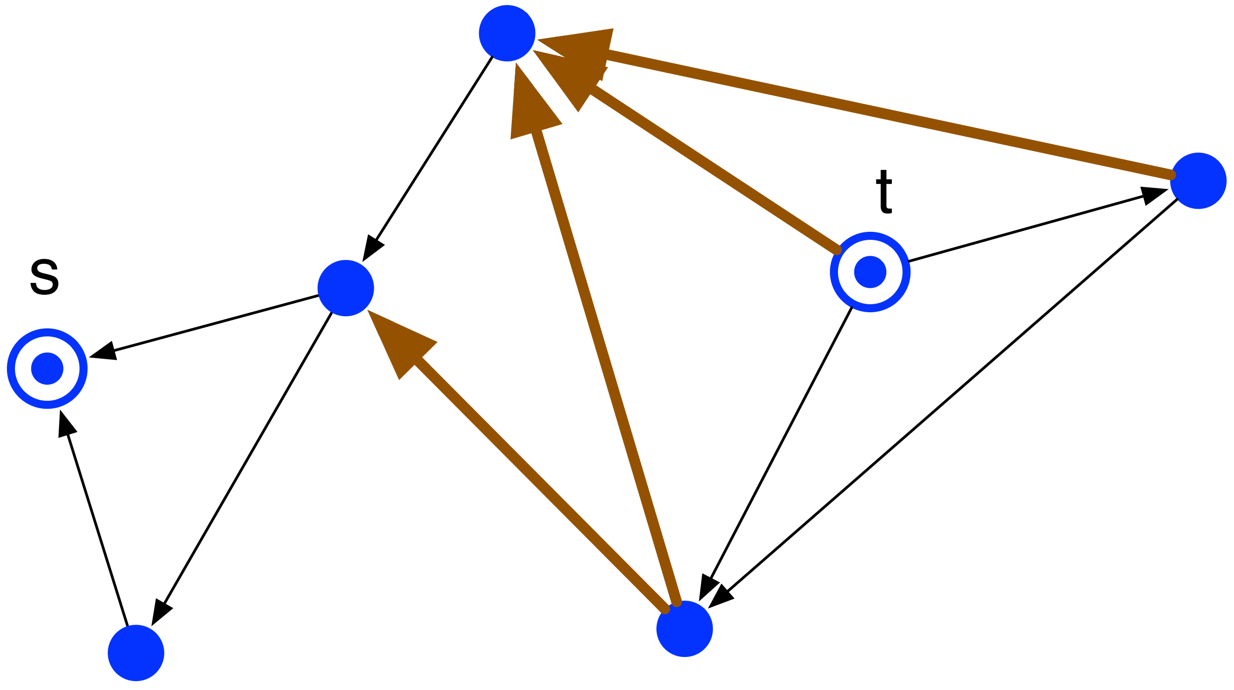
General Counting Idea

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



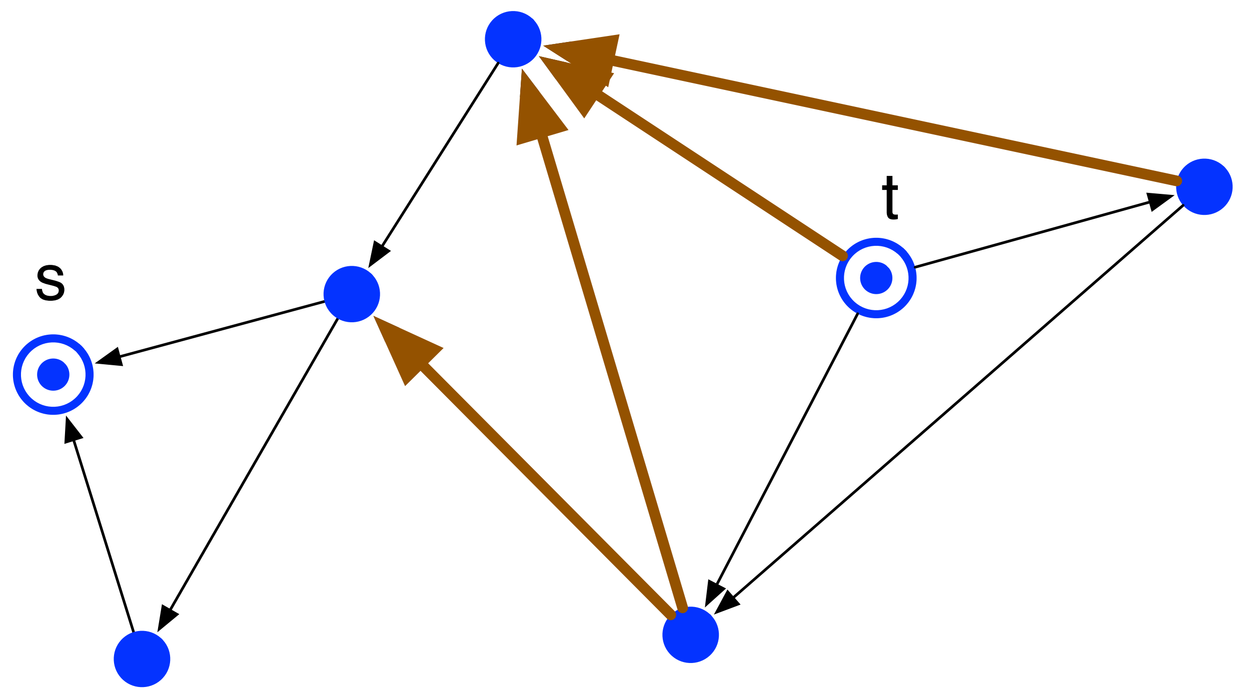
Forward Cuts

- 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



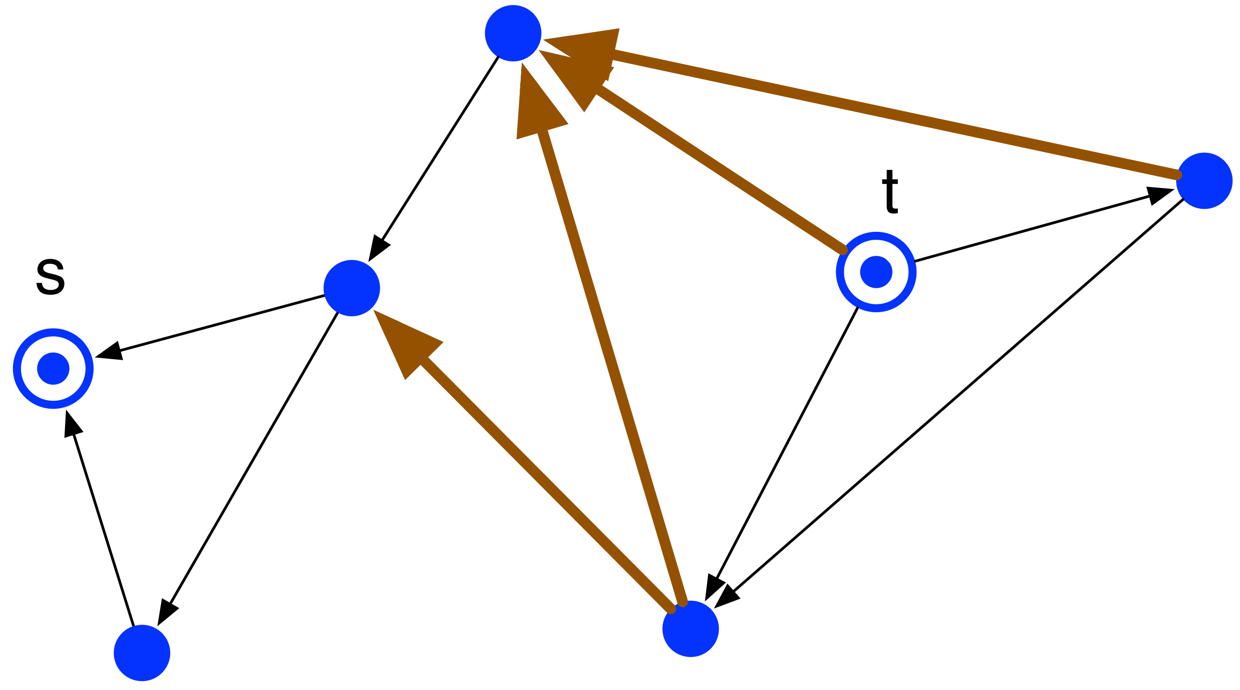
Forward Cuts

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



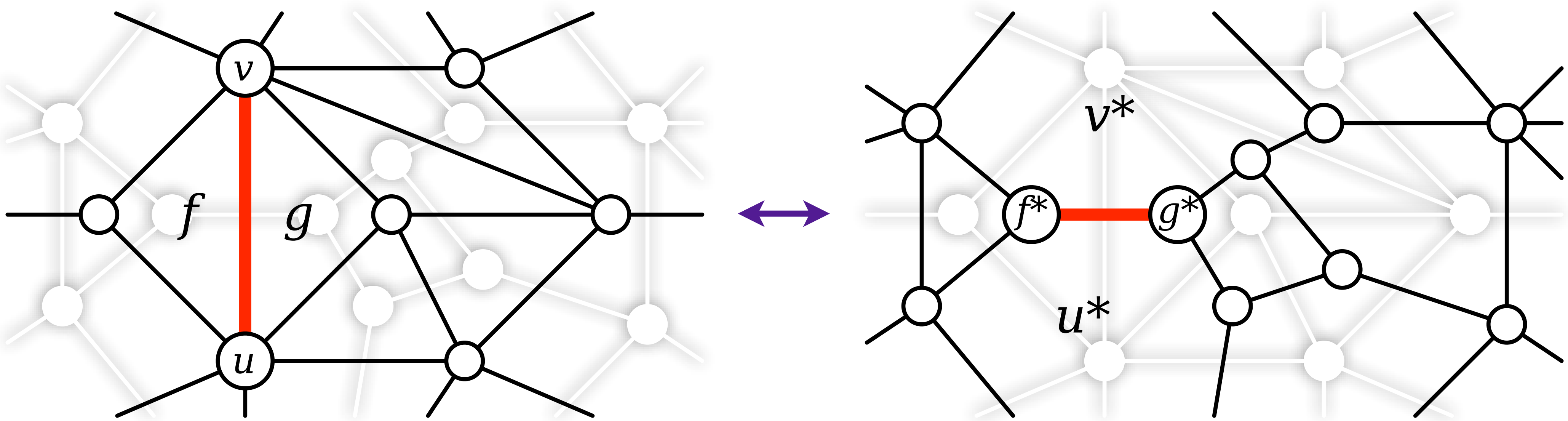
Forward Cuts

