Julius-Maximilians-
UNIVERSITÄT WÜRZBURG

Lehrstuhl für
INFORMATIK I
Algorithmen & Komplexität

Institut für Informatik

Simplification of Polyline Bundles

Joachim Spoerhase
Aalto University, Finland

Sabine Storandt
Universität Konstanz, Germany

Johannes Zink
Universität Würzburg, Germany

Agenda

Motivation

Motivation

- Maps often consist of polylines

Motivation

- Maps often consist of polylines

Motivation

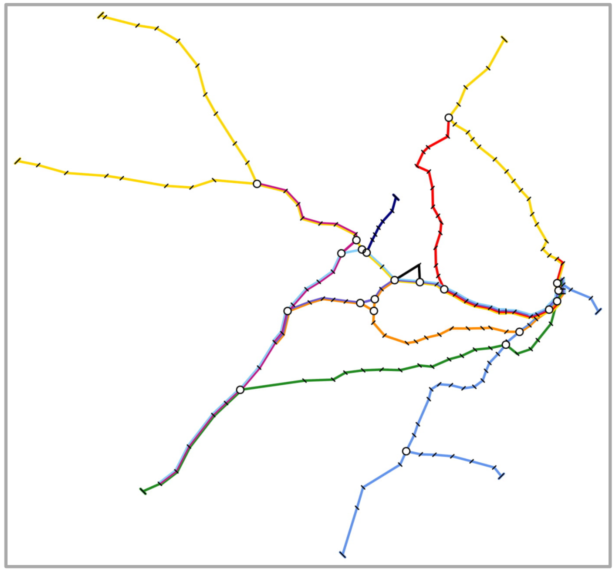
- Maps often consist of polylines

- Multiple polylines share bends and segments sectionwise

Motivation

- Maps often consist of polylines

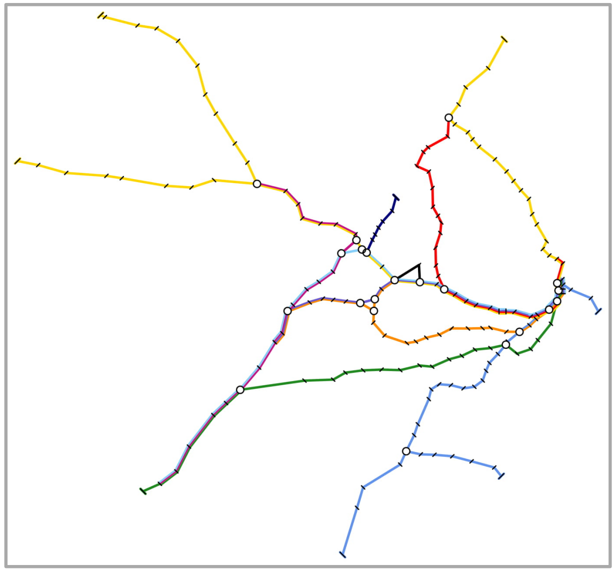
- Multiple polylines share bends and segments sectionwise



Motivation

- Maps often consist of polylines

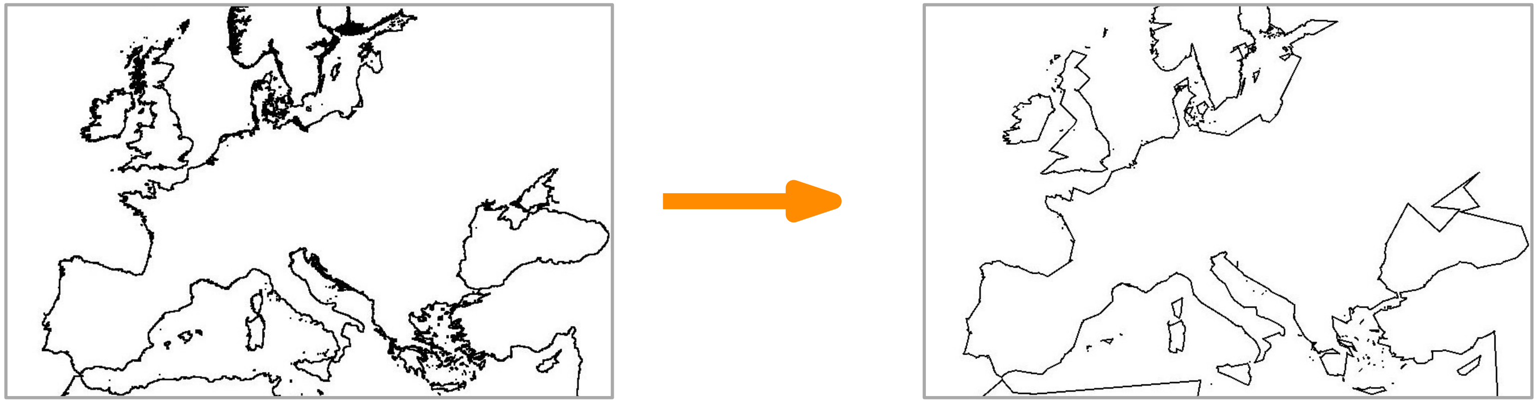
- Multiple polylines share bends and segments sectionwise



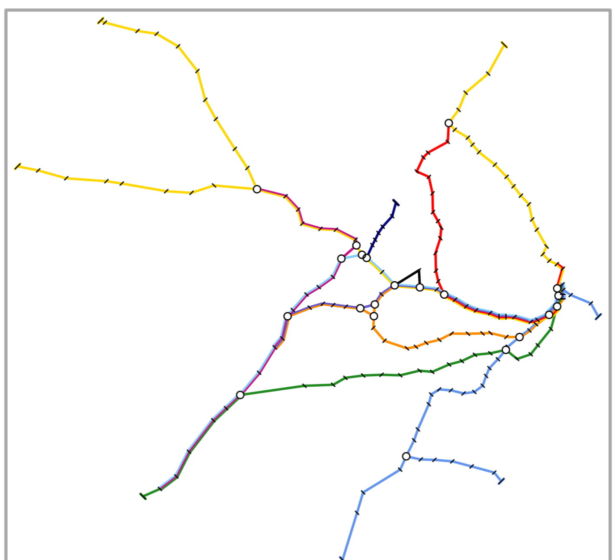
- Reduce full data for zooming or schematization

Motivation

- Maps often consist of polylines



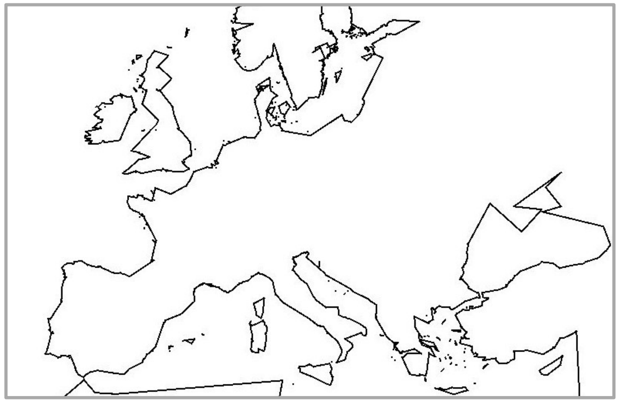
- Multiple polylines share bends and segments sectionwise



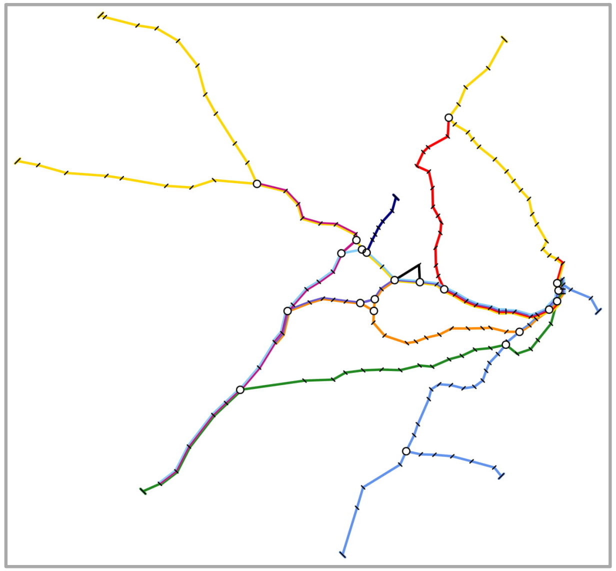
- Reduce full data for zooming or schematization

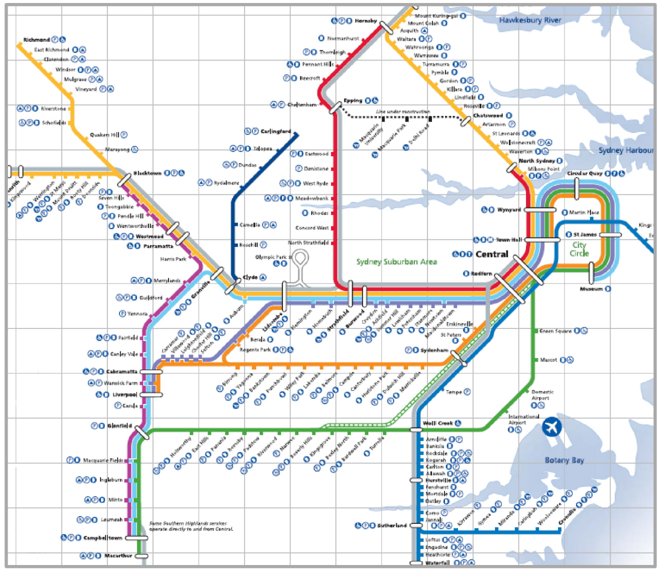
Motivation

- Maps often consist of polylines



- Multiple polylines share bends and segments sectionwise





- Reduce full data for zooming or schematization

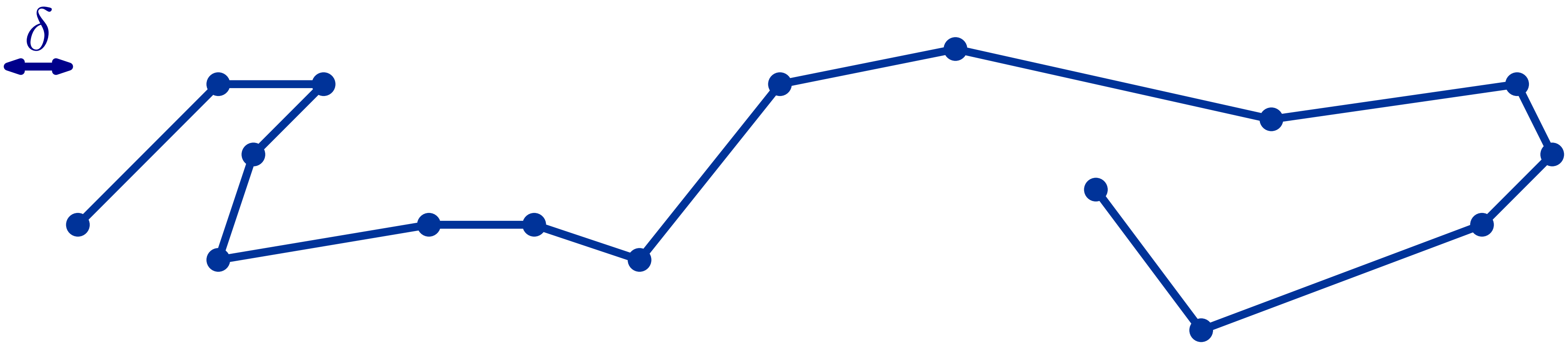
Introduction: Simplifying a Polyline

Introduction: Simplifying a Polyline

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

Introduction: Simplifying a Polyline

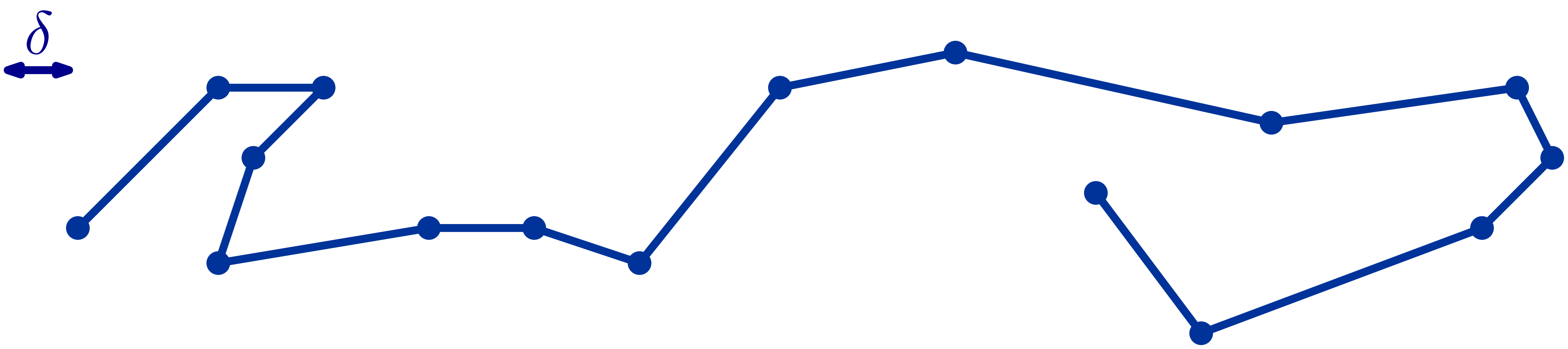
Given:
- polyline L as a sequence of points (bends) in the plane
- distance threshold δ



Introduction: Simplifying a Polyline

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

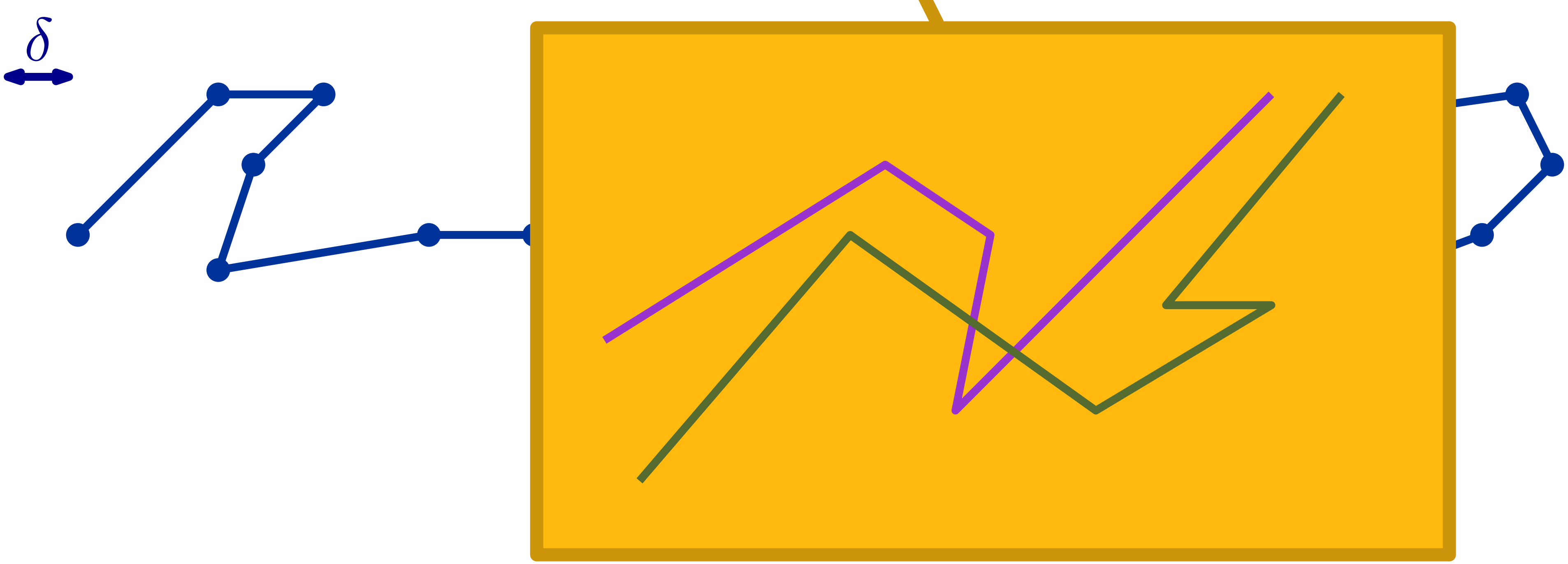
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

