Simplification of Polyline Bundles

Joachim Spoerhase
Aalto University, Finland
Sabine Storandt
Universität Konstanz, Germany
Johannes Zink
Universität Würzburg, Germany
Motivation
Motivation

- Maps often consist of polylines
Motivation

• Maps often consist of polylines
Motivation

- Maps often consist of polylines

- Multiple polylines share edges and vertices sectionwise
Motivation

- Maps often consist of polylines
- Multiple polylines share edges and vertices sectionwise
Motivation

- Maps often consist of polylines
- Multiple polylines share edges and vertices sectionwise
- Reduce full data for zooming or schematization
Motivation

- Maps often consist of polylines
- Multiple polylines share edges and vertices sectionwise
- Reduce full data for zooming or schematization
Motivation

- Maps often consist of polylines
- Multiple polylines share edges and vertices sectionwise
- Reduce full data for zooming or schematization
Introduction: Simplifying a Polyline
Introduction: Simplifying a Polyline

Given:

• polyline $L$ as a sequence of points in the plane
• distance threshold $\varepsilon$
Introduction: Simplifying a Polyline

Given:
- polyline $L$ as a sequence of points in the plane
- distance threshold $\varepsilon$
Introduction: Simplifying a Polyline

Given: 
- polyline \( L \) as a sequence of points in the plane
- distance threshold \( \varepsilon \)

Goal: Find a minimum size subsequence \( L' \) of \( L \), such that the segment-wise undirected Hausdorff distance between \( L' \) and \( L \) does not exceed \( \varepsilon \).
Introduction: Simplifying a Polyline

Given: • polyline $L$ as a sequence of points in the plane
  • distance threshold $\varepsilon$

Goal: Find a minimum size subsequence $L'$ of $L$, such that
the segment-wise undirected Hausdorff distance
between $L'$ and $L$ does not exceed $\varepsilon$. 
Introduction: Simplifying a Polyline

Given: • polyline $L$ as a sequence of points in the plane
  • distance threshold $\varepsilon$

Goal: Find a minimum size subsequence $L'$ of $L$, such that the segment-wise undirected Hausdorff distance between $L'$ and $L$ does not exceed $\varepsilon$. 

\[ \leq \varepsilon \]
Introduction: Simplifying a Polyline

Given: • polyline $L$ as a sequence of points in the plane
• distance threshold $\varepsilon$

Goal: Find a minimum size subsequence $L'$ of $L$, such that the segment-wise undirected Hausdorff distance between $L'$ and $L$ does not exceed $\varepsilon$.

• Can be solved efficiently in $O(|L|^2)$ time.

[Imai, Iri ’88], [Chan, Chin ’96]
Introduction: Simplifying a Polyline

Given: • polyline $L$ as a sequence of points in the plane  
• distance threshold $\varepsilon$

Goal: Find a minimum size subsequence $L'$ of $L$, such that the segment-wise undirected Hausdorff distance between $L'$ and $L$ does not exceed $\varepsilon$.

• Can be solved efficiently in $O(|L|^2)$ time. 
  [Imai, Iri '88], [Chan, Chin '96]
Our Generalization
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
Our Generalization

• Given: a set \( \mathcal{L} = \{L_1, \ldots, L_\ell\} \) of polylines possibly sharing vertices and edges
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
Our Generalization

• Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices $V$
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal $\text{Min-Vertices}$: find a $V^* \subseteq V$
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal $\text{Min-Vertices}$: find a $V^* \subseteq V$

We call the union of their vertices $V$. 

![Diagram showing polylines $L_1$, $L_2$, and $L_3$.]
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal **Min-Vertices**: find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges.
- Goal **Min-Vertices**:
  find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal **Min-Vertices**: find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no $L'_i$ and $L_i$ exceeding the maximum distance
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_{\ell}\}$ of polylines possibly sharing vertices and edges
- Goal **Min-Vertices**: find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_{\ell}\}$ on $\mathcal{L}$, such that there is no $L'_i$ and $L_i$ exceeding the maximum distance $\leq \varepsilon$
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

- Goal **Min-Vertices**: find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no $L'_i$ and $L_i$ exceeding the maximum distance and the number of preserved vertices $|V^*|$ is minimum.