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



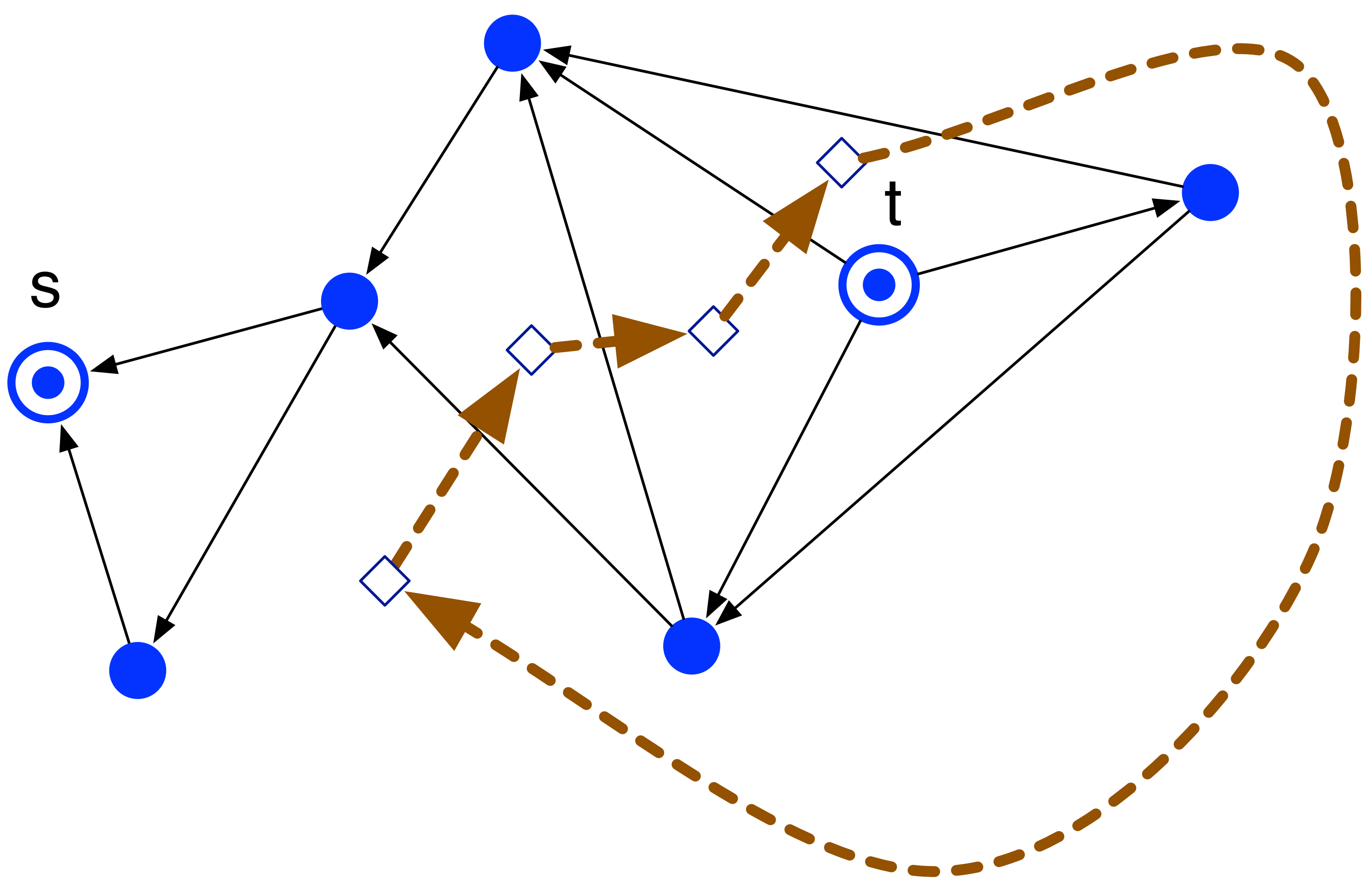
Dual Graphs

- Swap vertices and faces



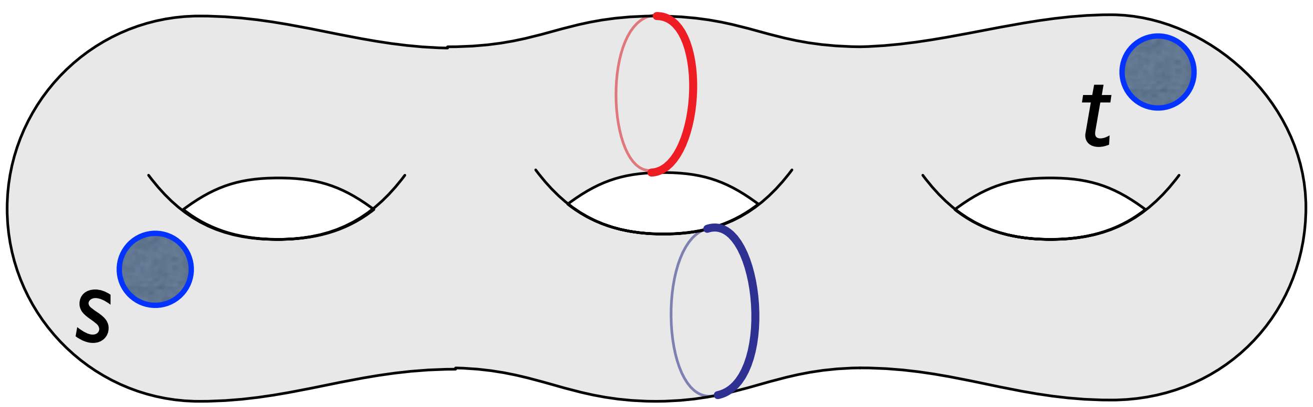
Planar Forward Cuts

- Forward cuts are dual to directed cycles separating *t* from *s*



Difficulties in the Surface

- Cuts may be dual to multiple cycles