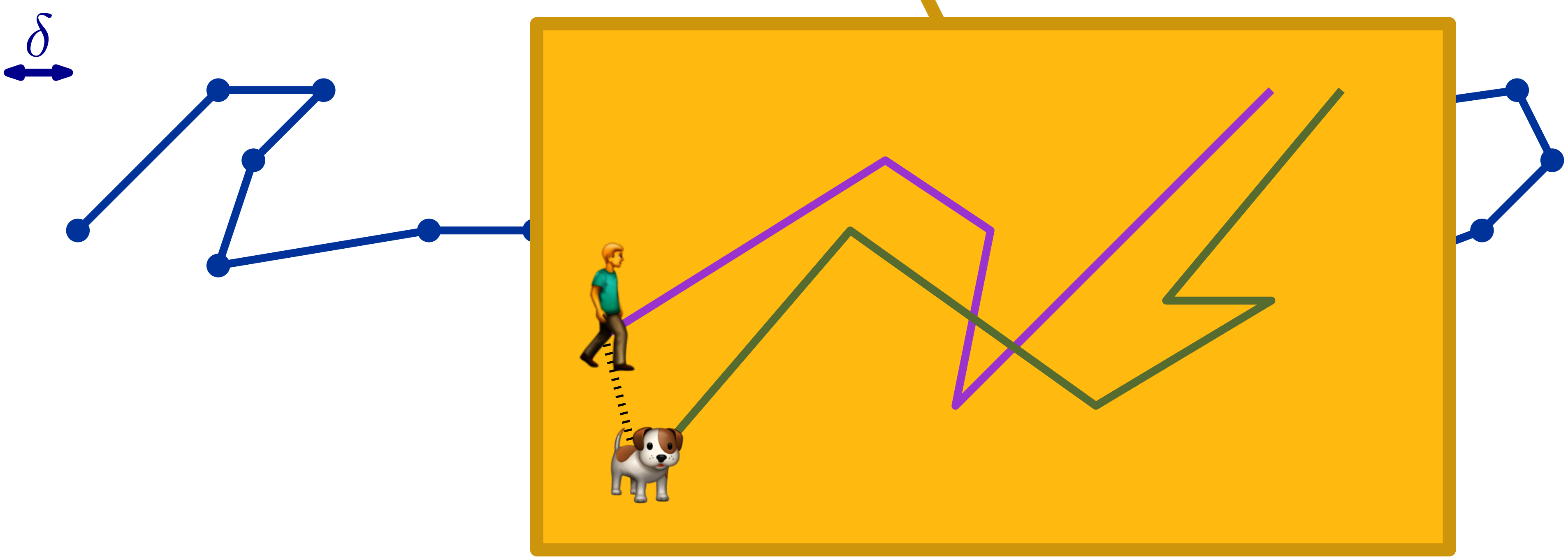
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

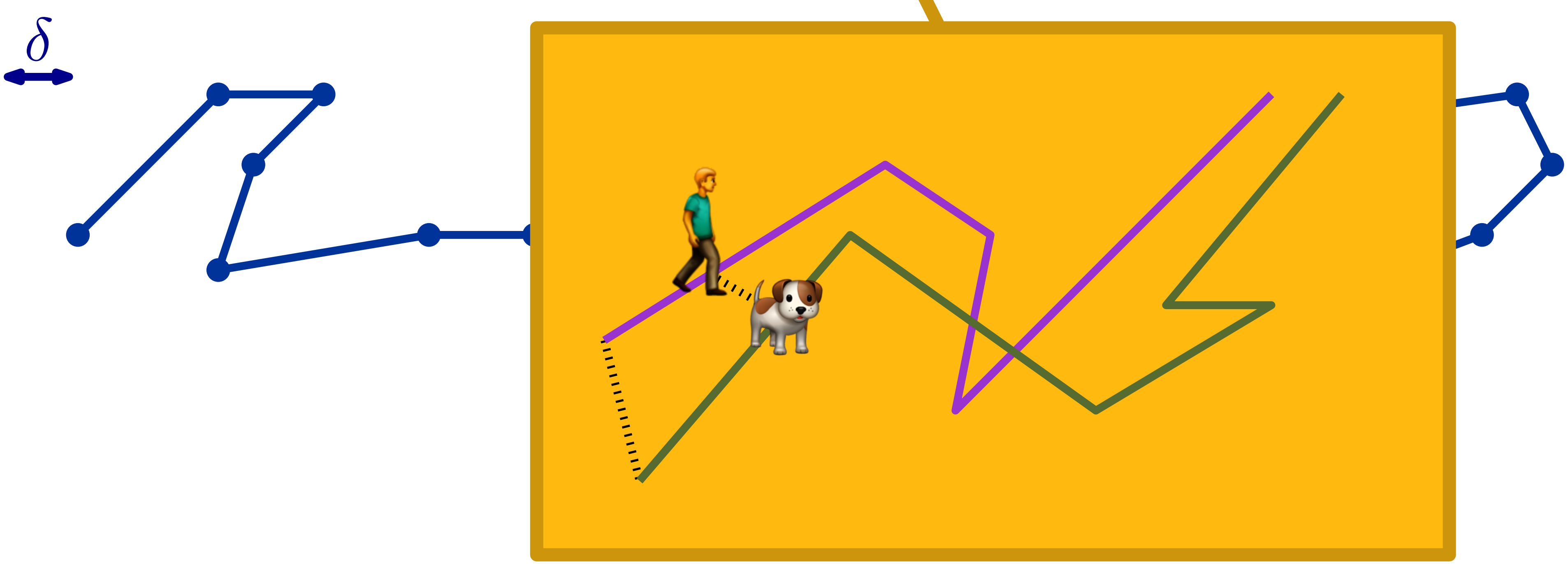
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

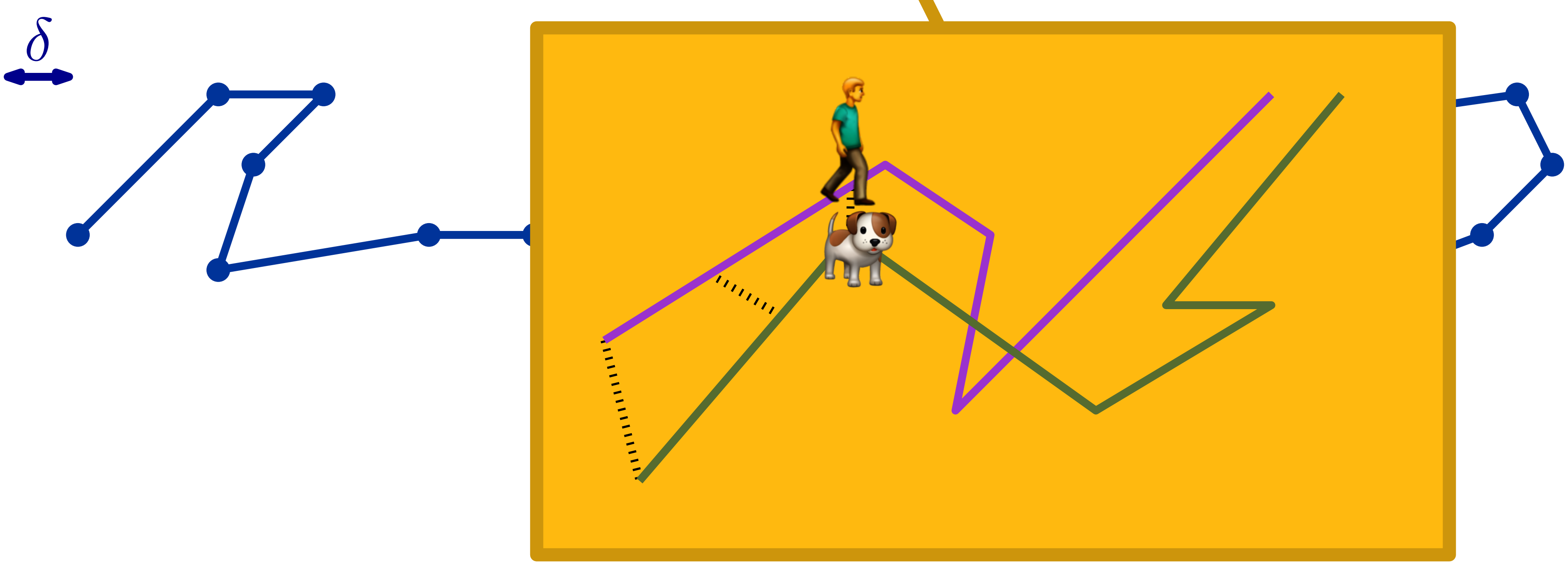
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

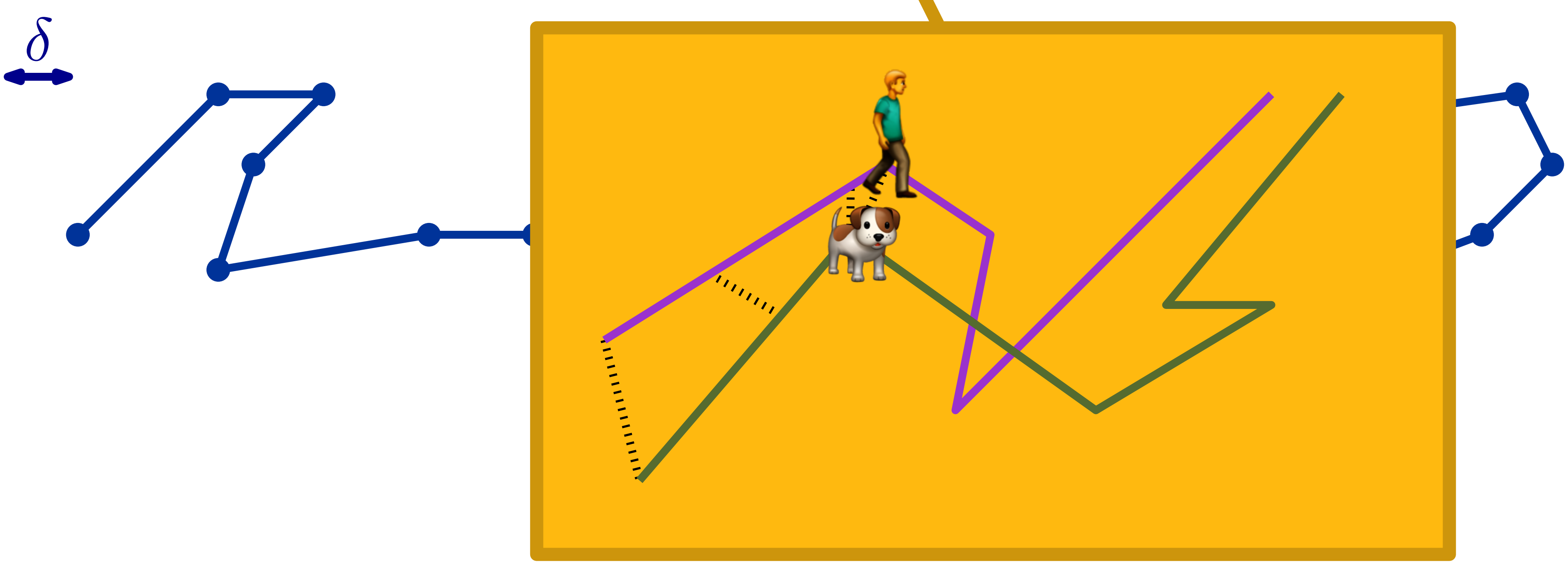
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

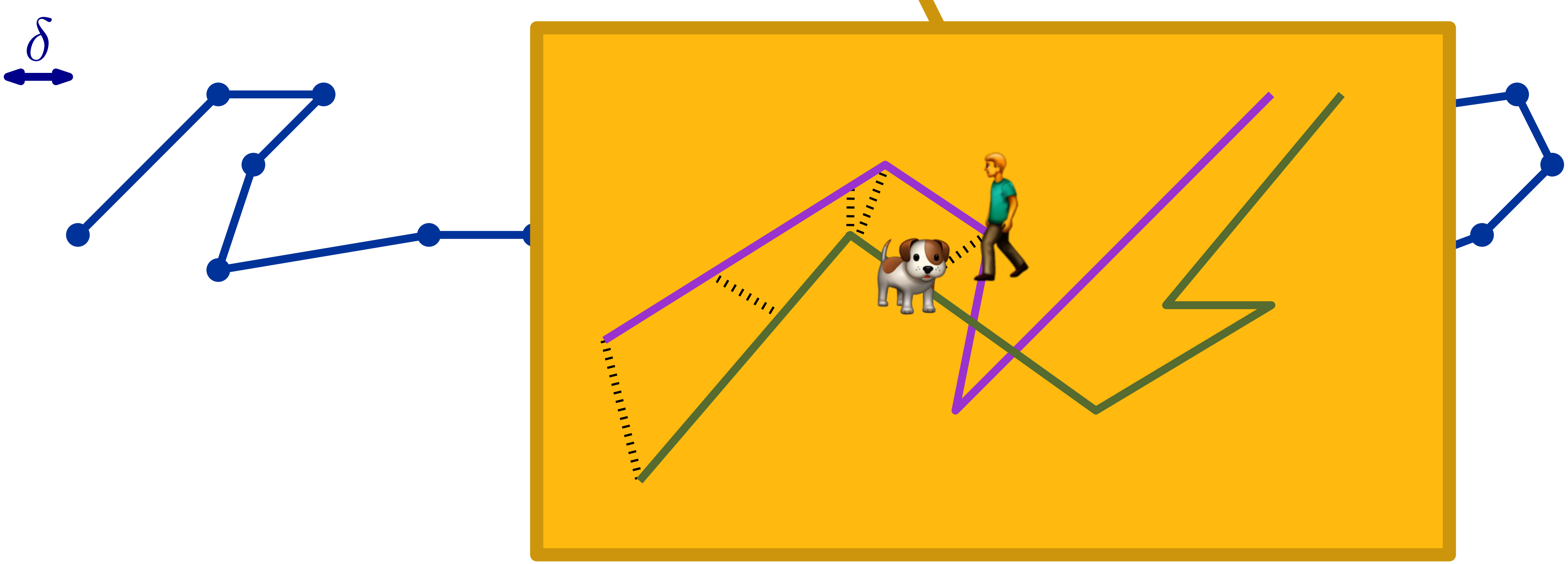
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