We call the union of their vertices $V$.
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- **Goal** Min-Vertices:
  find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no $L'_i$ and $L_i$ exceeding the maximum distance and the number of preserved vertices $|V^*|$ is minimum.
Our Generalization

• Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

• Goal $\text{MIN-VERTICES}$:
find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no $L'_i$ and $L_i$ exceeding the maximum distance and the number of preserved vertices $|V^*|$ is minimum.
Aren’t Both Goals the Same?
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:
Aren’t Both Goals the Same?

No! Here is a counterexample:

<table>
<thead>
<tr>
<th></th>
<th>Min-Vertices</th>
<th>Min-Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>base bundle</td>
<td>11 vertices</td>
<td>10 edges</td>
</tr>
<tr>
<td></td>
<td>9 vertices</td>
<td>6 edges</td>
</tr>
</tbody>
</table>
Aren’t Both Goals the Same?

No! Here is a counterexample:

<table>
<thead>
<tr>
<th></th>
<th>base bundle</th>
<th>optimal for Min-Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertices</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Edges</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>
Aren’t Both Goals the Same?

No! Here is a counterexample:

<table>
<thead>
<tr>
<th></th>
<th>Min-Vertices</th>
<th>Min-Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>base bundle</td>
<td>11 vertices</td>
<td>10 edges</td>
</tr>
<tr>
<td>optimal for Min-Vertices</td>
<td>8 vertices</td>
<td>7 edges</td>
</tr>
<tr>
<td>optimal for Min-Edges</td>
<td>9 vertices</td>
<td>6 edges</td>
</tr>
</tbody>
</table>
Why Not Simplifying Independently?
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded.
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

“simplifying” 3 polylines
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality

“simplifying” 3 polylines
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality

⇒ the total complexity might even increase
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

⇒ the total complexity might even increase

“simplifying” 3 polylines
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

⇒ the total complexity might even increase
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality

⇒ the total complexity might even increase
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

\[ \Rightarrow \text{misleading picture of the reality} \]

\[ \Rightarrow \text{the total complexity might even increase} \]
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality

⇒ the total complexity might even increase

14 vertices, 14 edges
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality

⇒ the total complexity might even increase

14 vertices, 14 edges

“simplifying” 6 polylines

“simplifying” 3 polylines

“simplifying” 6 polylines
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

“simplifying” 3 polylines

⇒ the total complexity might even increase

“simplifying” 6 polylines

14 vertices, 14 edges
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

⇒ misleading picture of the reality

⇒ the total complexity might even increase

14 vertices, 14 edges
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

⇒ the total complexity might even increase

14 vertices, 14 edges

“simplifying” 3 polylines

“simplifying” 6 polylines
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

⇒ the total complexity might even increase

14 vertices, 14 edges
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded
⇒ misleading picture of the reality

⇒ the total complexity might even increase

14 vertices, 14 edges
“simplifying” 3 polylines

14 vertices, 18 edges
“simplifying” 6 polylines
NP-Hardness
NP-Hardness

**Theorem 1:**
Simplifying a bundle of polylines is NP-hard for the goals \textsc{Min-Vertices} and \textsc{Min-Edges} even for 2 polylines.
**NP-Hardness**

**Theorem 1:**
Simplifying a bundle of polylines is NP-hard for the goals **Min-Vertices** and **Min-Edges** even for 2 polylines.

**Proof Sketch:**
NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals \text{Min-Vertices} and \text{Min-Edges} even for 2 polylines.

Proof Sketch:

• We reduce from \text{Max-2-SAT}.
NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals **Min-Vertices** and **Min-Edges** even for 2 polylines.

Proof Sketch:

• We reduce from **Max-2-SAT**.

• We use a polyline as a gadget for each literal in each clause. We connect them serially. → first polyline
NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals Min-Vertices and Min-Edges even for 2 polylines.

Proof Sketch:

• We reduce from Max-2-SAT.