- Can find cycles separating *t* from *s* in undirected graphs to compute minimum cuts [CEN '09; INSW '11, EFN '12]

Difficulties in the Surface

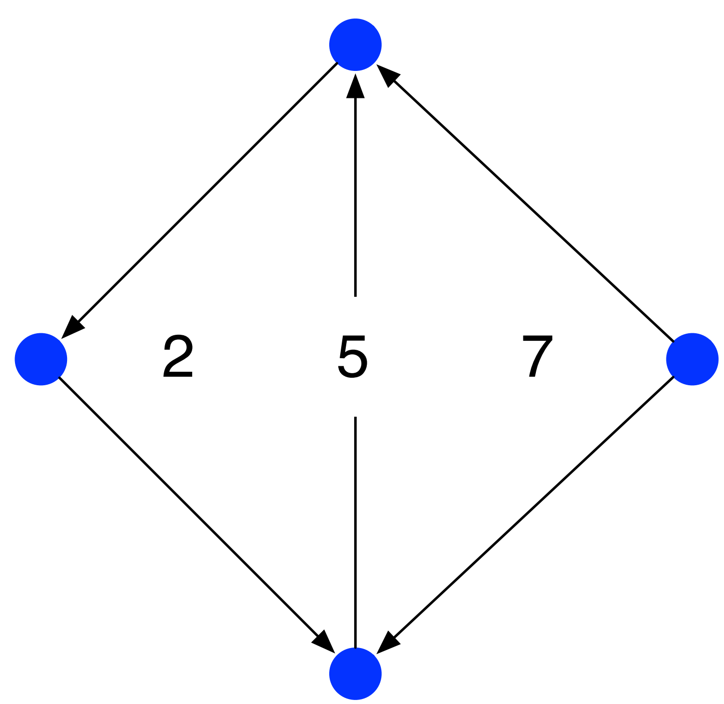
- Need to work directly with dual cycles in a directed graph
- Requires similar but more computationally difficult techniques than the undirected case
- Now need integer instead of $\mathbb{Z}_2$-homology
- Also new complications if DAG is not triangulated

