Dynamic Generators of Topologically Embedded Graphs

David Eppstein

Univ. of California, Irvine
School of Information and Computer Science

Outline

New results and related work

Review of topological graph theory

Solution technique: tree-cotree decomposition

Outline

New results and related work

Review of topological graph theory

Solution technique: tree-cotree decomposition

New Results

Given cellular embedding of graph on surface,
construct and maintain generators of fundamental group

Speed up other dynamic graph algorithms for
topologically embedded graphs (connectivity, MST)

Improve constant in separator theorem for low-genus graphs

Construct low-treewidth tree-decompositions of
low-genus low-diameter graphs

New Results: Fundamental Group Generators

Fundamental group is formed by loops on surface
Provides important topological information about surface,
used as basis for all our other algorithms

Can be described by a system of generators (independent loops)
and relations (concatenations of loops that bound disks)

Time to construct this system: $O(n)$

Related work:

Canonical schema of Vegter and Yap [SoCG 90]
(set of generators satisfying prespecified relations)

Time to construct canonical schema: $O(gn)$

New Results: Dynamic Graph Algorithms

Classify update operations on embedded graphs
including updates that change the surface topology

Maintain generators, graph connectivity, surface classification
Time: $O(\log n + \log g\,(\log \log g)^3)$ per update

Maintain minimum spanning tree
Time: $O(\log n + (\log g)^4)$ per update

Related work:

Dynamic plane graphs $O(\log n)$ per update [EITTWY, SODA 90]

Dynamic connectivity $O(\log n\,(\log \log n)^3)$, MST $O((\log n)^4)$
for arbitrary graphs [Holm et al, J. ACM 01; Thorup, STOC 00]

New Results: Improved Separator Theorem

Separator = small set of vertices, removal of which
partitions graph into pieces of size ≤ *cn*, for some *c* < 1

For oriented genus-*g* graphs,
separator size ≤ sqrt(8*gn*+O(*n*))

For unoriented graphs, size ≤ sqrt(4*gn*+O(*n*))

(Constant in O(*n*) comes from planar separator theorem)

Related work:
Much research on planar separator theorems

Previous best bound on separator size: sqrt(16*gn*+O(*n*))
[Aleksandrov and Djidjev, SIAM J. Discrete Math. 1996]

New Results: Tree-Decomposition

Various definitions, e.g.
tree-decomposition = representation of graph
as a subgraph of a chordal graph

tree-width = (max clique size of chordal graph) - 1

If graph has genus g and diameter D,
can construct tree-decomposition of width $O(gD)$
in time $O(gDn)$

Related work:

We previously showed that a tree-decomposition
with this width exists [Eppstein, Algorithmica 00]
but did not provide an algorithm for finding it.

