Minimum cycle and homology bases of surface embedded graphs

Glencora Borradaile
Oregon State University

Kyle Fox
Duke University

Erin Wolf Chambers
St. Louis University

Amir Nayyeri
Oregon State University
Cycles
Cycles
Cycles
Cycles

"even-degree subgraph"
Cycle Space

• Vector space isomorphic to $\mathbb{Z}_2^{m-n+1}$ on graphs with $n$ vertices and $m$ edges
Cycle Space

- Vector space isomorphic to $\mathbb{Z}_2^{m-n+1}$ on graphs with $n$ vertices and $m$ edges

- A *cycle basis* is a maximal set of independent cycles
Cycle Space

- Vector space isomorphic to $\mathbb{Z}_2^{m-n+1}$ on graphs with $n$ vertices and $m$ edges
- A cycle basis is a maximal set of independent cycles
- We want the minimum cycle basis, the one with the minimum number of edges (or of minimum total weight)
## Highlights

<table>
<thead>
<tr>
<th>How fast?</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(m^3 n)$</td>
<td>[Horton ’87]</td>
</tr>
<tr>
<td>$O(m^3 + mn^2 \log n)$</td>
<td>[de Pina ’95]</td>
</tr>
<tr>
<td>$O(n^\omega)$ (randomized)</td>
<td>[Amaldi et al. ’09]</td>
</tr>
<tr>
<td>$O(nm^2 / \log n + n^2 m)$</td>
<td>[Mehlhorn, Michail ’09]</td>
</tr>
</tbody>
</table>
Planar Graphs
Planar Graphs
Planar Graphs
Planar Graphs
Planar Graphs
Planar Graphs
Planar Graphs
Planar Graphs
Planar Highlights

<table>
<thead>
<tr>
<th>How fast?</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n^2 \log n)$</td>
<td>[Hartvigsen, Mardon ’94]</td>
</tr>
<tr>
<td>$O(n^2)$</td>
<td>[Amaldi et al. ’09]</td>
</tr>
<tr>
<td>$O(n \log^4 n)$ (rand. oracle)</td>
<td>[Borradaile et al. ’15]</td>
</tr>
<tr>
<td>$O(n \log^3 n)$ (rand. oracle)</td>
<td>[Borradaile et al. (40 minutes ago)]</td>
</tr>
</tbody>
</table>
Planar Highlights

<table>
<thead>
<tr>
<th>How fast?</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n^2 \log n)$</td>
<td>[Hartvigsen, Mardon ’94]</td>
</tr>
<tr>
<td>$O(n^2)$</td>
<td>[Amaldi et al. ’09]</td>
</tr>
<tr>
<td>$O(n \log^4 n)$ (rand. oracle)</td>
<td>[Borradaile et al. ’15]</td>
</tr>
<tr>
<td>$O(n \log^3 n)$ (rand. oracle)</td>
<td>[Borradaile et al. (40 minutes ago)]</td>
</tr>
</tbody>
</table>

All results also compute minimum cut basis in the dual graph.
Embedded Graphs
Embedded Graphs

Algorithms often parameterized by genus $g$
• Minimum cut oracle in $2^{O(g^2)} n \log^3 n$ time
  [Borradaile et al. (40 minutes ago)]
• Minimum cut oracle in $2^O(g^2) n \log^3 n$ time
  [Borradaile et al. (40 minutes ago)]

• But cycles are not necessarily dual to cuts when $g > 0$
• Minimum cut oracle in $2^{O(g^2)} n \log^3 n$ time
  [Borradaile et al. (40 minutes ago)]

• But cycles are not necessarily dual to cuts when $g > 0$

• So there is still have work to do on computing minimum cycle bases
Crossing Cycles
Crossing Cycles
Homology

boundary cycle
Homology

homologous
Homology Space

- Vector space isomorphic to $\mathbb{Z}_2^{2g}$ in graphs embedded on an orientable surface of genus $g$
Homology Space

- Vector space isomorphic to $\mathbb{Z}_2^{2g}$ in graphs embedded on an orientable surface of genus $g$

- A **homology basis** is a maximal set of *cycles* in independent homology classes
Homology Space

• Vector space isomorphic to $\mathbb{Z}_2^{2g}$ in graphs embedded on an orientable surface of genus $g$

• A **homology basis** is a maximal set of *cycles* in independent homology classes