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 v
- 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 $\partial\alpha(e) = \alpha(\mathit{right}(e)) - \alpha(\mathit{left}(e))$
- Boundary circulations are the boundaries of 2-chains



Homology

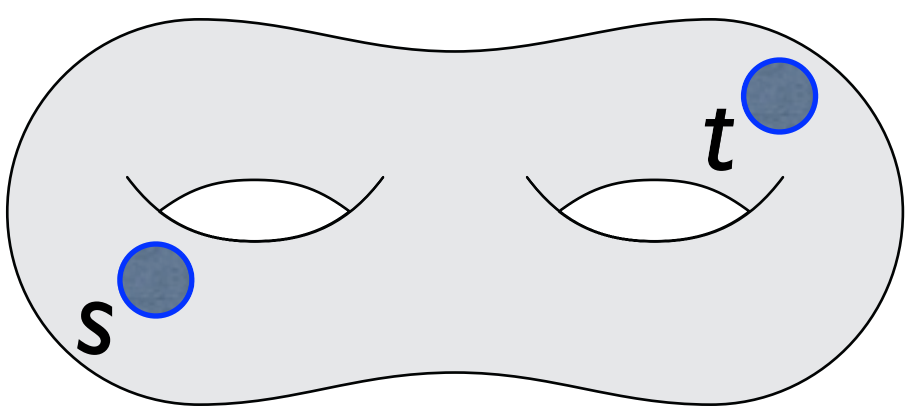
- Not all circulations are boundary circulations
- Circulations ϕ and Ψ are homologous if $\phi - \Psi$ is a boundary circulation
- An equivalence relation between circulations
- The vector space of homology classes is isomorphic to $\mathbb{Z}^{2g + \max\{0,b-1\}}$

Difficulties with Integer Homology

- Homology describe how cycles wrap around features in the surface
- Integer homology needed to understand directed cycles
- More efficient algorithms deal directly with $\mathbb{Z}_2$-homology where cycles have no direction
- $n^{O(g^2)}$ algorithm for minimum quotient cut uses integer homology [Patel '10]