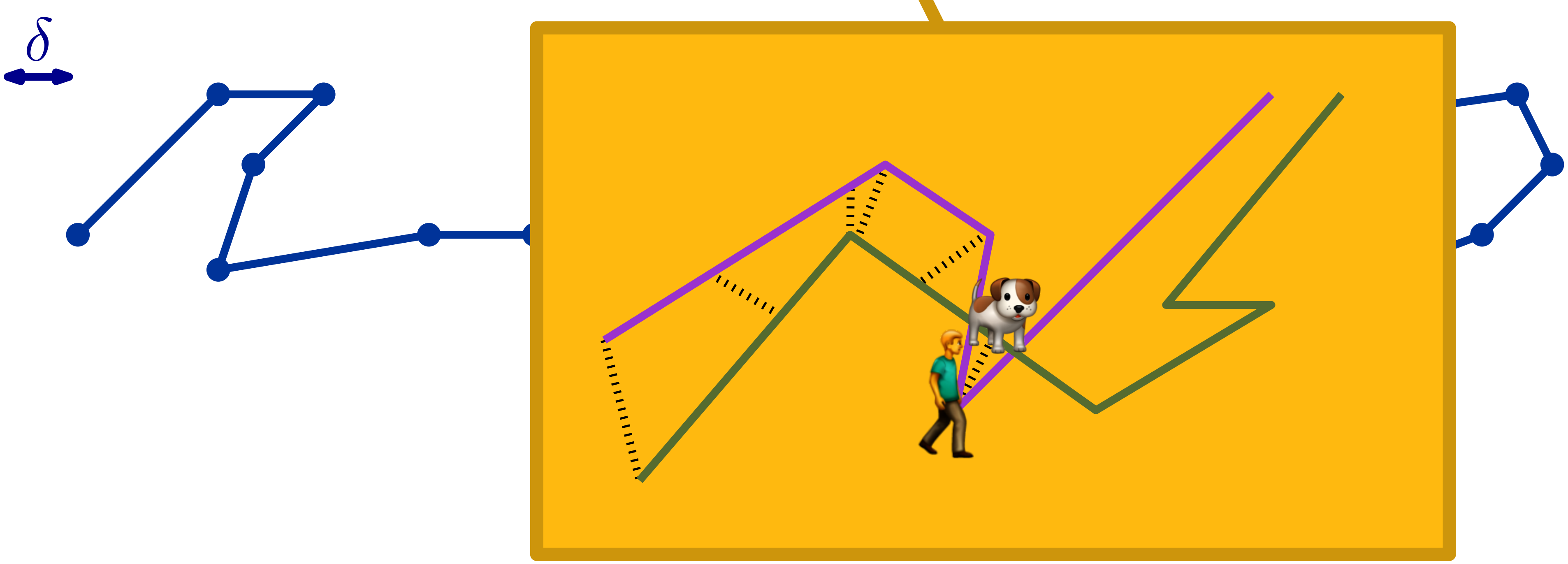
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

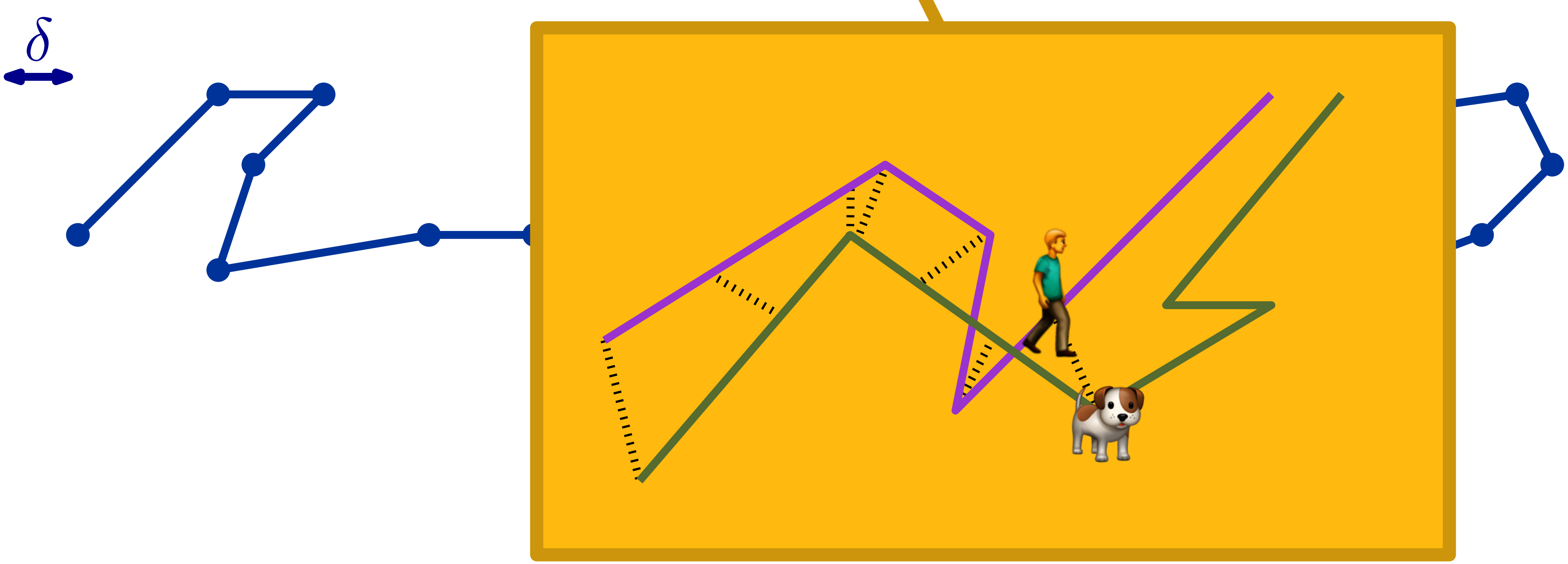
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

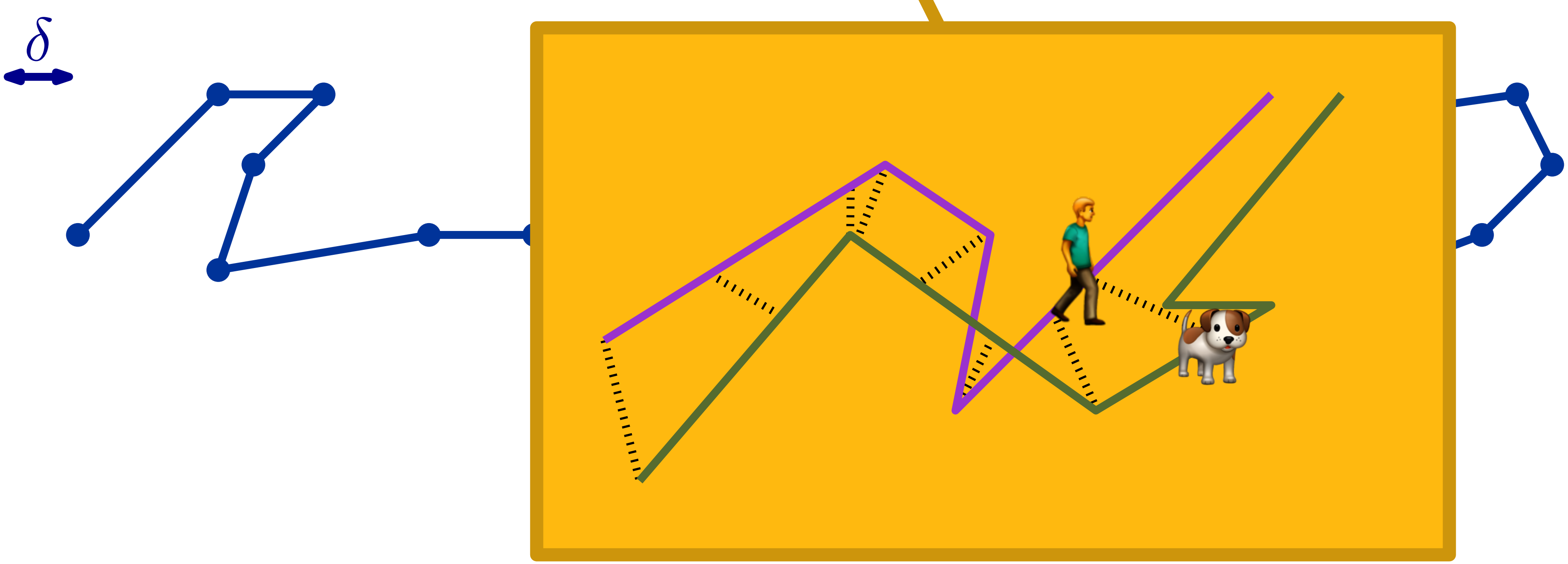
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

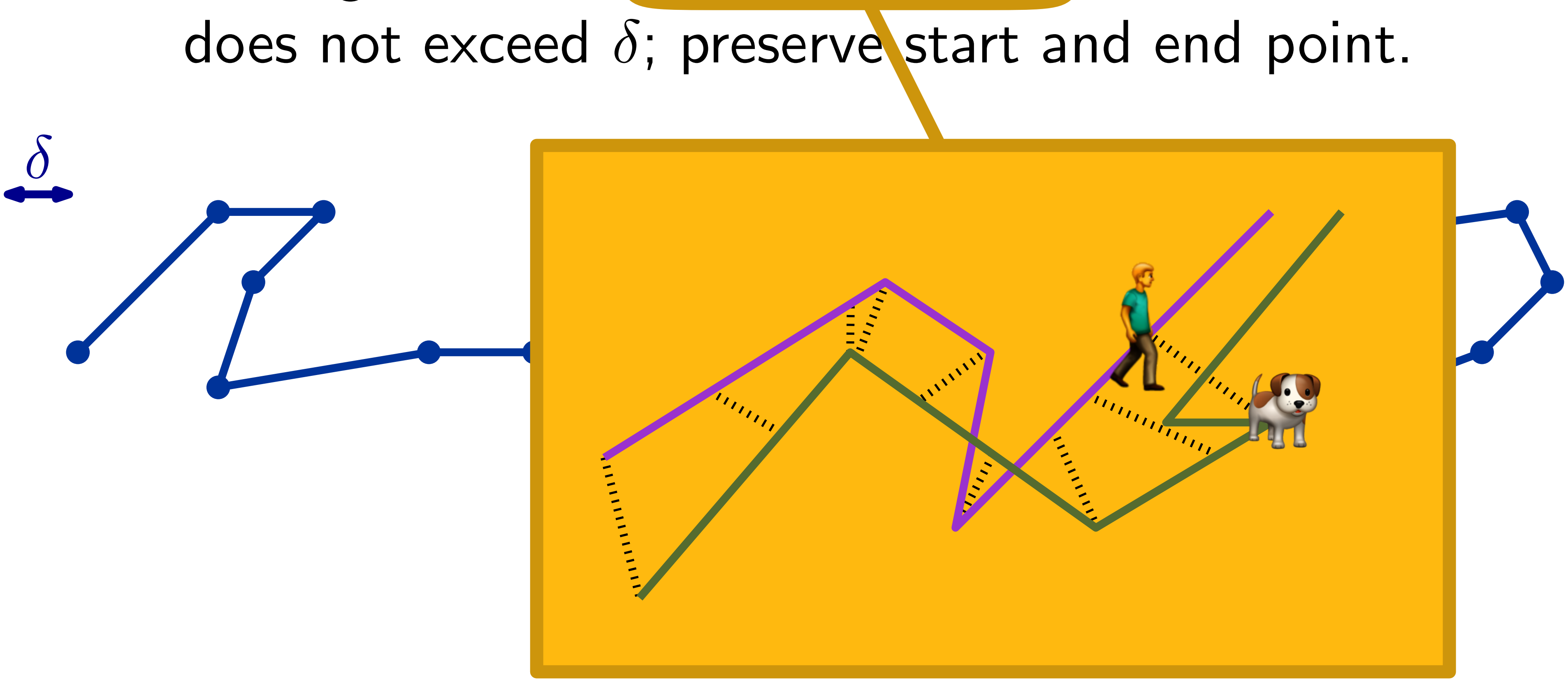
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

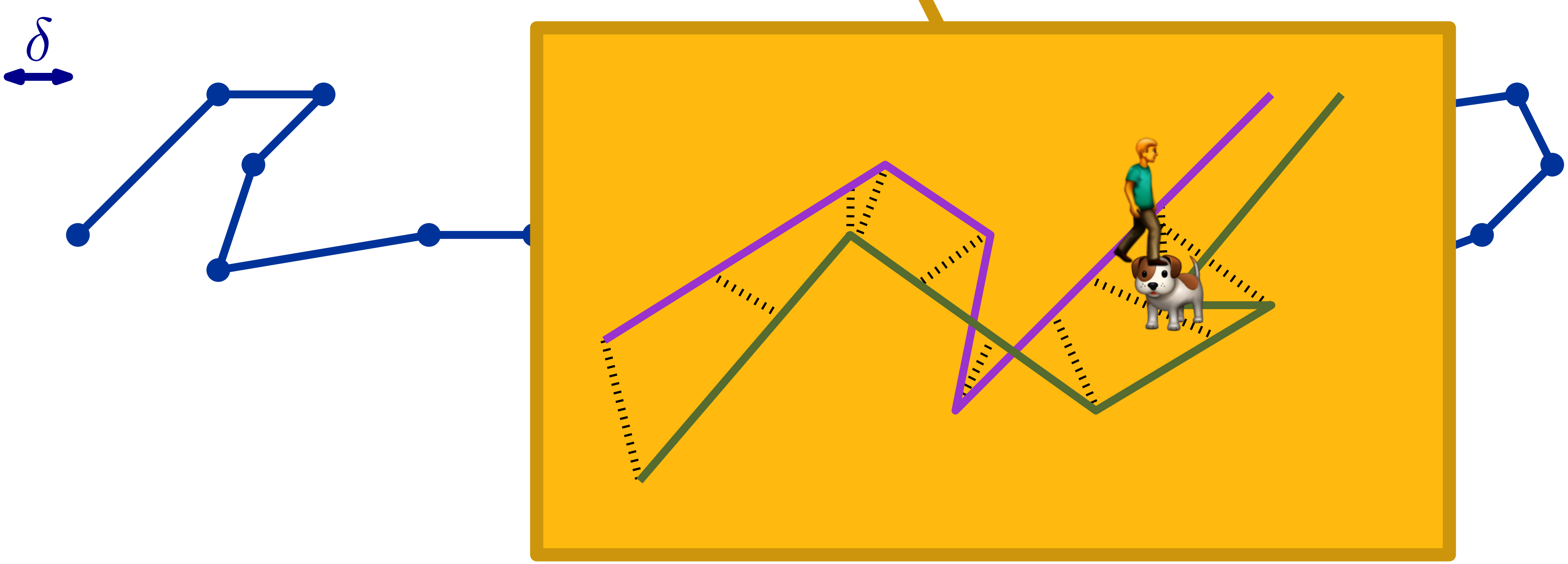
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

