SAT Formulations for the Minimum Genus Problem

Ivo Hedtke (hedtke@uos.de)
Theoretical Computer Science, Uni Osnabrück
joint work of

Stephan Beyer\textsuperscript{1}
Markus Chimani\textsuperscript{1}
Ivo Hedtke\textsuperscript{1}
Michal Kotrbčík\textsuperscript{2}

\textsuperscript{1} Osnabrück University, Germany
\textsuperscript{2} Masaryk University, Czech Republic
Genus & Surface

$S_0, S_1, S_2, \ldots$

Genus 1

Genus 0

Genus 2

Genus 1

Genus 0
Graphs

Vertex $v \in V$:  

Edge $e = \{u, v\} \in E$:  

- finite
- undirected
- simple
- connected
- min degree 3
What are we talking about?

Embedding of a $G$ in $S_g$ = representation of $G$ in $S_g$ without edge crossings

Minimum Genus Problem: What is the smallest $g$ such that a given graph has an embedding in $S_g$?

Example: $K_5$ on the Torus
Why?

1. Minimum Genus is a useful parameter in algorithm design: take advantage of the topological structure.

Ref: [Erickson, Fox and Nayyeri (2012). Global minimum cuts in surface embedded graphs. SODA ’12, p. 1309-1318.]

2. Mathematical community: the genus of specific graph families is of interest ever since Ringel’s celebrated determination of the genus of complete graphs.

Ref: [Ringel (1974). Map Color Theorem. Springer-Verlag]
Stop talking! It’s already done!

Ref: [Mohar. Embedding graphs in an arbitrary surface in linear time. STOC ’96, p. 392–397]
There are also other algorithms.

All of them with problems.

“There appears to be no way to fix these problems without creating algorithms which take exponential time.” Ref: [Myrvold and Kocay (2011). Errors in graph embedding algorithms. J. of Comp. and Sy. Sc., 77(2):430–438]
Roadmap

1. reduce
2. SAT
3. faster SAT
4. use it
Some Facts

- linear time planarity test

- the genus is additive over biconnected components

- the genus problem is susceptible to non-planar core reduction

- Euler's formula: \( f \equiv |E| - |V| \mod 2 \).
**SAT: Idea**

rotation $\pi_v = \text{cyclic order of the neighbours}$

rotation system $\pi = (\pi_v)_{v \in V}$

Example: $\pi_v = (u_1 \ u_2 \ u_3)$

facial walk: $u_1 \rightarrow u_2 \rightarrow (\pi_{u_2}(u_1) =: u_3) \rightarrow (\pi_{u_3}(u_2) =: u_4) \rightarrow ...$
An Easy SAT to Start With: $\psi_{\text{simple}}$

each arc is in one face:
$$\neg (c^i_a \land c^j_a)$$
each face contains an arc:
$$\forall a \in A \ c^i_a$$
the successor in $\pi$ is in the same face:
$$p^v_{u,w} \rightarrow (c^i_{uv} \leftrightarrow c^i_{vw})$$
p forms a rotation system:
$$\forall u \in N(v), u \neq w \ p^v_{u,w}$$
$$\neg (p^v_{u,w} \land p^v_{u,w'})$$
$$\forall w \in N(v), w \neq u \ p^v_{u,w}$$
$$\neg (p^v_{u,w} \land p^v_{u,w'})$$
$$\forall u \in U, w \in N(v) \setminus U \ p^v_{u,w}$$
What can $\psi_{\text{simple}}$ do for us?

Contradiction: between clauses "each face contains an arc" and "successor in the same face"

$max: 5$ faces

$f = 6$

$2 - 2g = |V| - |E| + f$

$\Rightarrow g = 1$
\( f = 5 \)

\[ \begin{align*}
& (c_1^a \lor \overline{c_2^a}) \\
& \land (c_3^a \lor \overline{c_3^a}) \land (c_3^a \lor \overline{c_3^a}) \\
& \land (c_4^a \lor \overline{c_4^a}) \land (c_4^a \lor \overline{c_4^a}) \land (c_4^a \lor \overline{c_4^a}) \\
& \land (c_5^a \lor \overline{c_5^a}) \land (c_5^a \lor \overline{c_5^a}) \land (c_5^a \lor \overline{c_5^a}) \land (c_5^a \lor \overline{c_5^a})
\end{align*} \]

\( a \in f_{Il} \) \( \Rightarrow \) \( \|_{10} = 1011_{10} \)

<table>
<thead>
<tr>
<th>digit no.</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( b_a^i )</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>
SAT with Binary Face Indices

before ($\psi_{\text{simple}}$):

- $c^i_a$
- # clauses for $c^i_a$ = $f \times |A|$ = $O(f \times |A|^2)$

now ($\psi_{\text{bin}}$):

- $b^i_a$
- # clauses for $b^i_a$ = $\log f \times |A|$ = 0

+ some other changes ...

result: speed-up $100x$
result: running times increase 100x
Faster SAT

- binary face indices
- ordering around nodes
- degree 3 nodes
- face skipping
- incremental SAT
Press Play

- comparison to existing genus computations
  - complete graphs
  - circulant graphs
- real world graphs
Comparison $K_7$

- brute force enumeration scheme
- + special knowledge for complete graphs
- 896 hours

our SAT
- 1 hour

Comparison $C_{11}(1,2,4)$

3 pages theoretical analysis
+ MAGMA comp. algebra system
mental activity + 85 hours

our SAT + ILP
180 hours

Non-Planar Rome Graphs

<table>
<thead>
<tr>
<th>NPC</th>
<th>E/V</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>225</td>
<td>300</td>
</tr>
</tbody>
</table>

# instances: 1 2 3 4 5 6

% success: 0 10 20 30

# instances solved:

<table>
<thead>
<tr>
<th># instances</th>
<th>200</th>
<th>400</th>
<th>600</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

# solved instances:

<table>
<thead>
<tr>
<th># solved instances</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>400</td>
</tr>
</tbody>
</table>

% success:

<table>
<thead>
<tr>
<th>% success</th>
<th>0</th>
<th>25</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>30</td>
<td>45</td>
</tr>
</tbody>
</table>
Conclusions

first implementable general-purpose minimum genus algorithms
⇒ practical algorithms for small to medium-sized graphs

adaptable to tackle other related problems:
  - existence of polyhedral embeddings
  - embeddings with given face lengths (graph-theoretic models of carbon molecules)
  - non-oriented surfaces

our implementations cannot deal with too large graphs
SAT Formulations for the Minimum Genus Problem

Ivo Hedtke (hedtke@uos.de)

Thank you!