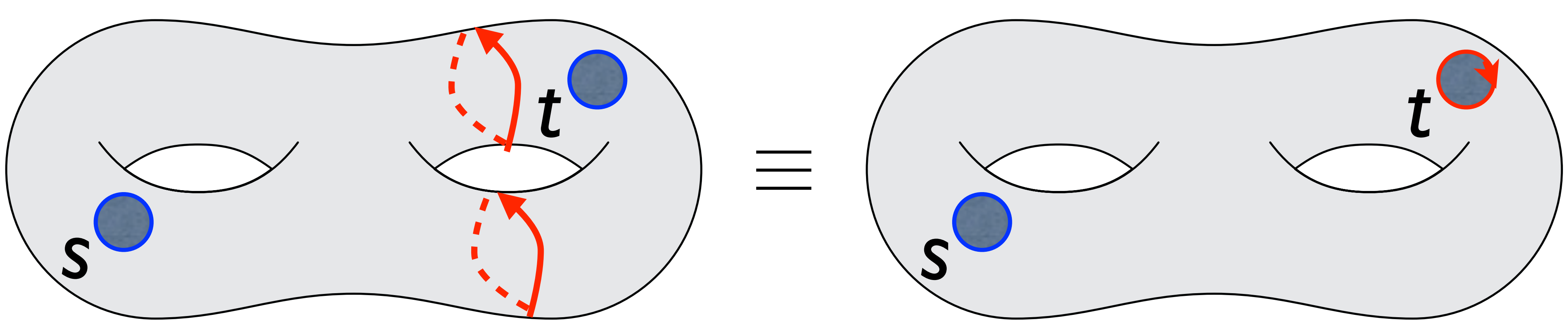
Dual of the DAG

- 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^*



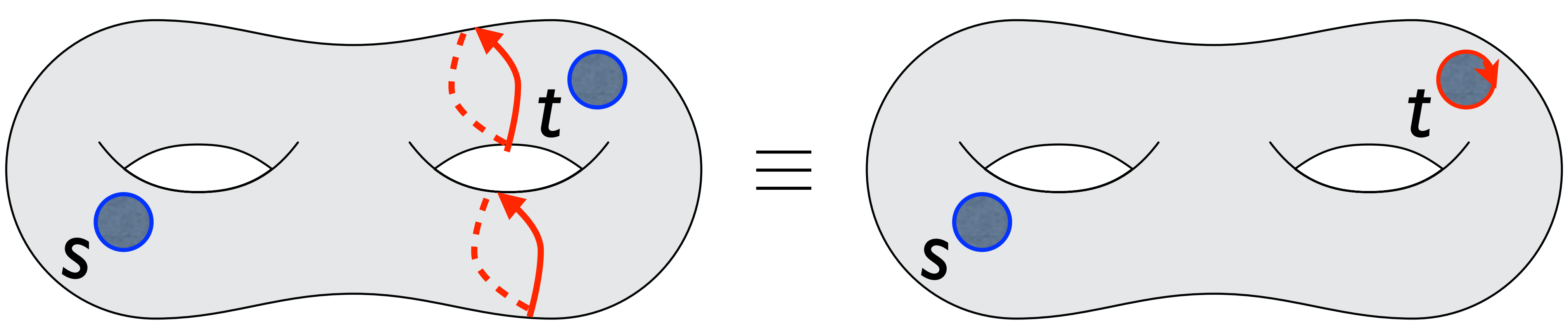
Dual of a Cut

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



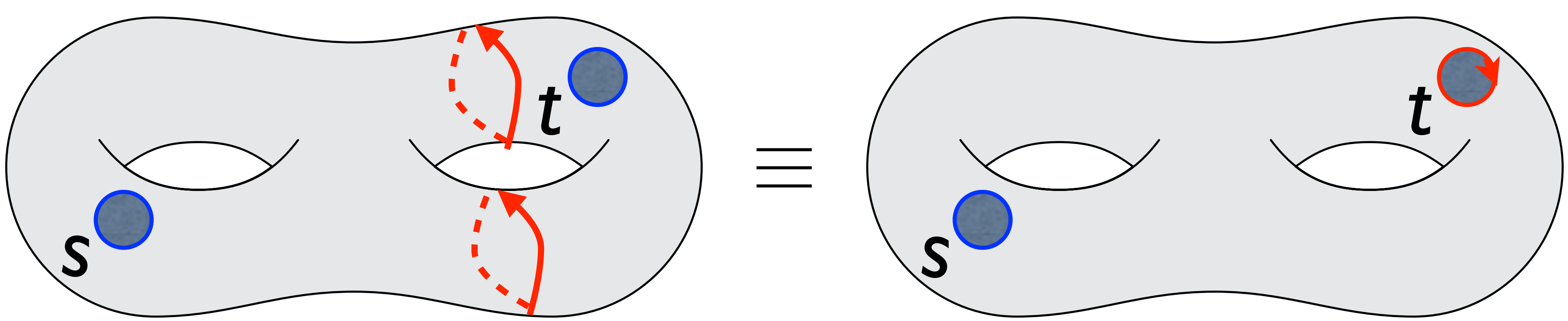
Dual of a Cut

- **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



Cuts as Cycles

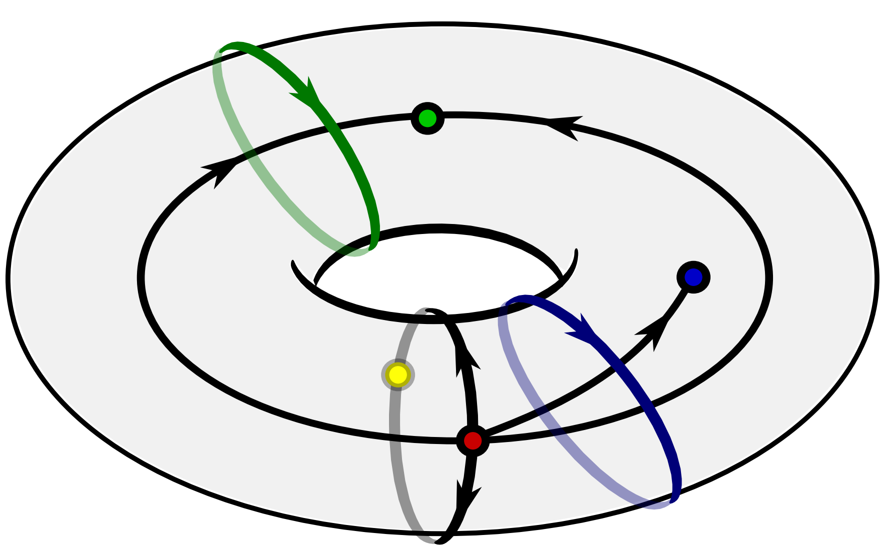
- There exists a bijection between forward *t*,*s*-cuts and (0,1)-circulations in the dual in a particular homology class
- Can represent (0,1)-circulations as collections of edge disjoint cycles

Counting Cycles

- Can modify prior techniques to count cycles generating circulations of a particular homology class [Kutz '06; Chambers *et al.* '08; Erickson, Nayyeri '11; BF '12]

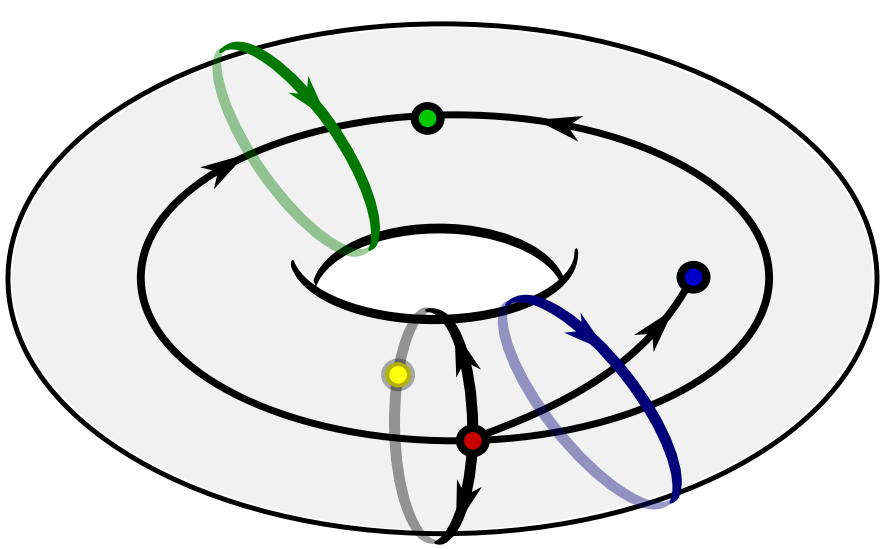
Polygonal Schema

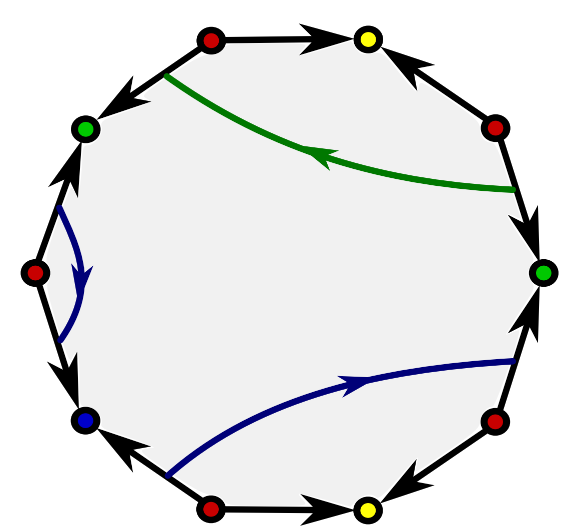
- Compute a system Λ of 2g loops and one directed path in the primal graph based at t
- Each loop made from two directed paths
- Directed paths cross any forward t,s-cut at most once