• We use a polyline as a gadget for each literal in each clause. We connect them serially. → first polyline

• We use a polyline as a gadget for each variable and each clause. We connect them serially. → second polyline
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget
Variable-Gadget

Interpretation:
Variable-Gadget

Interpretation:
shortcut taken ⇔ variable set to true
Variable-Gadget

Interpretation:
shortcut taken $\iff$ variable set to false
shortcut taken $\iff$ variable set to true
Variable-Gadget

Interpretation:
shortcut taken ⇔ variable set to true
shortcut taken ⇔ variable set to false

literal gadgets
Variable-Gadget

Interpretation:
- shortcut taken $\iff$ variable set to true
- shortcut taken $\iff$ variable set to false

Literal gadgets
Variable-Gadget

Interpretation:
- Shortcut taken $\iff$ variable set to true
- Shortcut taken $\iff$ variable set to false

literal gadgets
Variable-Gadget

Interpretation:
- Shortcut taken \( \iff \) variable set to true
- Shortcut taken \( \iff \) variable set to false

Positive literal

Literal gadgets
Variable-Gadget

Interpretation:

shortcut taken ⇔ variable set to false

shortcut taken ⇔ variable set to true

positive literal

negative literal

literal gadgets
Variable-Gadget

Interpretation:

- shortcut taken $\iff$ variable set to true
- shortcut taken $\iff$ variable set to false

skipped $\iff$ literal satisfies its clause

positive literal

negative literal

literal gadgets

shortcut taken $\iff$ variable set to false

shortcut taken $\iff$ variable set to true
Variable-Gadget

Interpretation:

shortcut taken ⇔ variable set to true

shortcut taken ⇔ variable set to false

positive literal

negative literal

skipped ⇔ literal satisfies its clause

literal gadgets

literal satisfies its clause

ε

8
Variable-Gadget

Interpretation:

shortcut taken ⇔ variable set to true

shortcut taken ⇔ variable set to false

skipped ⇔ literal satisfies its clause

positive literal

negative literal

literal gadgets

negative literal

literal gadgets
Variable-Gadget

Interpretation:
- shortcut taken ⇔ variable set to true
- shortcut taken ⇔ variable set to false

skipped ⇔ literal satisfies its clause

positive literal

negative literal

literal gadgets

shortcut taken ⇔ variable set to false

8
Variable-Gadget

- shortcut taken $\iff$ variable set to true
- positive literal
- negative literal

skipped $\iff$ literal satisfies its clause

TRUE

shortcut taken $\iff$ variable set to true
Variable-Gadget

skipped ⇔ literal satisfies its clause

positive literal

negative literal

shortcut taken ⇔ variable set to false

FALSE
Clause-Gadget
Clause-Gadget
Clause-Gadget
Clause-Gadget
Clause-Gadget
Clause-Gadget
Clause-Gadget
Clause-Gadget
Clause-Gadget

literal gadgets
Clause-Gadget

literal gadgets
Clause-Gadget

Interpretation:

literal gadgets
Clause-Gadget

Interpretation:
skipped $\iff$ clause satisfied by this literal

literal gadgets
Clause-Gadget

**Interpretation:**
skipped ⇔ clause satisfied by this literal

```
skipped ⇔ clause satisfied by this literal
```

literal gadgets
Clause-Gadget

**Interpretation:**
skipped $\Leftrightarrow$ clause satisfied by this literal

none skipped $\Leftrightarrow$ clause remains unsatisfied
Full Example

\((x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\)
Full Example

\[
\begin{align*}
(x_1 \lor x_2) \land \\
(\neg x_1 \lor x_3) \land \\
(\neg x_3)
\end{align*}
\]
\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
\((x_1 \lor x_2) \land 
(\neg x_1 \lor x_3) \land 
(\neg x_3)\)
\[
\begin{align*}
(x_1 \lor x_2) &\land (\neg x_1 \lor x_3) \land (\neg x_3) \\
&\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad}$
(x₁ ∨ x₂) ∧
(¬x₁ ∨ x₃) ∧
(¬x₃)
\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
Full Example