• We want the **minimum homology basis**, the one with the minimum number of edges (or of minimum total weight)
### Highlights

<table>
<thead>
<tr>
<th>How fast?</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n^2 \log n + gn^2 + g^3 n)$</td>
<td>[Erickson, Whittlesey ’05]</td>
</tr>
<tr>
<td>$2^{O(g)} n \log n$</td>
<td>[Erickson, Nayyeri ’11]</td>
</tr>
<tr>
<td>$g^{O(g)} n \log \log n$</td>
<td>[Italiano et al. ’11]</td>
</tr>
</tbody>
</table>
# Highlights

<table>
<thead>
<tr>
<th>How fast?</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n^2 \log n + gn^2 + g^3 n)$</td>
<td>[Erickson, Whittlesey '05]</td>
</tr>
<tr>
<td>$2^{O(g)} n \log n$</td>
<td>[Erickson, Nayyeri '11]</td>
</tr>
<tr>
<td>$g^{O(g)} n \log \log n$</td>
<td>[Italiano et al. '11]</td>
</tr>
</tbody>
</table>

Cheapest non-boundary cycle in $O(g^2 n \log n)$ [Cabello et al. '13]
Our Results

- A deterministic $O(n^\omega + 2^{2g}n^2)$ time algorithm for minimum cycle basis in genus $g$ graphs
Our Results

- A deterministic $O(n^\omega + 2^{2g} n^2)$ time algorithm for minimum cycle basis in genus $g$ graphs
  - Matches randomized algorithm of [Amaldi et al. ’09] when $g$ is constant
Our Results

- A \textit{deterministic} $O(n^\omega + 2^{2g}n^2)$ time algorithm for minimum cycle basis in genus $g$ graphs
  - Matches randomized algorithm of [Amaldi et al. '09] when $g$ is constant

- A randomized $O(g^3 n \log n)$ time algorithm for minimum homology basis in genus $g$ graphs
Our Results

• A deterministic $O(n^\omega + 2^{2g}n^2)$ time algorithm for minimum cycle basis in genus $g$ graphs
  - Matches randomized algorithm of [Amaldi et al. ’09] when $g$ is constant

• A randomized $O(g^3n \log n)$ time algorithm for minimum homology basis in genus $g$ graphs
  - First with polynomial dependence on $g$ and near-linear dependence on $n$
Our Results

⭐ A deterministic $O(n^\omega + 2^{2g}n^2)$ time algorithm for minimum cycle basis in genus $g$ graphs

- Matches randomized algorithm of [Amaldi et al. ’09] when $g$ is constant

• A randomized $O(g^3n \log n)$ time algorithm for minimum homology basis in genus $g$ graphs

  - First with polynomial dependence on $g$ and near-linear dependence on $n$
de Pina’s Algorithm

- Iteratively adds cycles $\gamma_1, \ldots, \gamma_{m-n+1}$ to the basis
de Pina’s Algorithm

- Iteratively adds cycles $\gamma_1, \ldots, \gamma_{m-n+1}$ to the basis

- Assigns each cycle $\gamma$ an $(m-n+1)$-bit vector $[\gamma]$ called the *cycle signature*
de Pina’s Algorithm

- Iteratively adds cycles $\gamma_1, \ldots, \gamma_{m-n+1}$ to the basis
- Assigns each cycle $\gamma$ an $(m-n+1)$-bit vector $[\gamma]$ called the *cycle signature*
- Maintains *support vectors* $S_1, \ldots, S_{m-n+1}$
de Pina’s Algorithm

- Iteratively adds cycles $\gamma_1, \ldots, \gamma_{m-n+1}$ to the basis

- Assigns each cycle $\gamma$ an $(m-n+1)$-bit vector $[\gamma]$ called the cycle signature

- Maintains support vectors $S_1, \ldots, S_{m-n+1}$

- $\gamma_j$ is the cheapest simple cycle $\gamma$ where $\langle S_j, [\gamma] \rangle = 1$
de Pina’s Algorithm

- Iteratively adds cycles $\gamma_1, \ldots, \gamma_{m-n+1}$ to the basis
- Assigns each cycle $\gamma$ an $(m-n+1)$-bit vector $[\gamma]$ called the *cycle signature*
- Maintains *support vectors* $S_1, \ldots, S_{m-n+1}$
- $\gamma_j$ is the cheapest simple cycle $\gamma$ where $\langle S_j, [\gamma] \rangle = 1$
- Naively takes $O(n^2)$ time to find $\gamma_j$
Isometric Cycles

• A simple cycle is *isometric* if it contains a shortest path between each pair of its vertices
Isometric Cycles

• A simple cycle is *isometric* if it contains a shortest path between each pair of its vertices

• Every minimum basis cycle is isometric \([\text{Hartvigsen, Mardon } '94]\)
Isometric Cycles

- A simple cycle is *isometric* if it contains a shortest path between each pair of its vertices.