Polygonal Schema

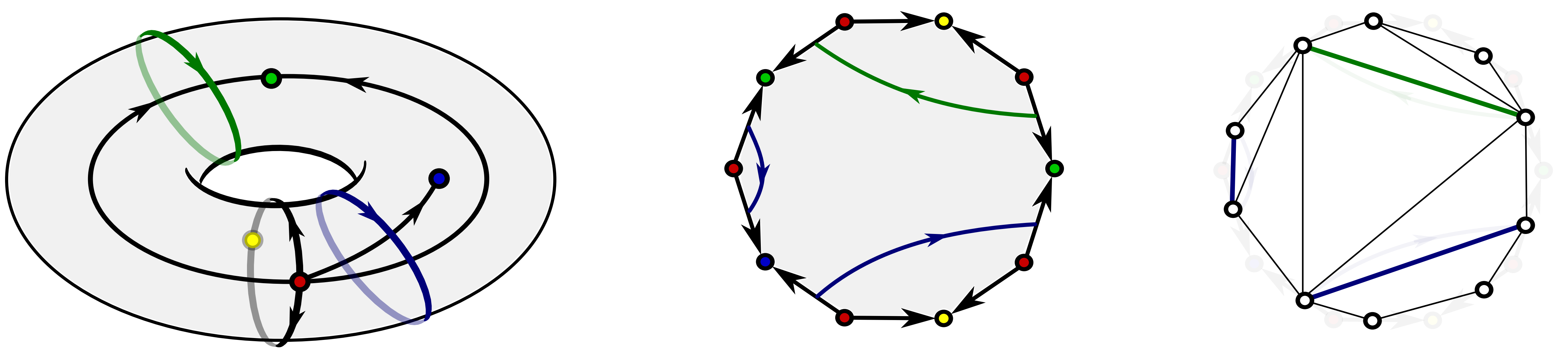
- Cutting along Λ turns the surface into a disk or **polygonal schema** with $8g + 2$ edges
- Duals of forward t,s-cuts cross paths as arcs in the schema