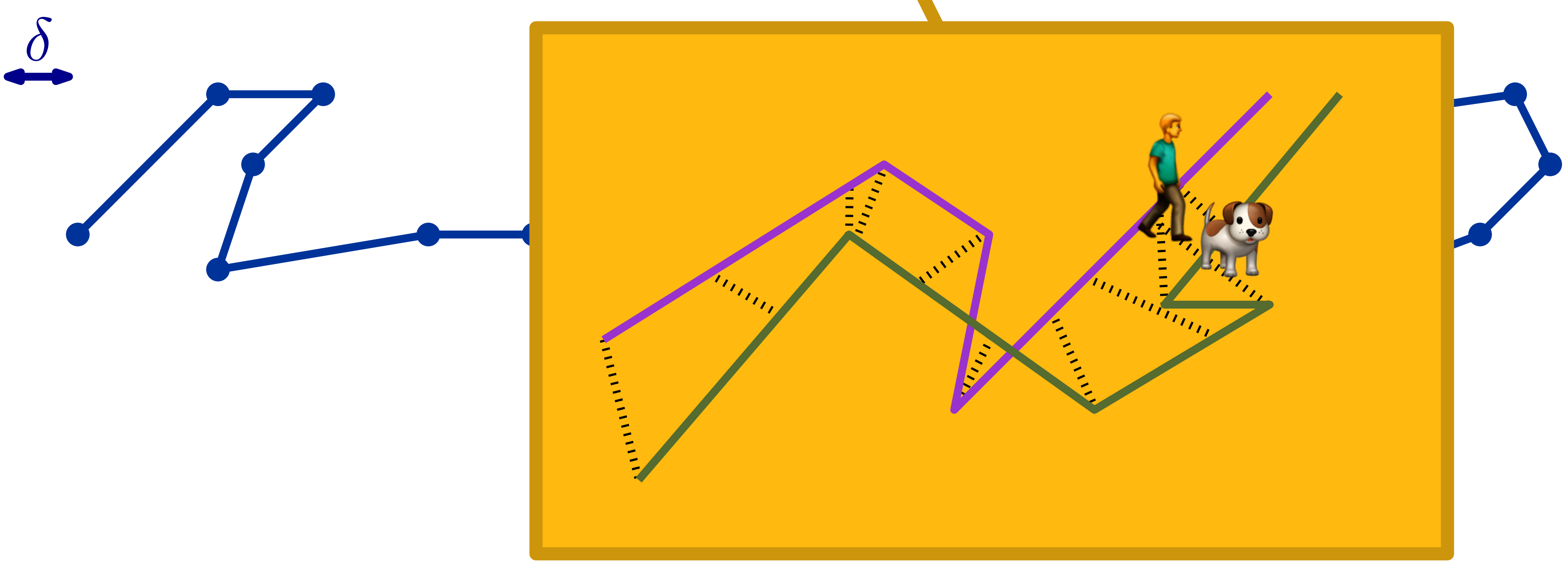
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

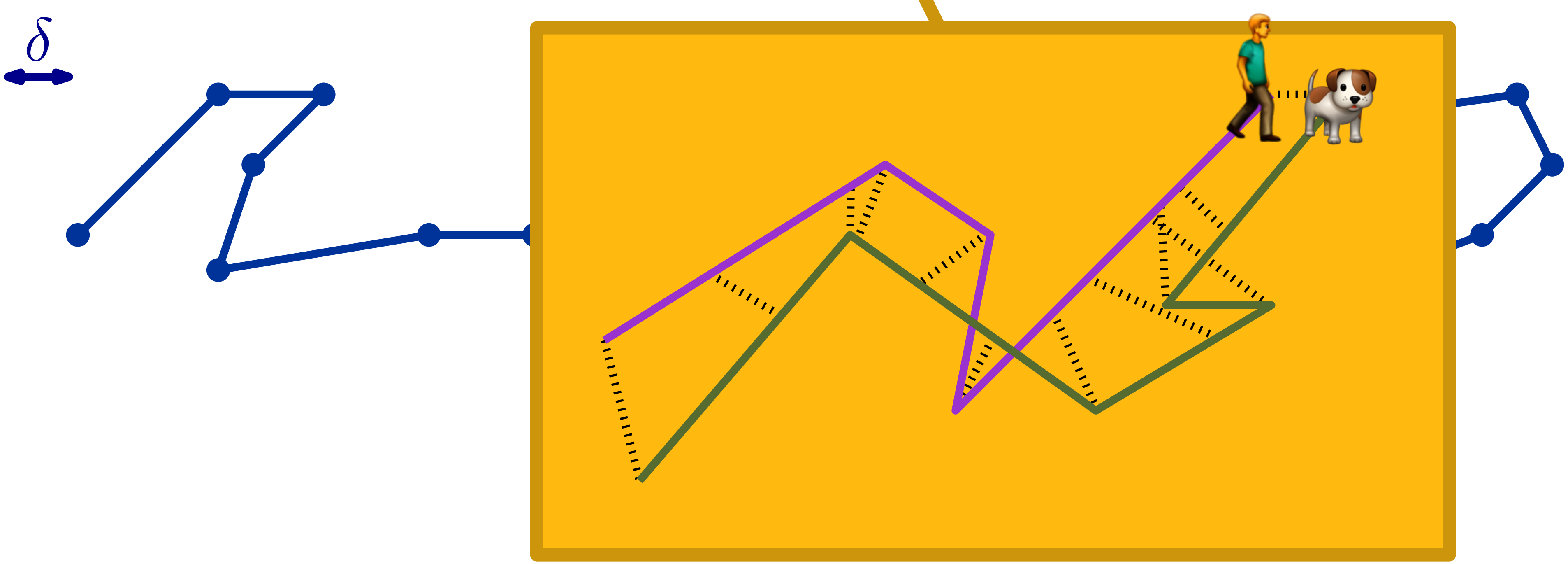
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

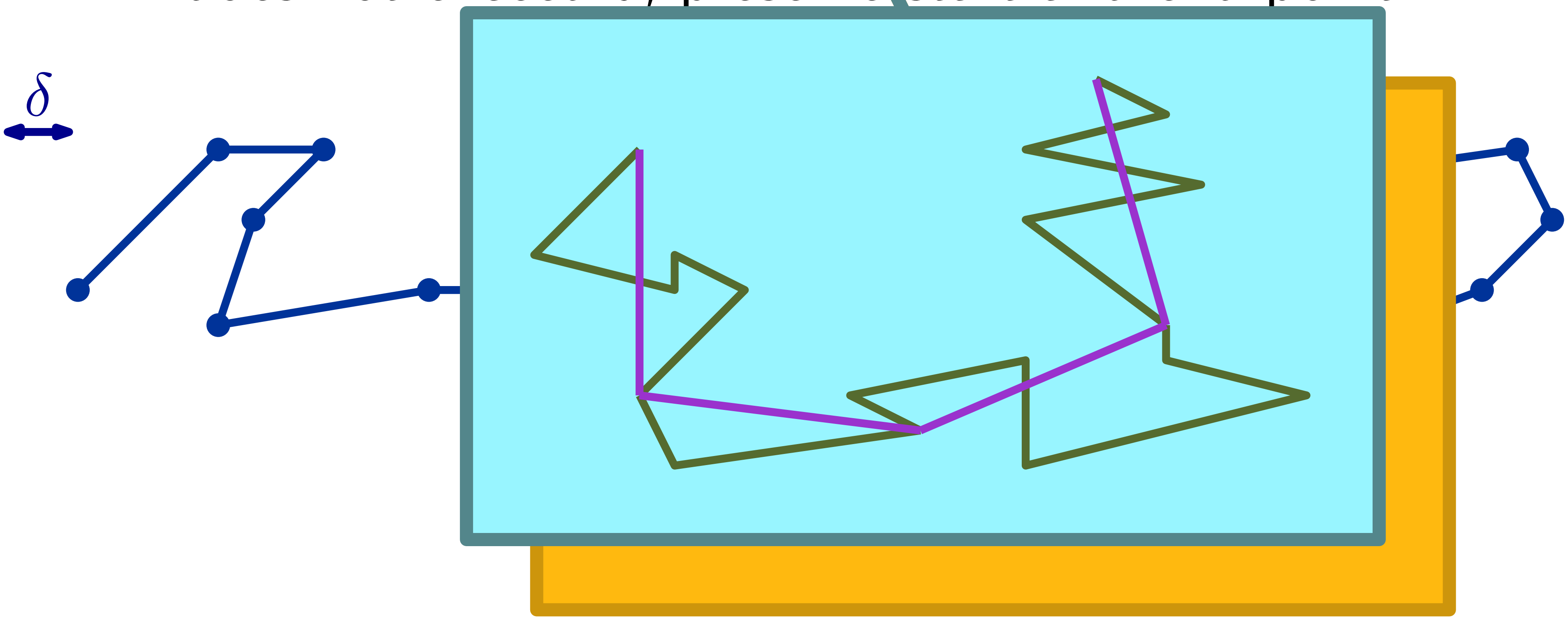
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

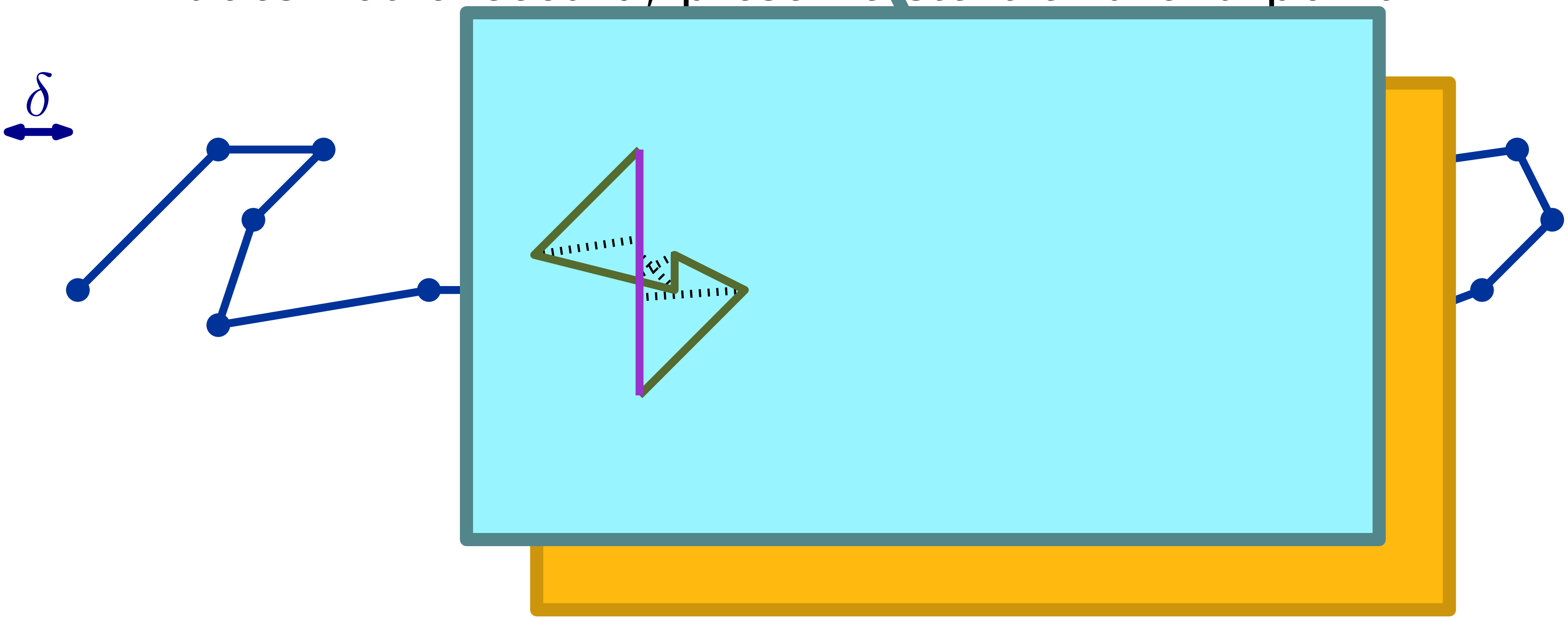
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

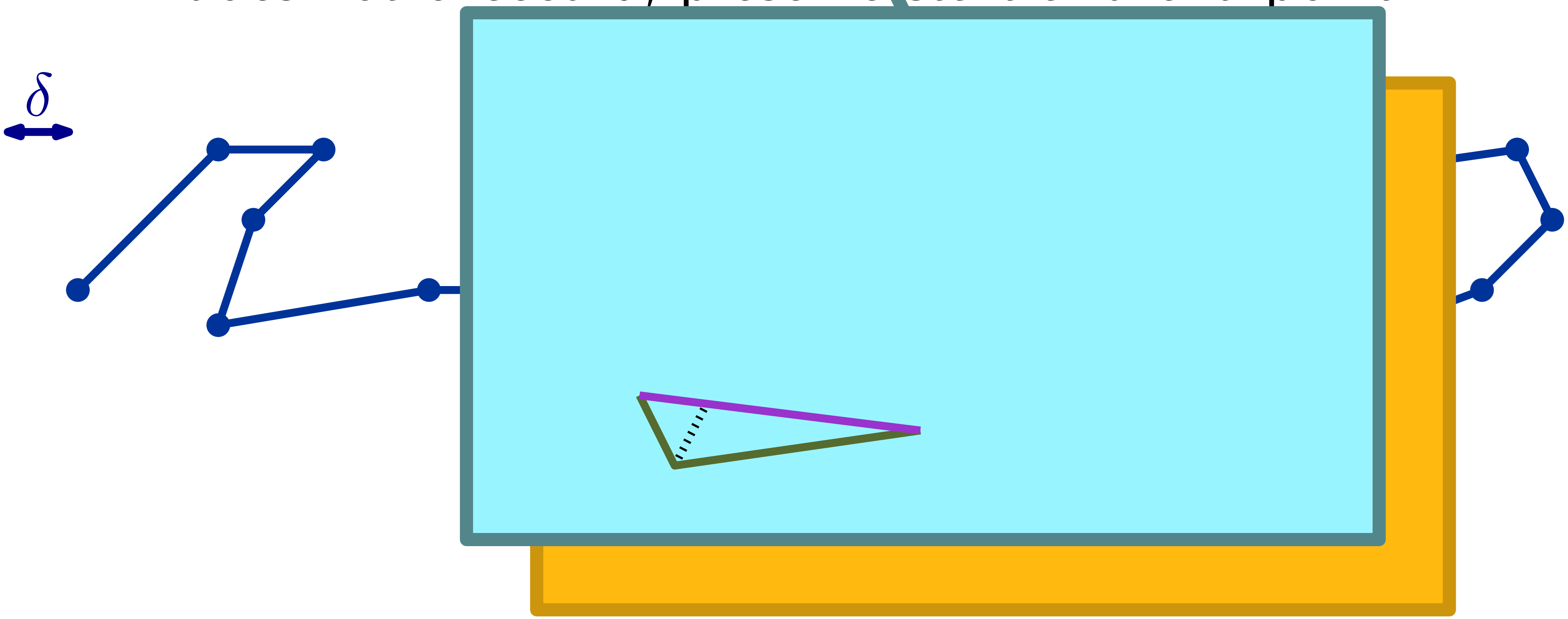
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