Outline

New results and related work

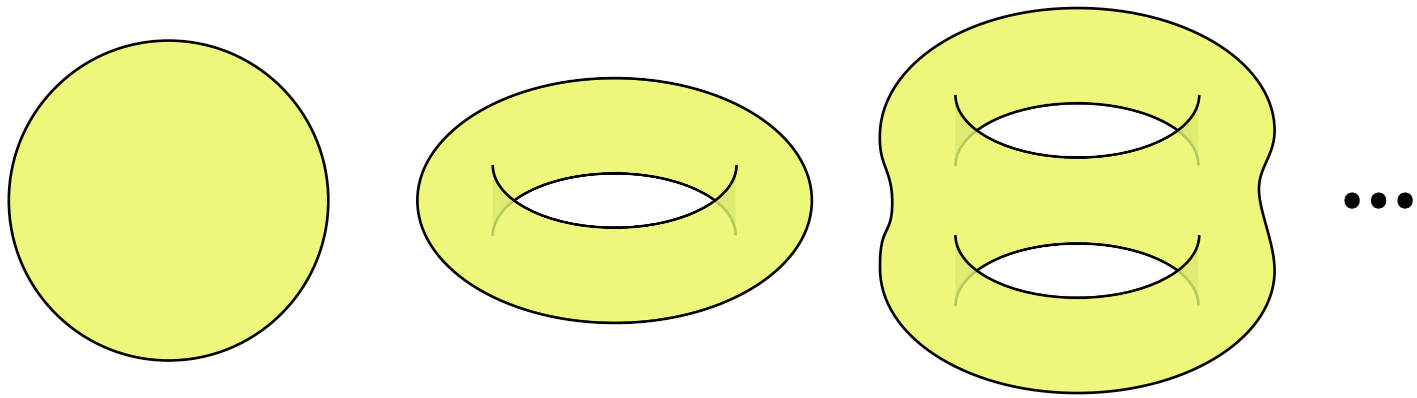
Review of topological graph theory

Solution technique: tree-cotree decomposition

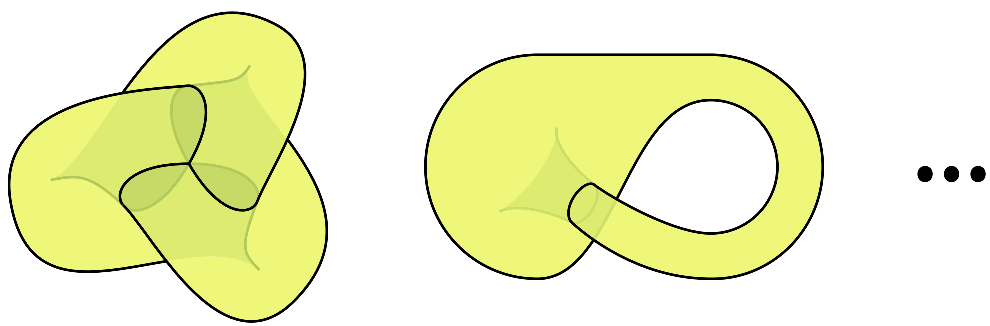
Classification of surfaces

Surface (2-manifold) = topological space in which each point has a neighborhood topologically equivalent to a disk

Oriented surfaces: sphere, *k*-torus



Unoriented surfaces: sphere with *k* cross-caps
k=1: projective plane, *k*=2: Klein bottle, ...



Graph embeddings

Embedding of graph *G* on surface *S*:
map vertices to points, edges to curves
s.t. no two edges meet except at endpoints

Faces of embedding = components of *G* - *S*
may have complicated topology

Cellular embedding: all faces are disks

Any embedding can be converted to a cellular
embedding with smaller or equal genus

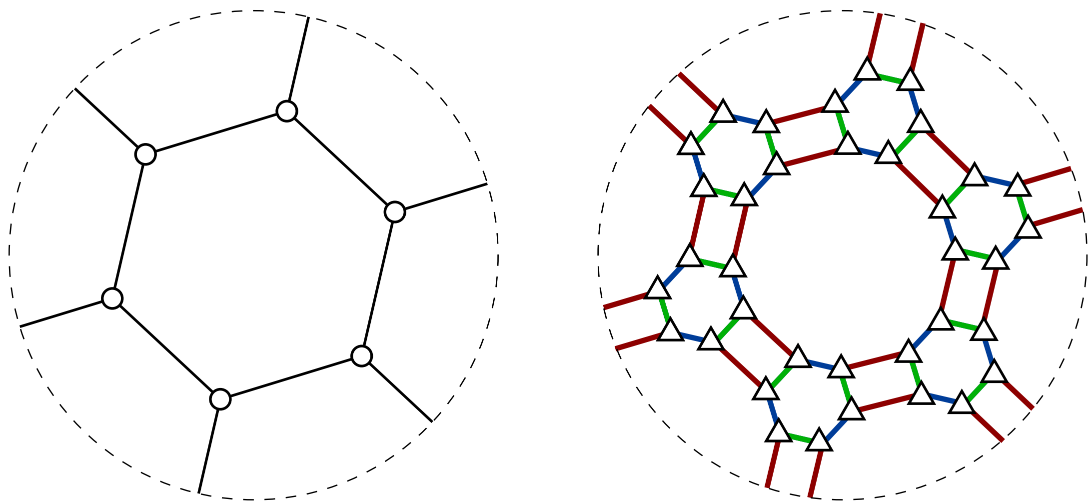
In this paper, all embeddings cellular
(other embeddings could be handled by
viewing *G* as a subgraph of a larger
cellularly embedded graph)

Gem Representation

[Lins, J. Combinatorial Th. 1982]

Given cellular embedding of graph G,
a flag is a pairwise adjacent set (vertex, edge, face)