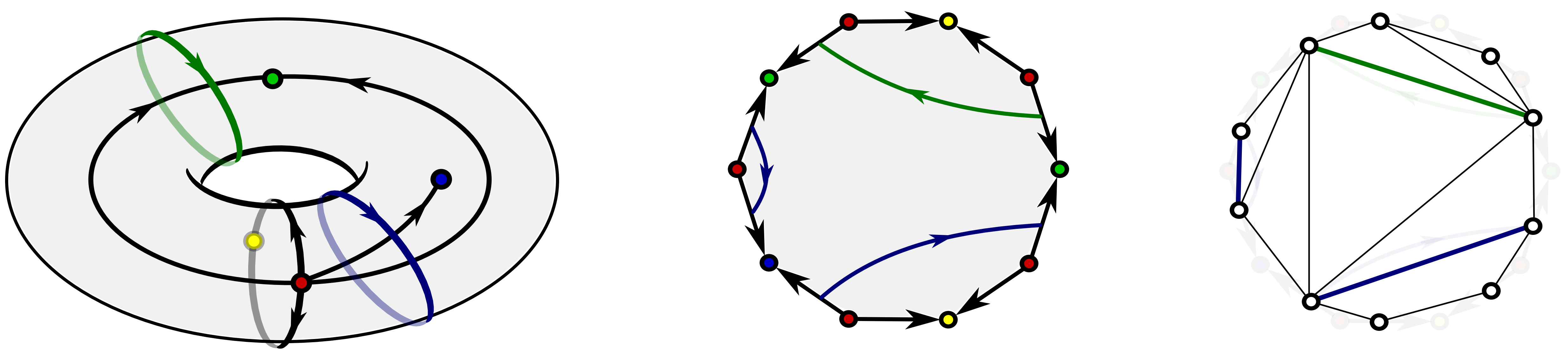
Polygonal Schema

- 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



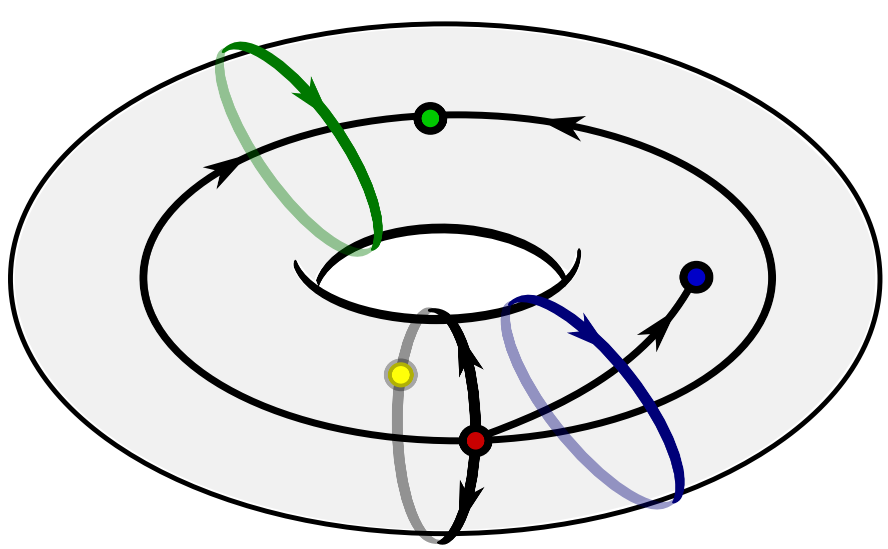
Crossing Sequences

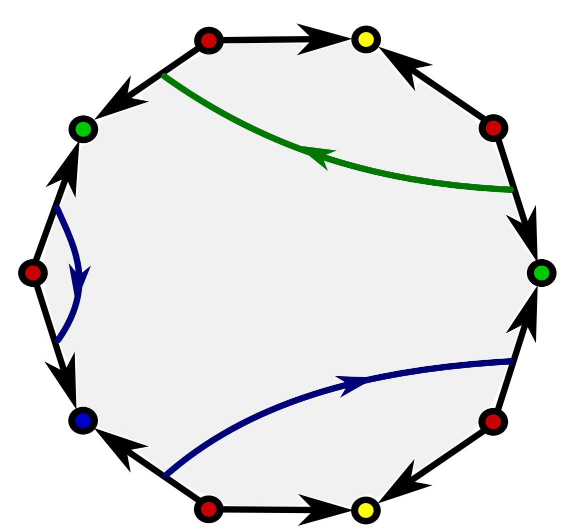
- Exist only $2^{O(g)}$ partial triangulations
- Each describes several **crossing sequences** between dual cycles and the system Λ

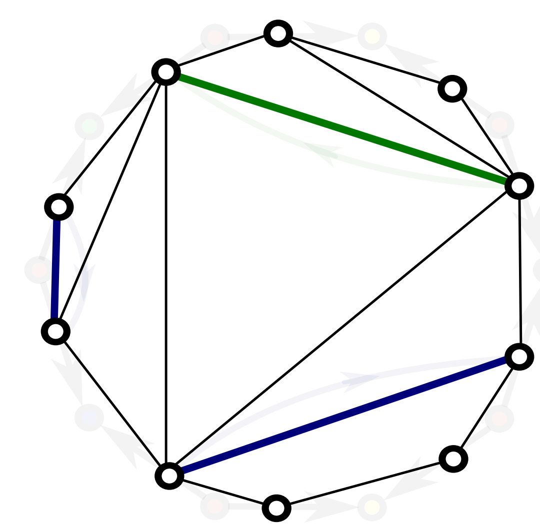


Crossing Sequences

- 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 subsets of the **universal covering space**

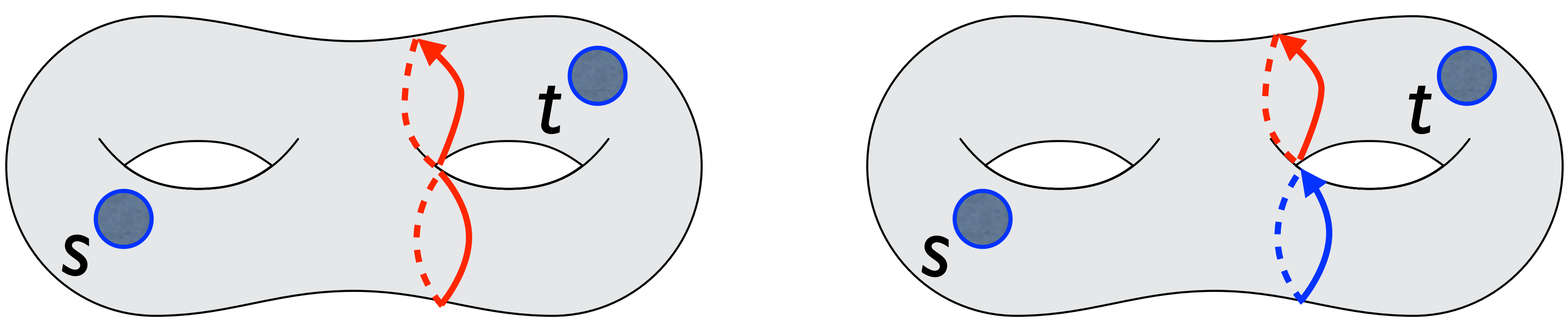






Overcounting

- Cycles in the correct homology class are edge disjoint
- But, distinct collections of edge disjoint cycles may contain the same set of edges and trivially generate the same circulation