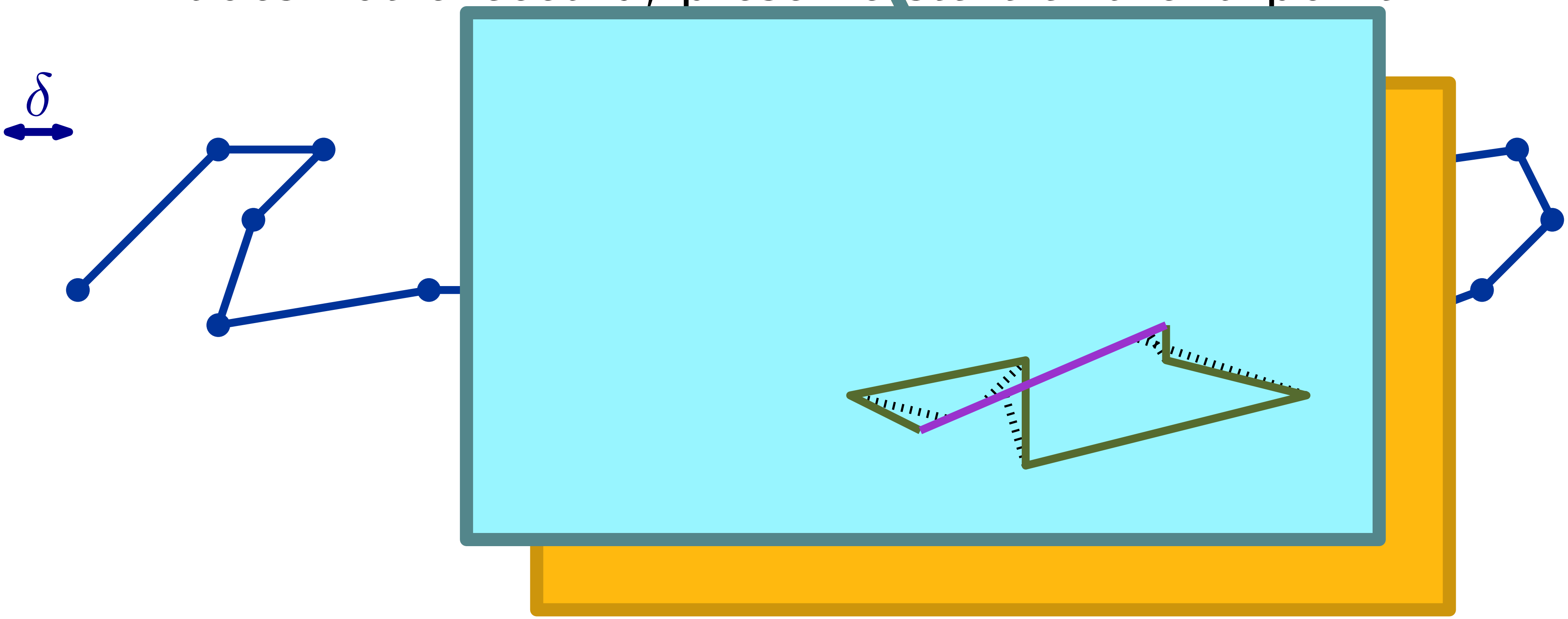
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

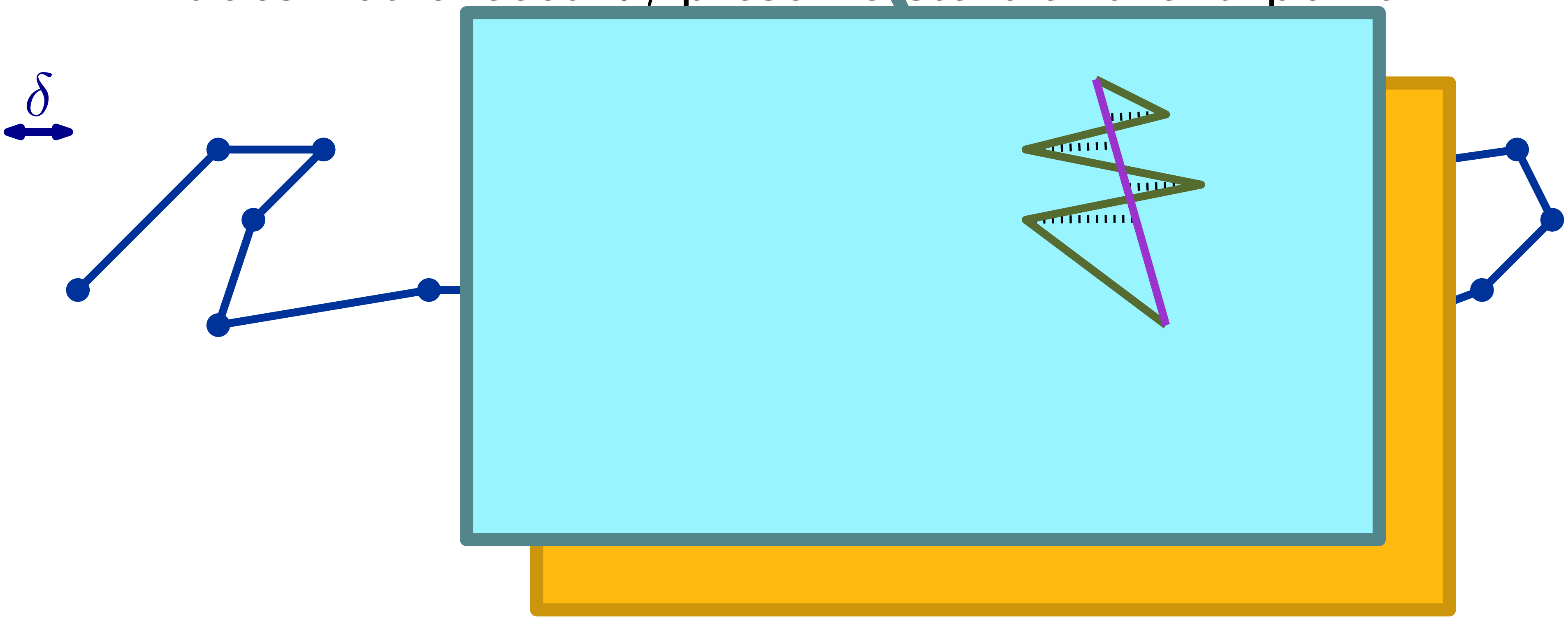
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

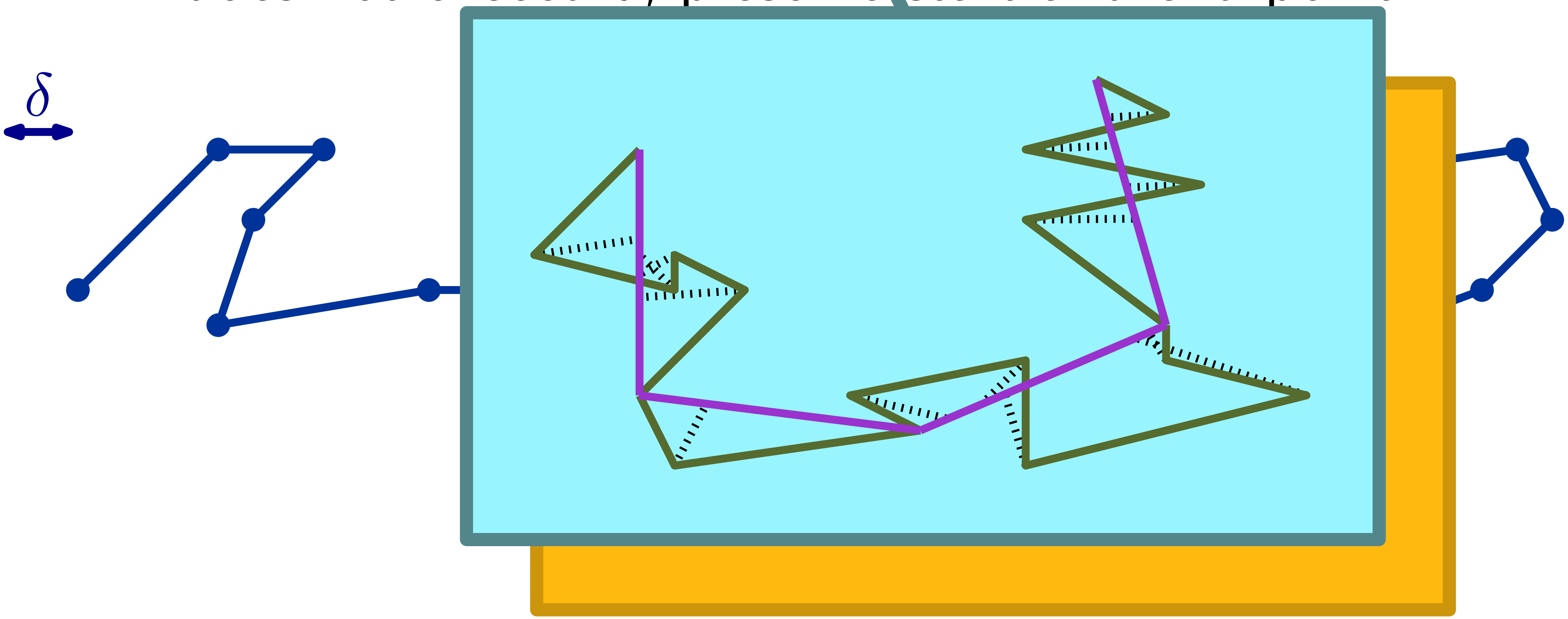
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

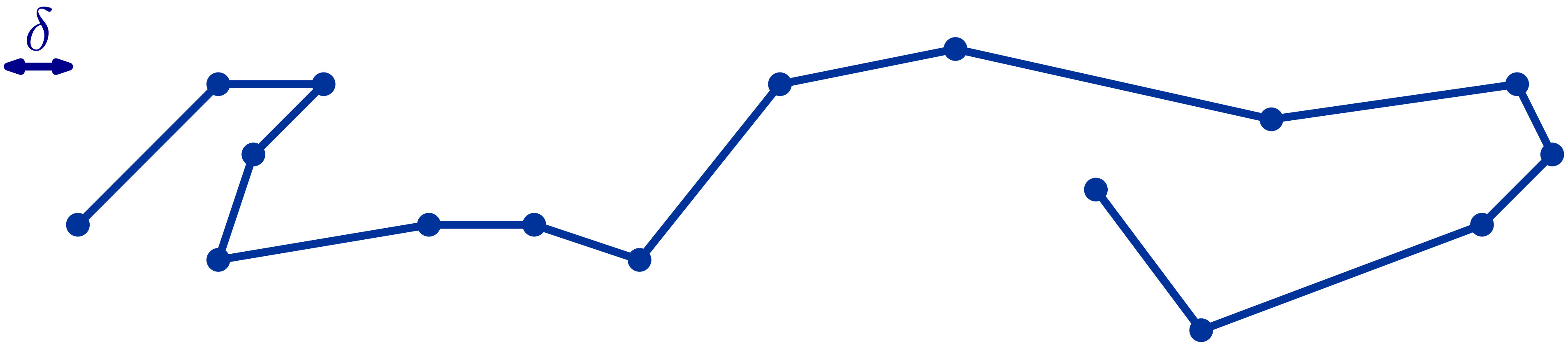
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

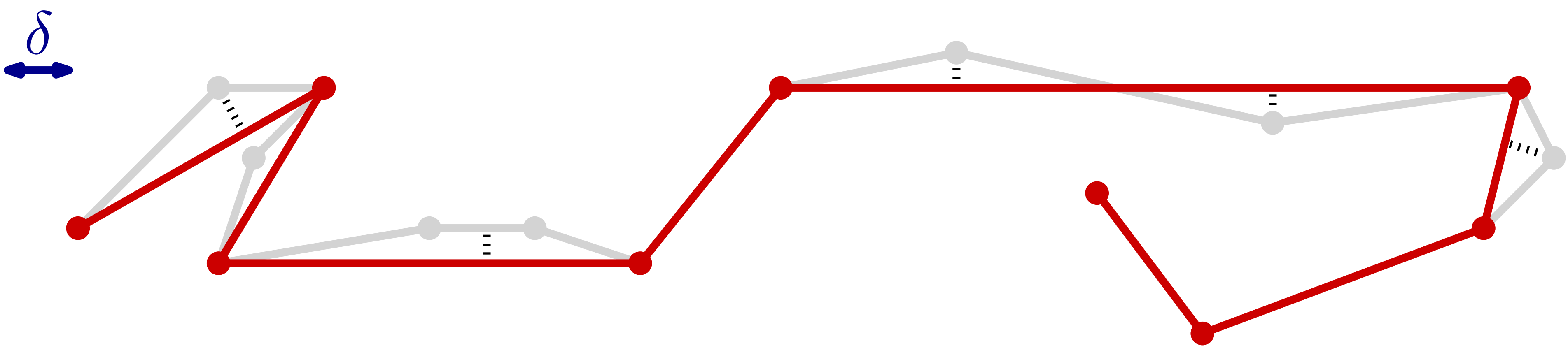
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

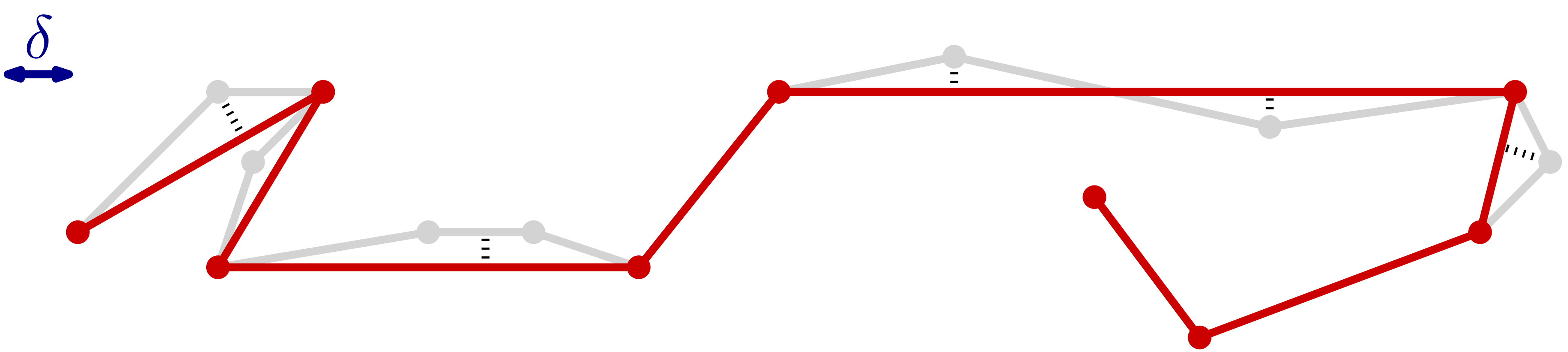
Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



Introduction: Simplifying a Polyline

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

Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.