- Every minimum basis cycle is isometric [Hartvigsen, Mardon '94]

- We consider the complete set of isometric cycles as candidates for each basis cycle $\gamma_j$. 
Homologous Isometric Cycles
Homologous Isometric Cycles
Homologous Isometric Cycles
Homologous Isometric Cycles
Homologous Isometric Cycles
Homologous Isometric Cycles
Region Trees

• Each homology class has $O(n)$ isometric cycles
Region Trees

• Each homology class has $O(n)$ isometric cycles

• These cycles induce the **region tree** for the class where each arc represents a cycle and each node represents the faces bounded by its arcs’ cycles
Region Trees

- Each homology class has $O(n)$ isometric cycles.
- These cycles induce the region tree for the class where each arc represents a cycle and each node represents the faces bounded by its arcs’ cycles.
Region Trees

- Each homology class has $O(n)$ isometric cycles.
- These cycles induce the *region tree* for the class where each arc represents a cycle and each node represents the faces bounded by its arcs’ cycles.
Region Trees

- Each homology class has $O(n)$ isometric cycles.
- These cycles induce the **region tree** for the class where each arc represents a cycle and each node represents the faces bounded by its arcs’ cycles.
Region Trees

- Each homology class has $O(n)$ isometric cycles.

- These cycles induce the **region tree** for the class where each arc represents a cycle and each node represents the faces bounded by its arcs’ cycles.
Region Trees

• Each homology class has $O(n)$ isometric cycles

• These cycles induce the region tree for the class where each arc represents a cycle and each node represents the faces bounded by its arcs’ cycles
• Compute \( \langle S_j, [\gamma] \rangle \) for every isometric cycle \( \gamma \) in a homology class using dynamic programming on the region tree
• Compute $\langle S_j, [\gamma] \rangle$ for every isometric cycle $\gamma$ in a homology class using dynamic programming on the region tree

• Takes $O(n)$ time per class; $O(2^{2g}n)$ time across all classes
• Compute $\langle S_j, [\gamma] \rangle$ for every isometric cycle $\gamma$ in a homology class using dynamic programming on the region tree

• Takes $O(n)$ time per class; $O(2^{2g} n)$ time across all classes

• $\gamma_j$ is the cheapest cycle $\gamma$ where $\langle S_j, [\gamma] \rangle = 1$
• Compute $\langle S_j, [\gamma] \rangle$ for every isometric cycle $\gamma$ in a homology class using dynamic programming on the region tree.

• Takes $O(n)$ time per class; $O(2^{2g}n)$ time across all classes.

• $\gamma_j$ is the cheapest cycle $\gamma$ where $\langle S_j, [\gamma] \rangle = 1$.

• Overall, $O(2^{2g}n^2)$ time spent picking basis cycles.
• \( O(n^\omega) \) time spent maintaining \( S_1, \ldots, S_{m-n+1} \)

[Kavitha et al. ’08]
- $O(n^\omega)$ time spent maintaining $S_1, \ldots, S_{m-n+1}$
  
  [Kavitha et al. ’08]

- So total running time is $O(n^\omega + 2^{2^g n^2})$
• $O(n^\omega)$ time spent maintaining $S_1, \ldots, S_{m-n+1}$
  [Kavitha et al. ’08]

• So total running time is $O(n^\omega + 2^{2g}n^2)$

• Reducing the time spent maintaining $S_1, \ldots, S_{m-n+1}$ means reducing the overall running time!
Homology Basis

- Minimum homology basis algorithm uses de Pina’s algorithm with \(2g\)-bit signatures instead
Homology Basis

- Minimum homology basis algorithm uses de Pina’s algorithm with $2g$-bit signatures instead.

- Searches extension of cyclic double cover [Erickson ’11] to find basis cycles in $O(g^2 n \log n)$ time each.
Thank you!
Thank you!