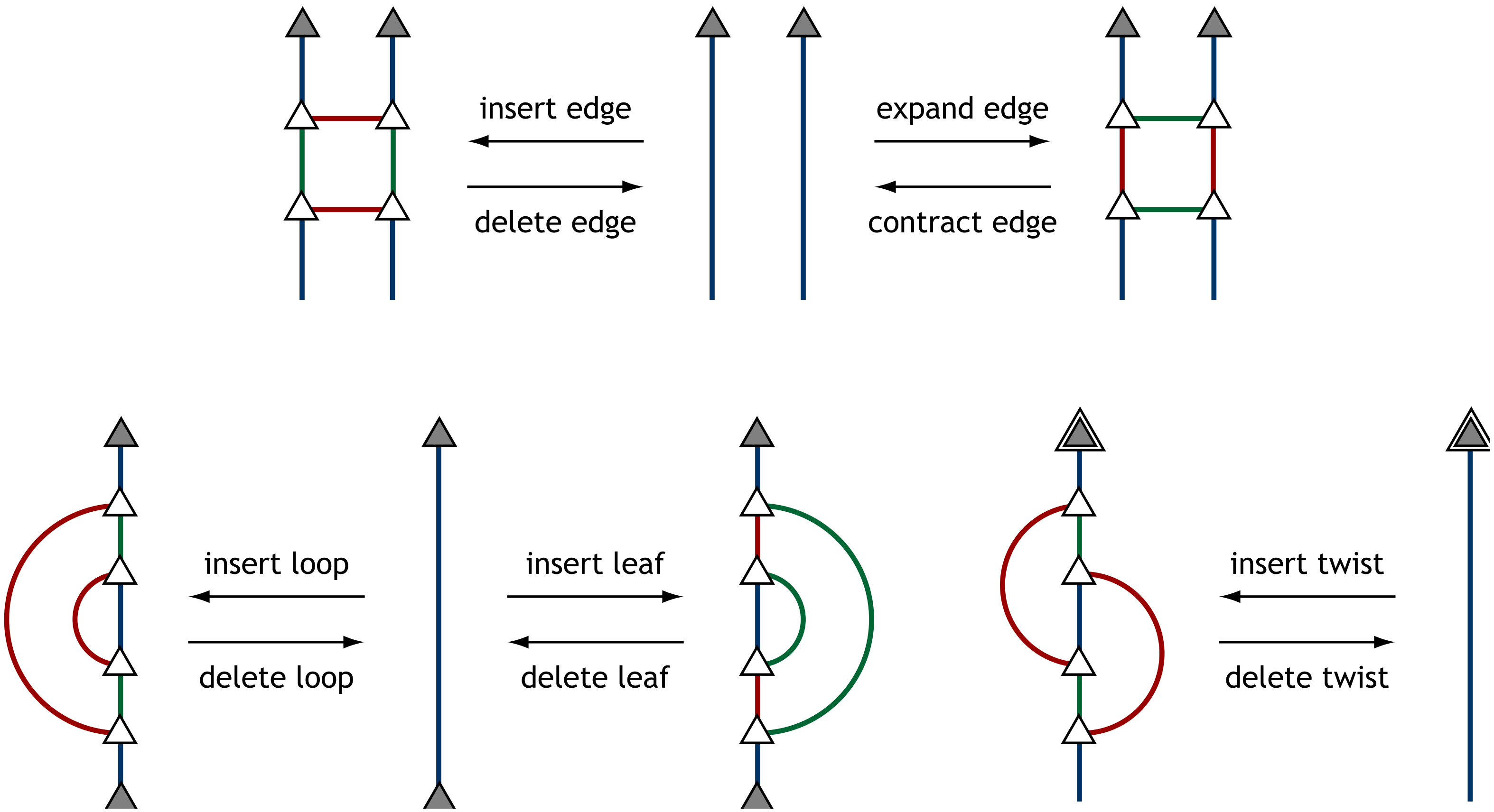
Form gem graph H, vertices(H) = flags,
two flags adjacent if their sets intersect in two elements



Each flag has one adjacency of each type
(change vertex, change edge, change face)

G and its embedding determined by H and its edge types
Permuting edge types forms dual embedding

Dynamic graph operations (and their duals) translated to Gem representation



edge to be inserted is specified by flags for its two endpoints
(shown as gray triangles in figure)

Effects of dynamic graph updates on surface topology

cell-merging insertion:

both endpoints of inserted edge belong to different faces
surface topology is changed by adding a handle

cell-splitting insertion:

both endpoints of inserted edge belong to same face
surface topology is unchanged

cell-twisting insertion:

both endpoints of inserted edge belong to same face,
but with inconsistent orientations
surface topology is changed by adding a crosscap

similar classification for deletions, expansions, and contractions...