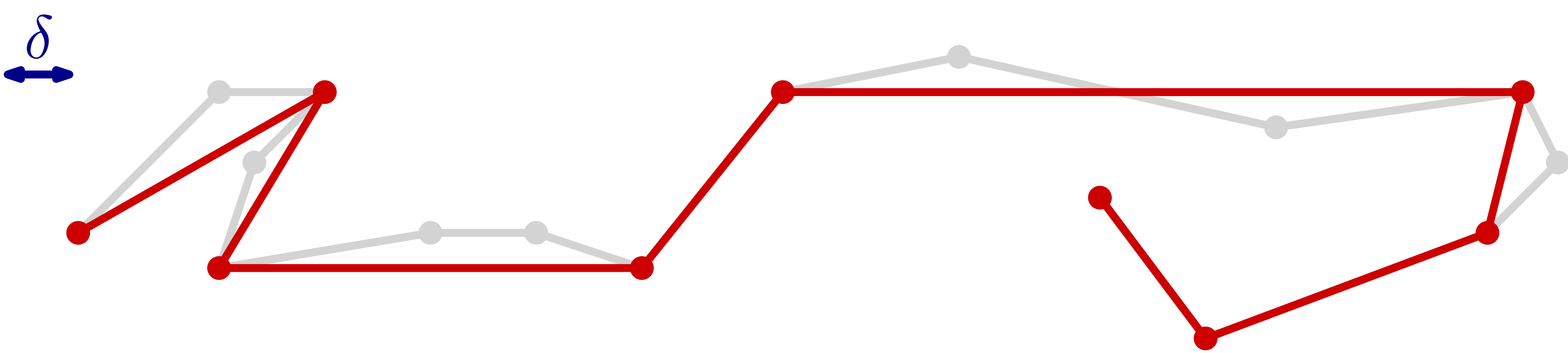


- Can be solved efficiently in $O(|L|^3)$ time.
  [Imai, Iri '88], [Godau '91]

Introduction: Simplifying a Polyline

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

Goal: Find a minimum size **subsequence** L' of L, such that the segment-wise Fréchet distance between L' and L does not exceed δ; preserve start and end point.



- Can be solved efficiently in $O(|L|^3)$ time.
  [Imai, Iri '88], [Godau '91]

Why Not Simplifying Independently?

Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

Why Not Simplifying Independently?

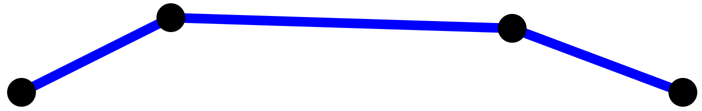
A bend or segment might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality

Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

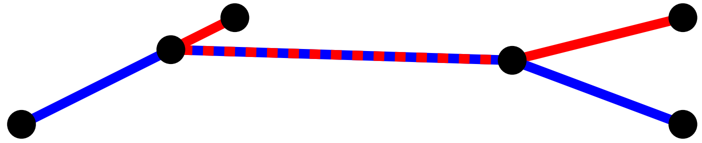
$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

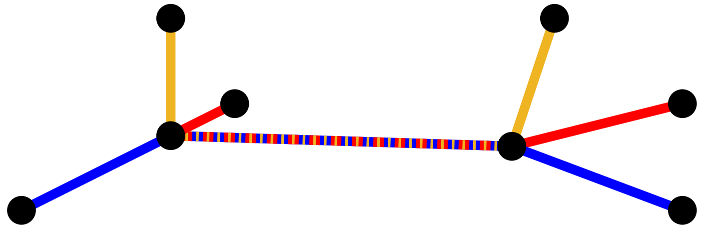
$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

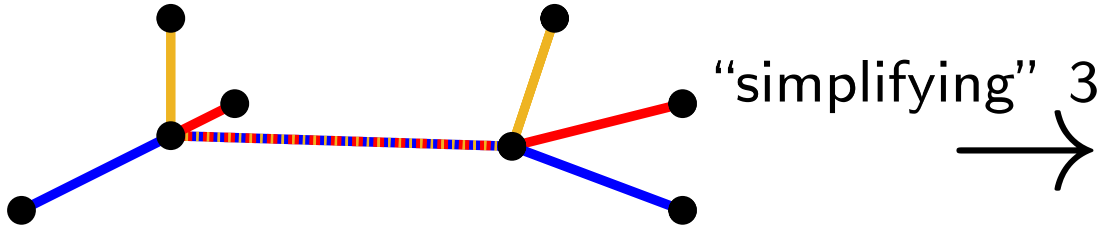
$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

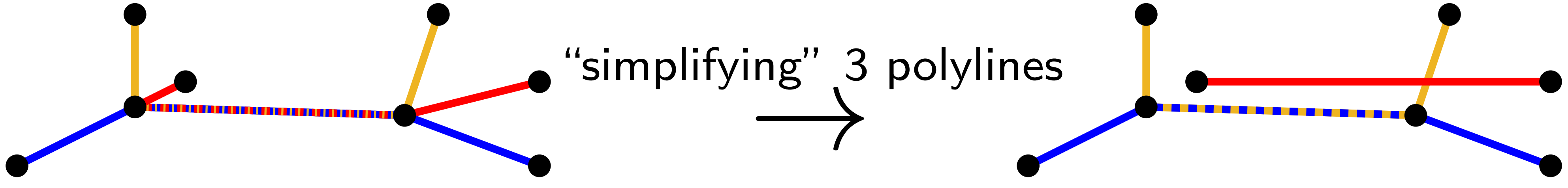
⇒ misleading picture of the reality



Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

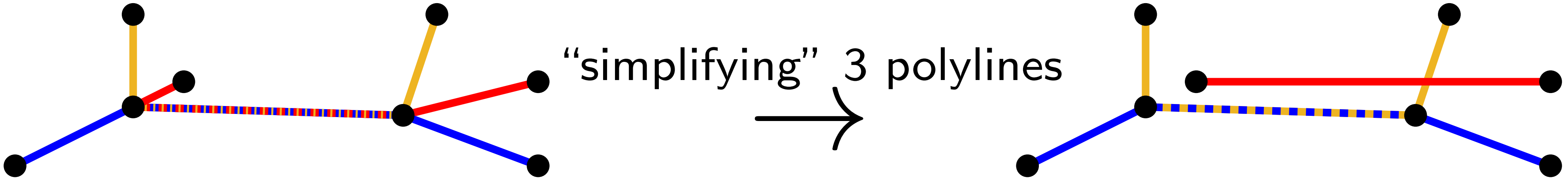
$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality

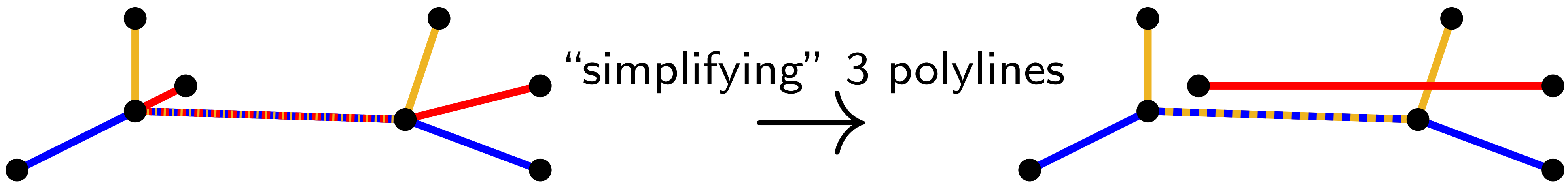


⇒ the total complexity might even increase

Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality

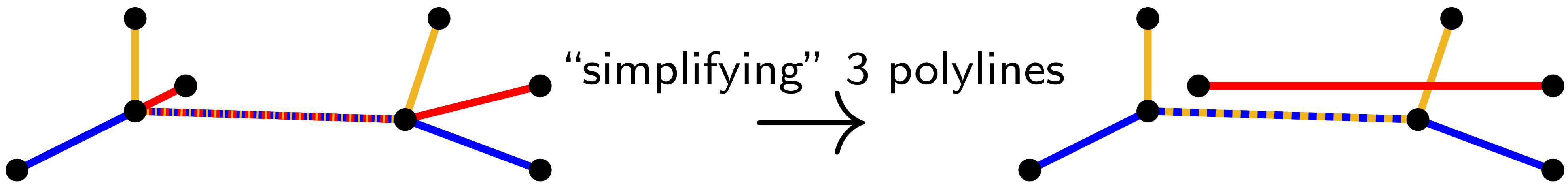


⇒ the total complexity might even increase

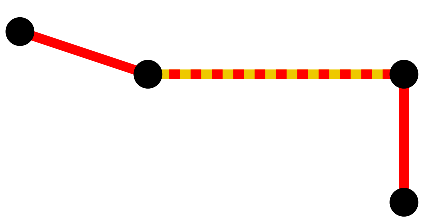
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



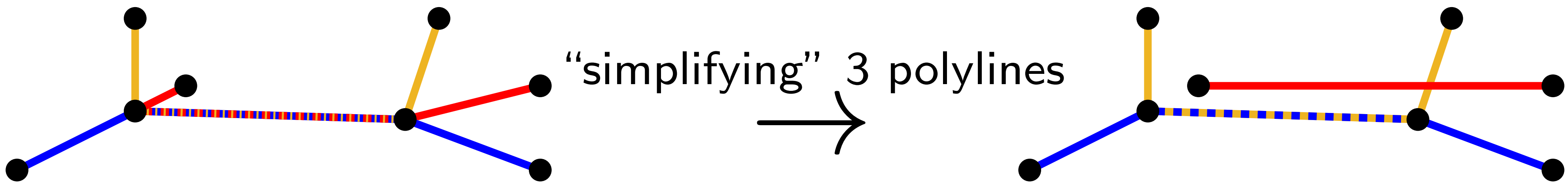
$\Rightarrow$ the total complexity might even increase



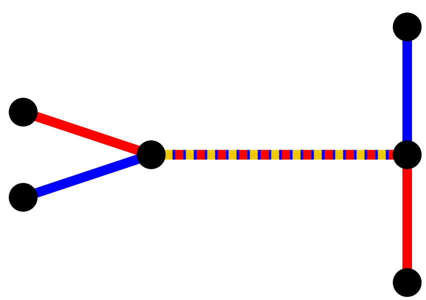
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



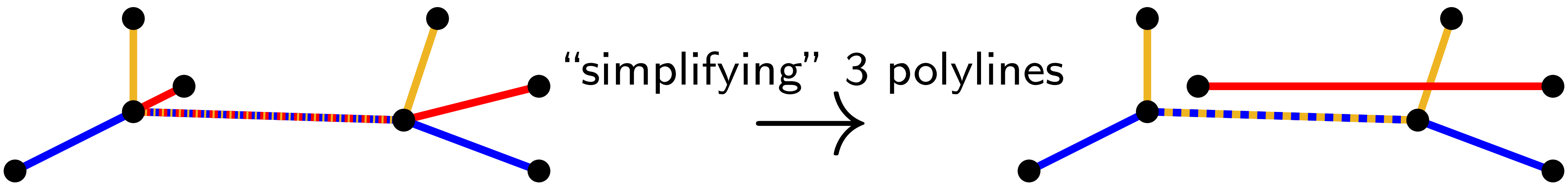
⇒ the total complexity might even increase



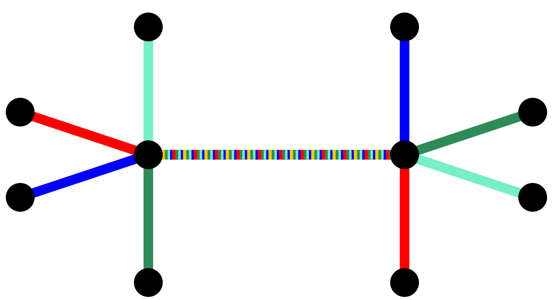
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



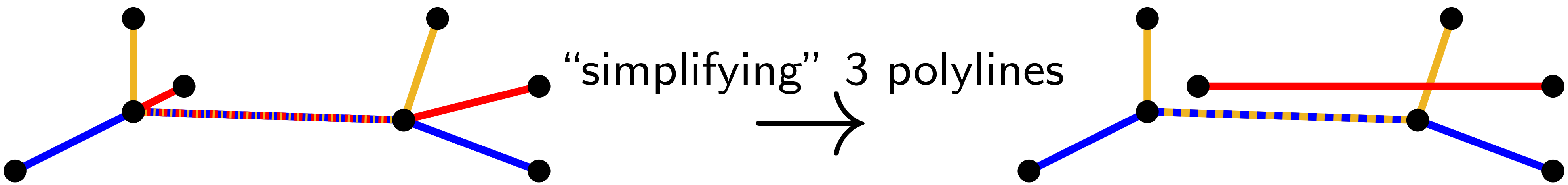
⇒ the total complexity might even increase



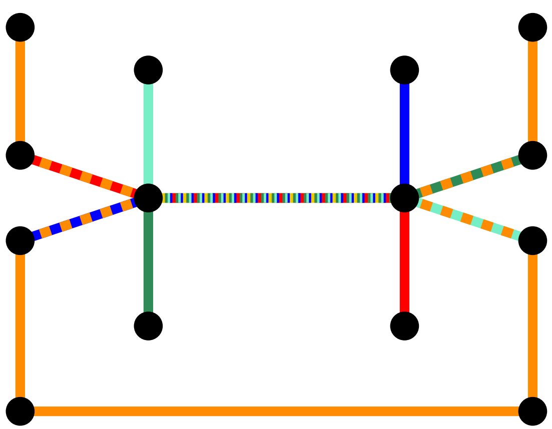
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



⇒ the total complexity might even increase

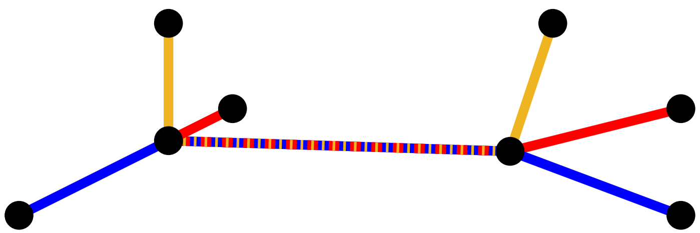


14 bends, 14 segments

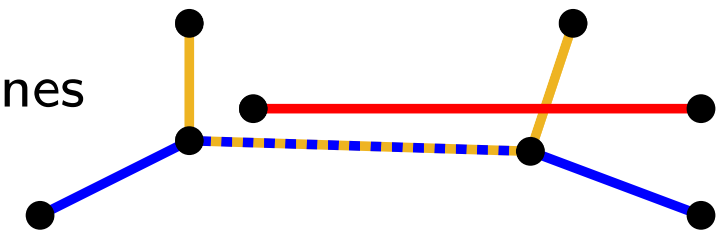
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