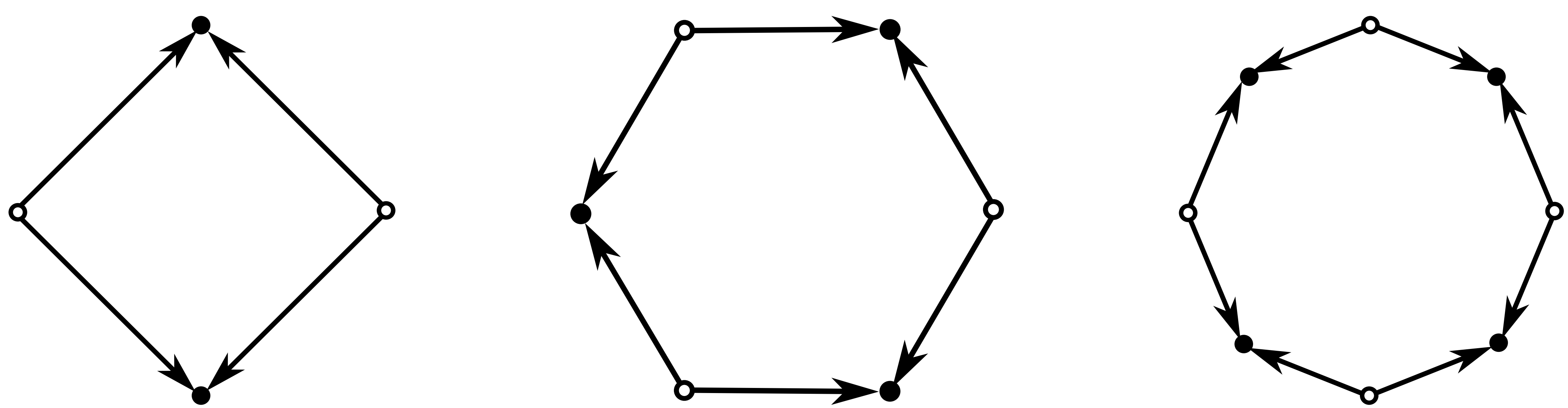


Triangulations Help

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

Triangulating is Hard

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

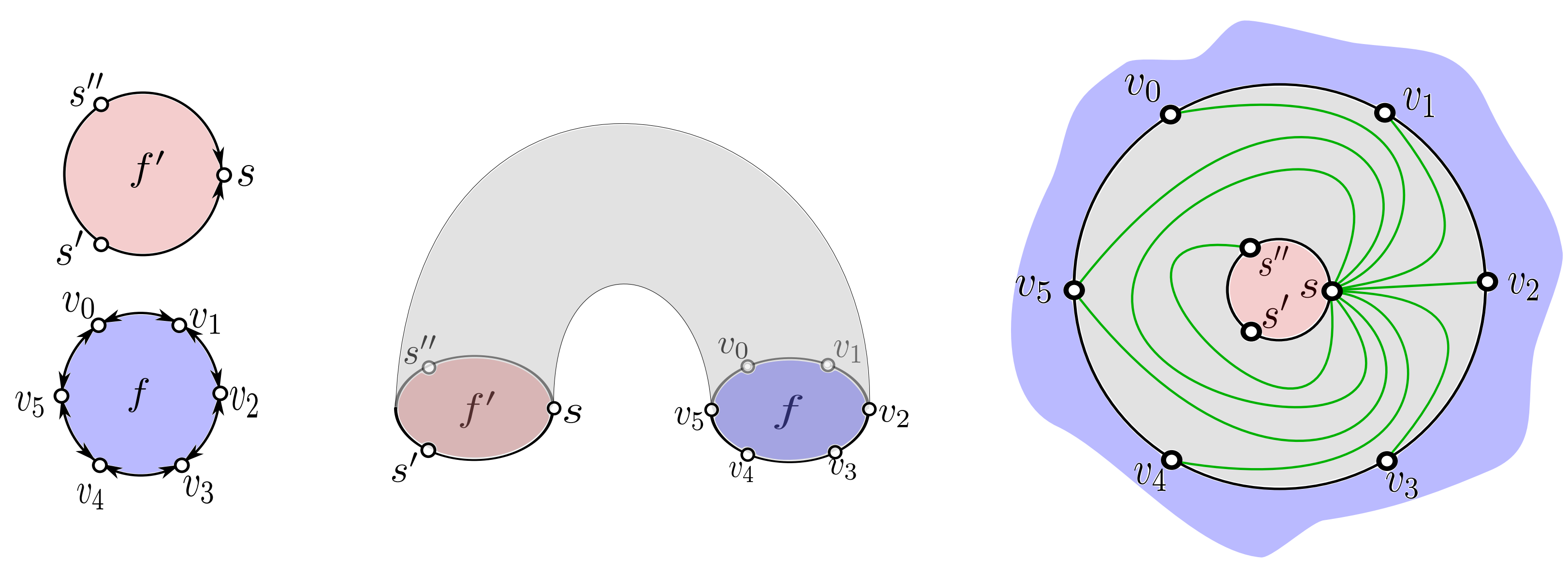


Limited Irreducibility

- **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

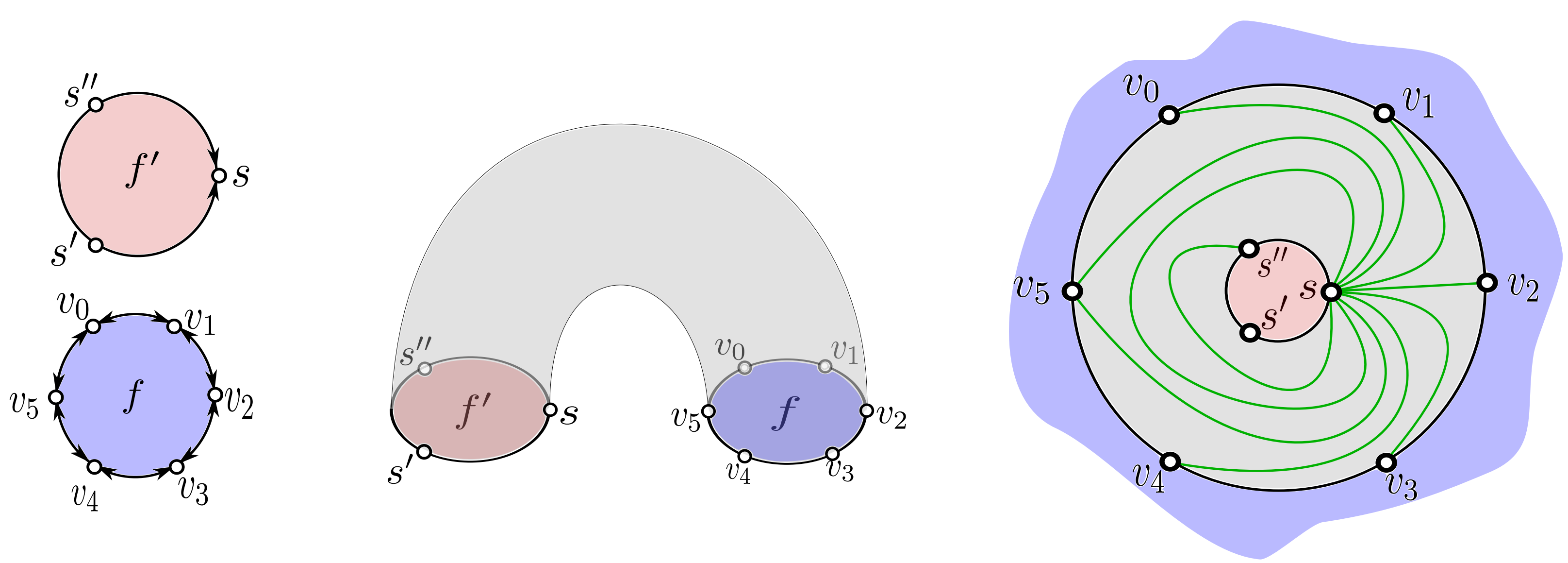
Adding Handles

- Iteratively add a handle between each irreducible face *f* and any face *f'* incident to *s*
- Add edges with head *s* to triangulate the handle



Adding Handles

- Genus of surface and number of edges increases by O(g)
- Can now safely apply cycle counting algorithm



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