Outline

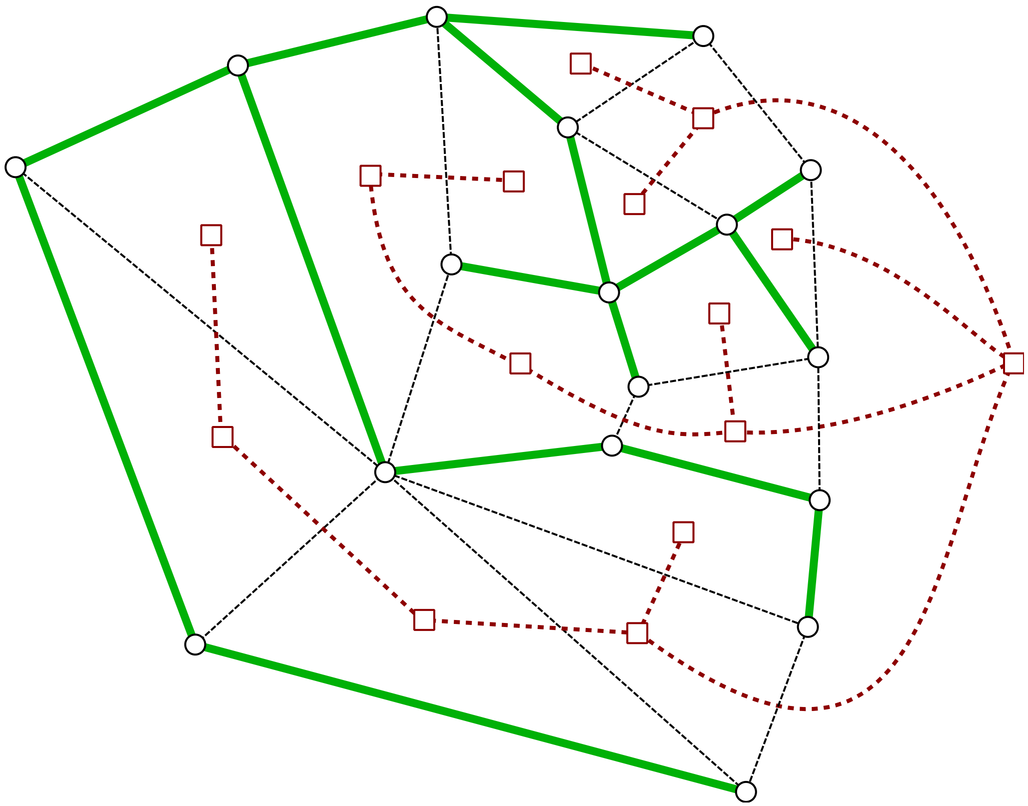
New results and related work

Review of topological graph theory

Solution technique: tree-cotree decomposition

Interdigitating trees

In any planar graph, dual complement of a spanning tree
is also a spanning tree of the dual graph



in particular, dual complement of minimum spanning tree
is maximum spanning tree of dual graph
used e.g. by [EITTWY SODA 90] for dynamic plane graph algorithms

Interdigitating trees on higher-genus surfaces

By Euler's formula, primal and dual spanning trees
don't have enough edges to cover whole graph

However...

Primal spanning tree doesn't separate dual

So...

For any primal spanning tree,
there is a disjoint dual spanning tree
(if primal = MinST, dual can be MaxST)

leftover edges = O(g)

(primal tree, dual tree, leftover edges) = tree-cotree decomposition

Solution strategy for dynamic graph problems

Maintain dynamic trees [Sleator & Tarjan, JCSS 1983]
for primal spanning tree and complementary dual spanning tree

Use auxiliary data structures to classify update type

Contract leaves and long paths in trees
except at endpoints of leftover edges
[Separator based sparsification, EGIS, JCSS 1996]

Use general-purpose dynamic graph algorithm
on union of contracted tree and leftover edges
to find replacement edges in trees after updates
[Holm et al, J. ACM 2001; Thorup, STOC 00]

Recover desired additional info (e.g. surface orientation)
by translating into queries on trees

Solution strategy for separator & tree-decomposition

Form breadth-first search tree,
find complementary dual tree

(Separator only:) Cut BFS tree every O(sqrt n) levels
with careful choice of starting level

Cut from each leftover edge to root of BFS tree
or higher cut level

Remaining graph is planar,
apply known techniques for planar graphs

Conclusions

Several new algorithms for low-genus graphs

Allow genus to be non-constant and varying

Tree-cotree decomposition is useful fundamental tool