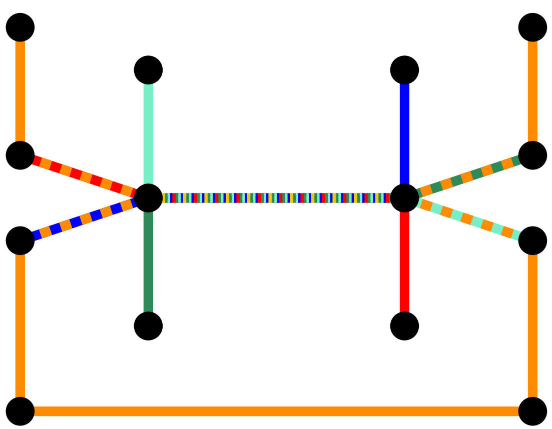
⇒ misleading picture of the reality



“simplifying” 3 polylines ⟶



⇒ the total complexity might even increase



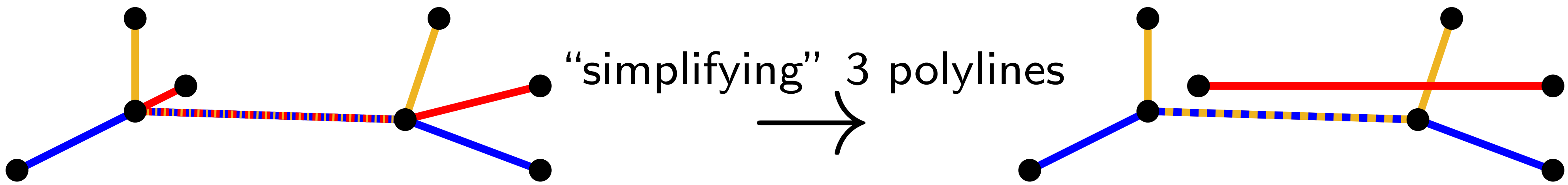
“simplifying” 6 polylines ⟶

14 bends, 14 segments

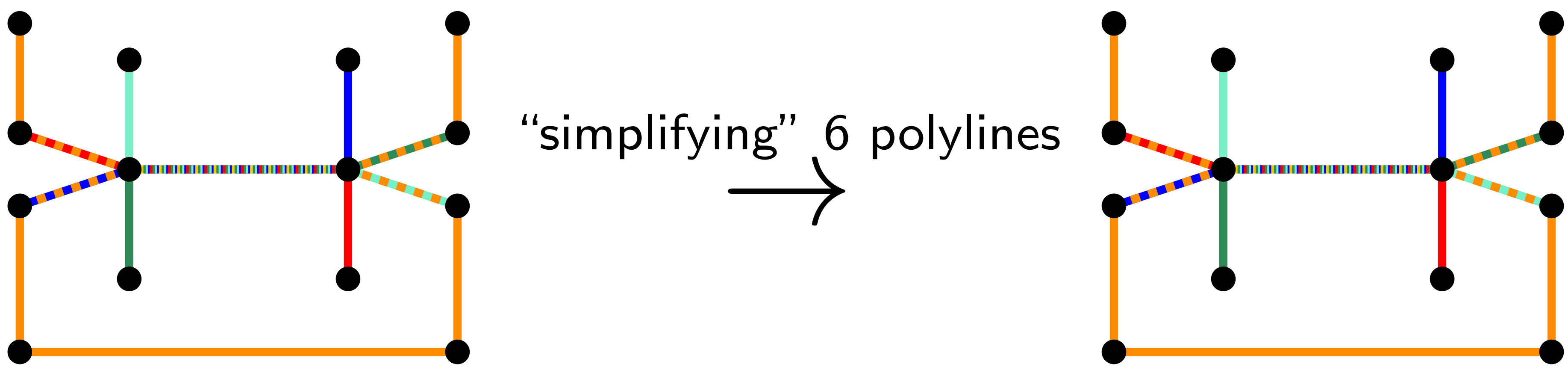
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



⇒ the total complexity might even increase

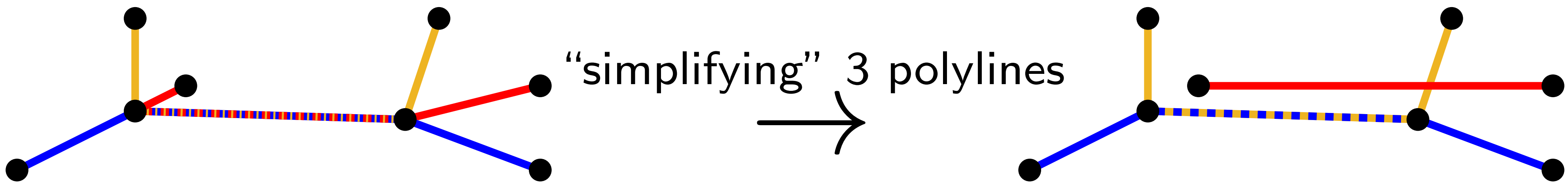


14 bends, 14 segments

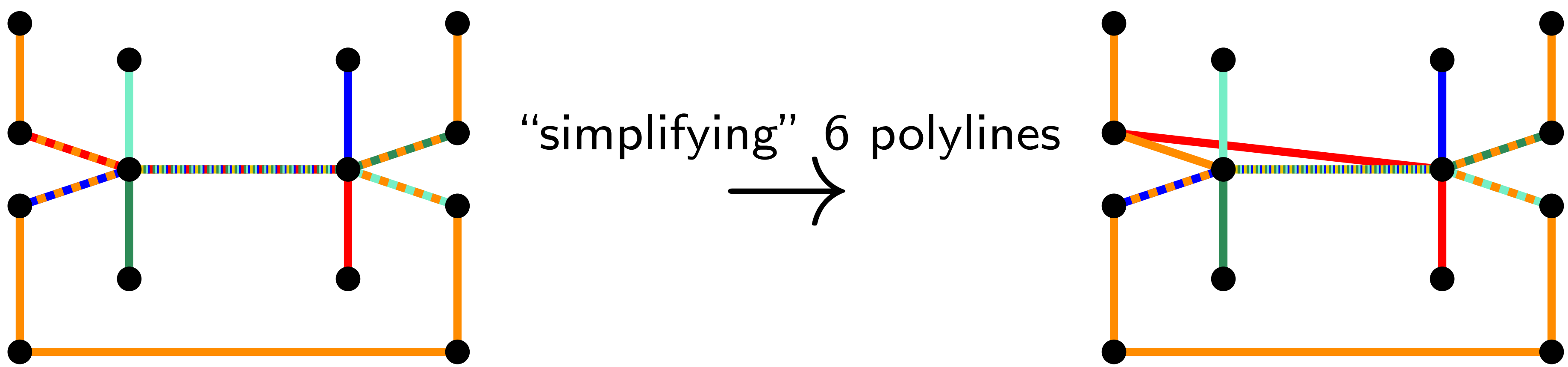
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



⇒ the total complexity might even increase

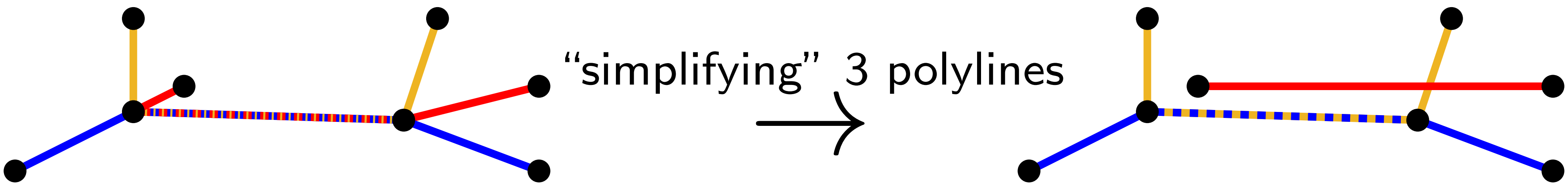


14 bends, 14 segments

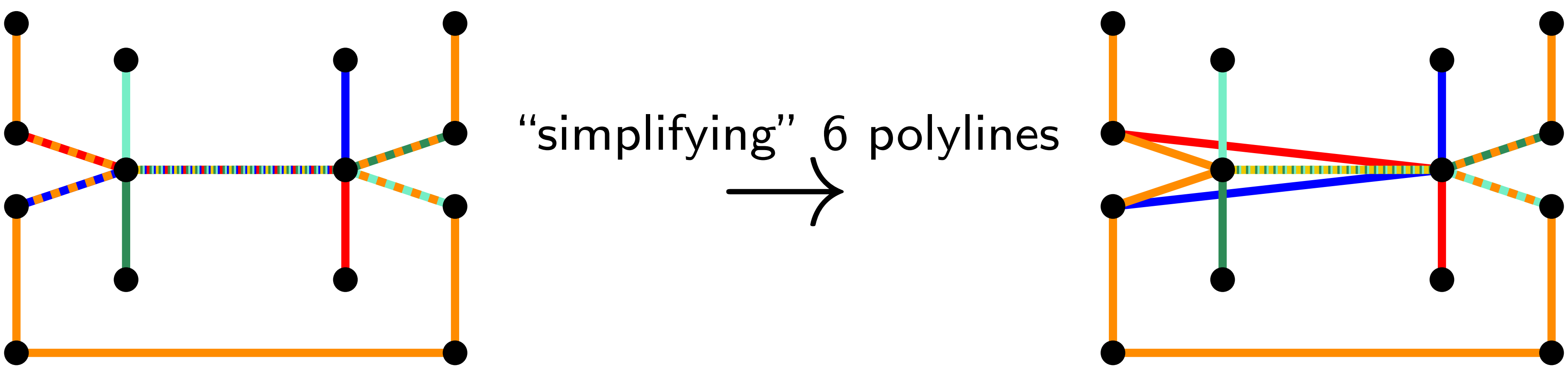
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



⇒ the total complexity might even increase

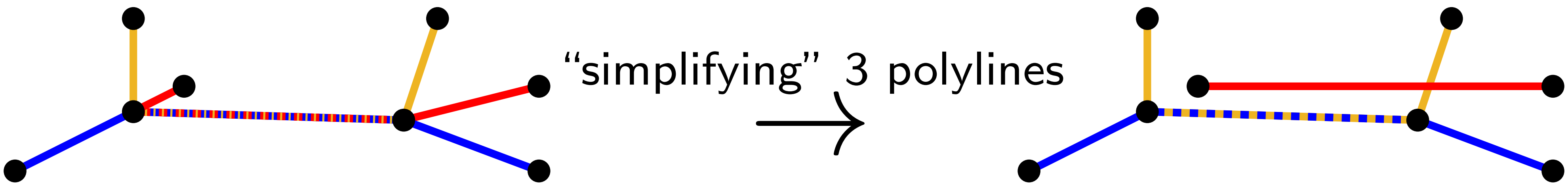


14 bends, 14 segments

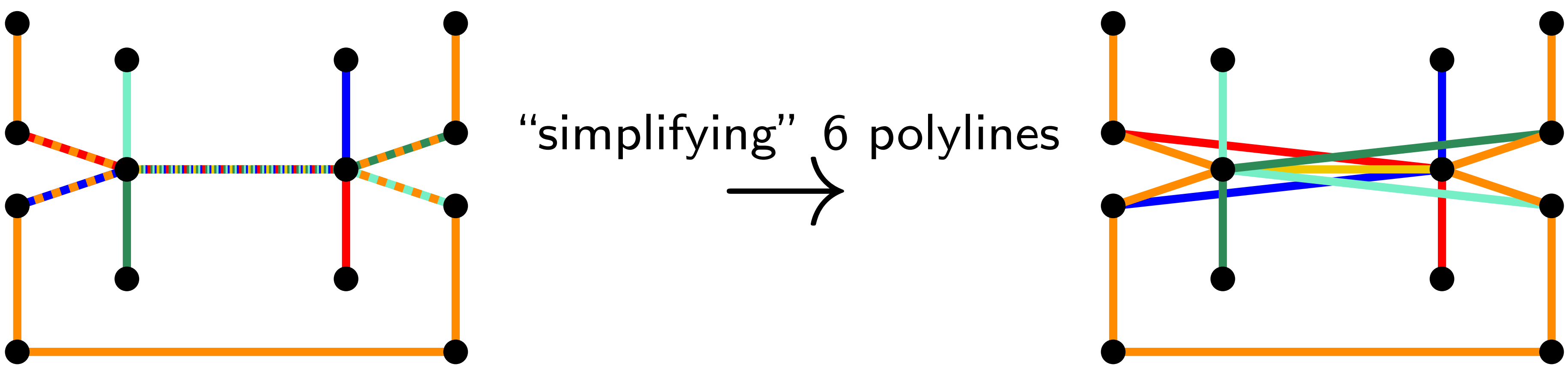
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



⇒ the total complexity might even increase

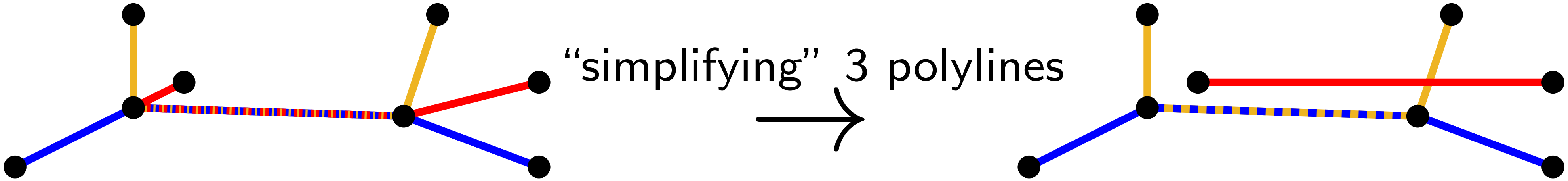


14 bends, 14 segments

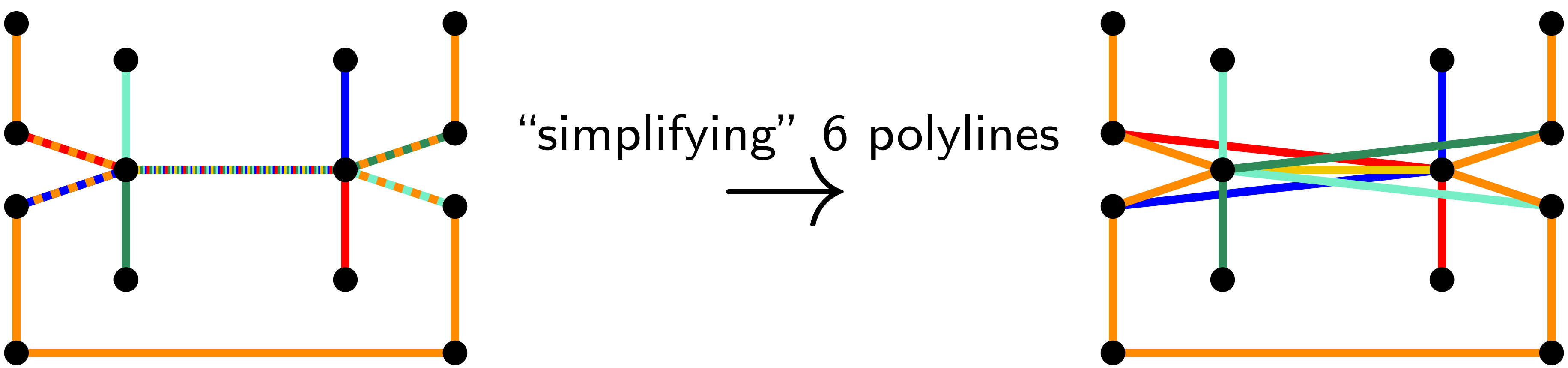
Why Not Simplifying Independently?

A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality



⇒ the total complexity might even increase



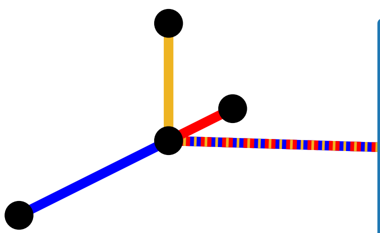
14 bends, 14 segments

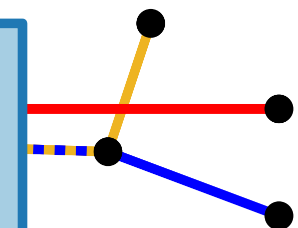
14 bends, 18 segments

Why Not Simplifying Independently?