\[(x_1 \lor \square x_2) \land
(\neg x_1 \lor x_3) \land
(\neg x_3)\]
(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)
$$\begin{align*}
(x_1 \lor x_2) \land \\
(\neg x_1 \lor x_3) \land \\
(\neg x_3)
\end{align*}$$
Full Example

\[(x_1 \lor x_2) \land
(\neg x_1 \lor x_3) \land
(\neg x_3)\]
Full Example

\[(x_1 \lor x_2) \land 
 (\neg x_1 \lor x_3) \land 
 (\neg x_3)\]
\[(x_1 \lor x_2) \land
(\neg x_1 \lor x_3) \land
(\neg x_3)\]
Full Example

\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
Full Example

\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
Full Example

\((x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\)
Full Example

\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
(x_1 \lor x_2) \land
(\neg x_1 \lor x_3) \land
(\neg x_3)
Full Example

\[(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)\]
\[(x_1 \lor x_2) \land \\
(\neg x_1 \lor x_3) \land \\
(\neg x_3)\]
\[(x_1 \lor x_2) \land \\
(\neg x_1 \lor x_3) \land \\
(\neg x_3)\]
(x_1 \lor x_2) \land (\neg x_1 \lor x_3) \land (\neg x_3)

We can even obtain APX-hardness by the reduction from MAX-2-SAT
Fixed-Parameter Tractability
Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals Min-Vertices and Min-Edges.
Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals Min-Vertices and Min-Edges.

Proof Sketch:
Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals \text{Min-Vertices} and \text{Min-Edges}.

Proof Sketch:

- Assume for each subset $V'$ of the shared vertices $V_{\text{shared}}$ that $V'$ is in the optimal solution and $V_{\text{shared}} \setminus V'$ is not.
Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals \textsc{Min-Vertices} and \textsc{Min-Edges}.

Proof Sketch:

- Assume for each subset $V'$ of the shared vertices $V_{\text{shared}}$ that $V'$ is in the optimal solution and $V_{\text{shared}} \setminus V'$ is not.
- Compute the simplification of the remaining (simple-polyline) sections in the classic way, e.g., with the algorithm by Chan and Chin.
Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals \textsc{Min-Vertices} and \textsc{Min-Edges}.

Proof Sketch:

- Assume for each subset $V'$ of the shared vertices $V_{\text{shared}}$ that $V'$ is in the optimal solution and $V_{\text{shared}} \setminus V'$ is not.

- Compute the simplification of the remaining (simple-polyline) sections in the classic way, e.g., with the algorithm by Chan and Chin.

- Running time in $O(2^k \cdot \ell n^2)$

\[ k := |V_{\text{shared}}|, \quad \ell := \# \text{ polylines}, \quad n := \# \text{ vertices} \]
Summary
Problem:
Simplify a set of polylines sharing some vertices and edges

**Goal 1:** Minimize the number of vertices
**Goal 2:** Minimize the number of edges
Summary

Problem:
Simplify a set of polylines sharing some vertices and edges

Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

• Generalizes the well-known problem of simplifying a single polyline
Problem:
Simplify a set of polylines sharing some vertices and edges

Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

• Generalizes the well-known problem of simplifying a single polyline
• Becomes NP-hard and APX-hard
Problem:
Simplify a set of polylines sharing some vertices and edges

**Goal 1:** Minimize the number of vertices
**Goal 2:** Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline
- Becomes NP-hard and APX-hard
- FPT in the number of shared vertices
Summary

Problem:
Simplify a set of polylines sharing some vertices and edges

**Goal 1**: Minimize the number of vertices
**Goal 2**: Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline
- Becomes NP-hard and APX-hard
- FPT in the number of shared vertices
- Not FPT in the number of polylines
Summary

Problem:
Simplify a set of polylines sharing some vertices and edges

Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

• Generalizes the well-known problem of simplifying a single polyline
• Becomes NP-hard and APX-hard
• FPT in the number of shared vertices
• Not FPT in the number of polylines

• Since there is no PTAS, is there a constant factor approximation algorithm?