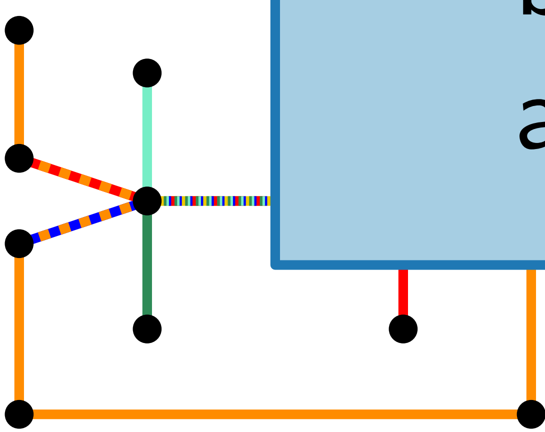
A bend or segment might once be kept and once be discarded

⇒ misleading picture of the reality

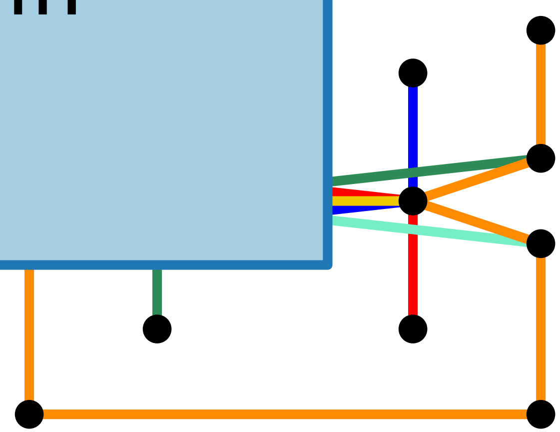




⇒ th



14 bends, 14 segments



14 bends, 18 segments

⇒ Simplify consistently! (keep or discard a bend or segment in all polylines)

Agenda

1. Motivation and Introduction
2. **Problem Definition**
3. Hardness of Approximation (+ Proof Sketch)
4. Bi-Criteria Approximation (+ Proof Sketch)
5. Summary

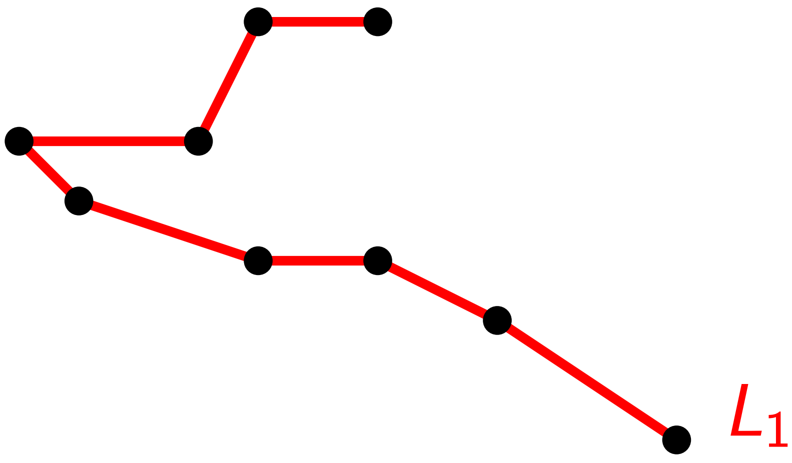
Our Generalization

Our Generalization

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

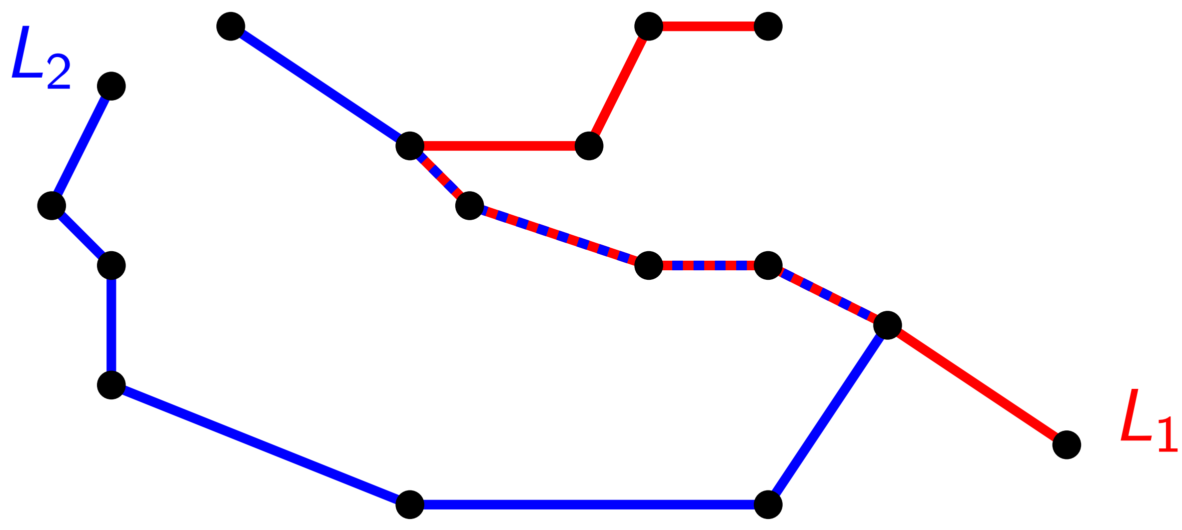
Our Generalization

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



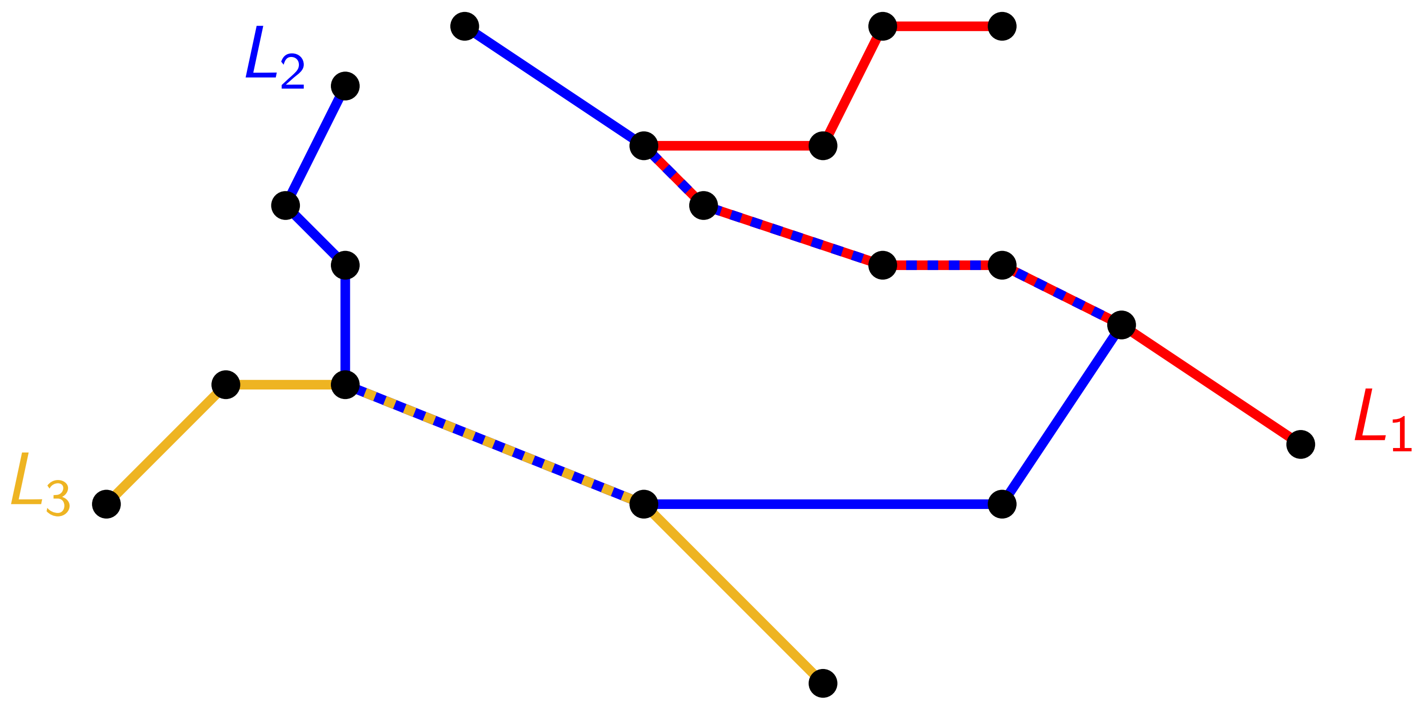
Our Generalization

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



Our Generalization

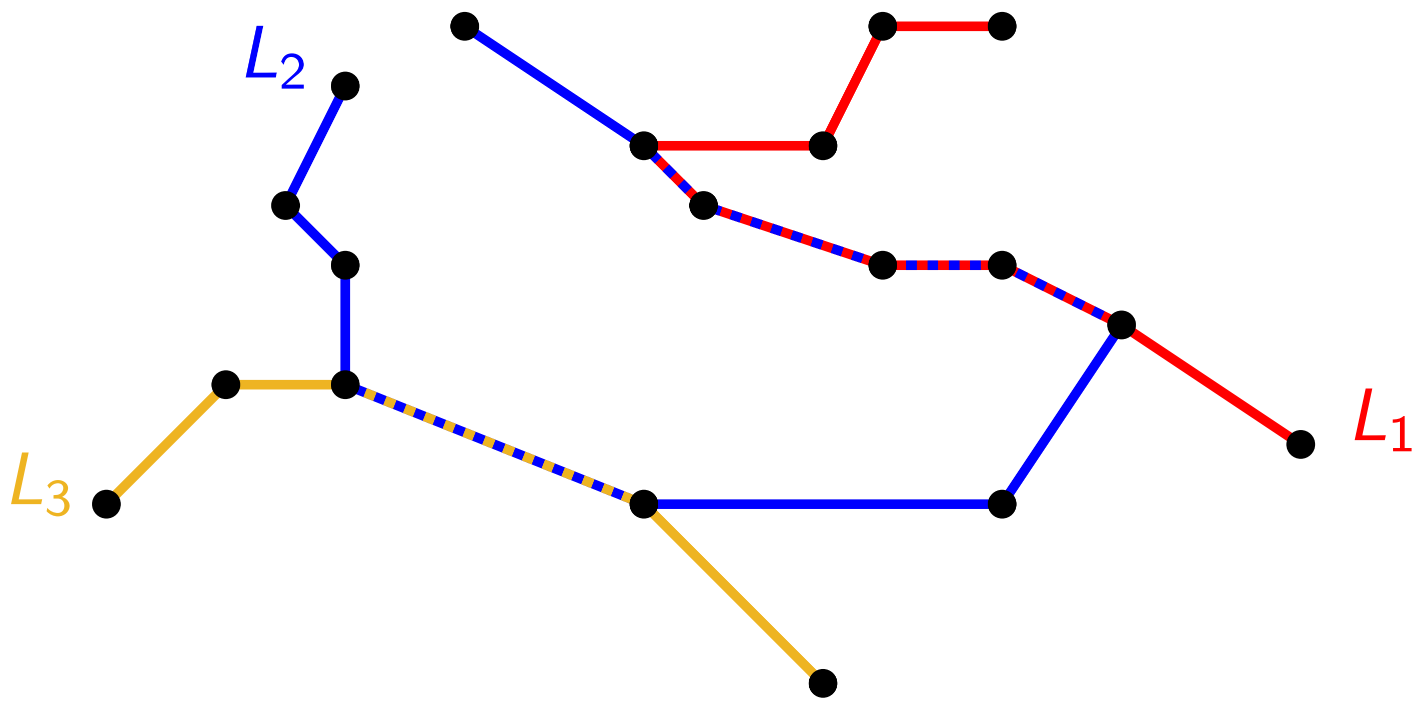
- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments,



Our Generalization

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

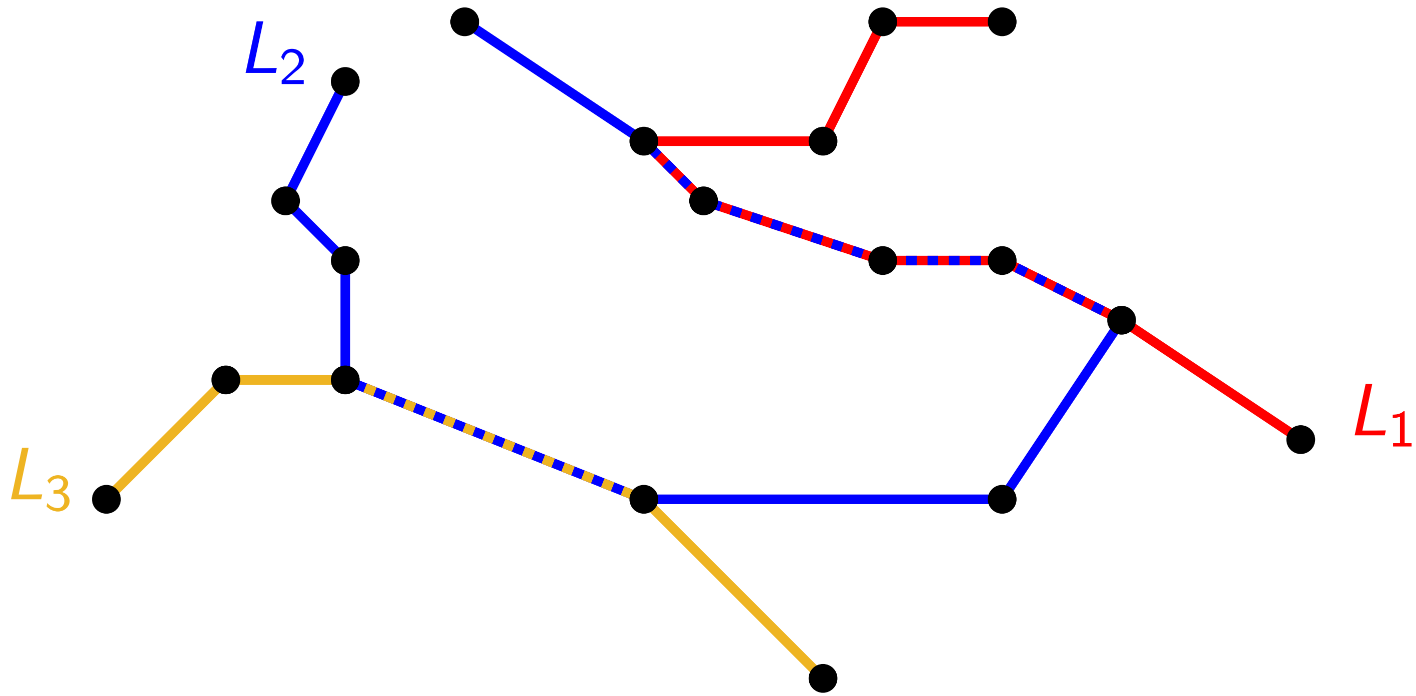
We call the union of their bends B and $n := |B|$



Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

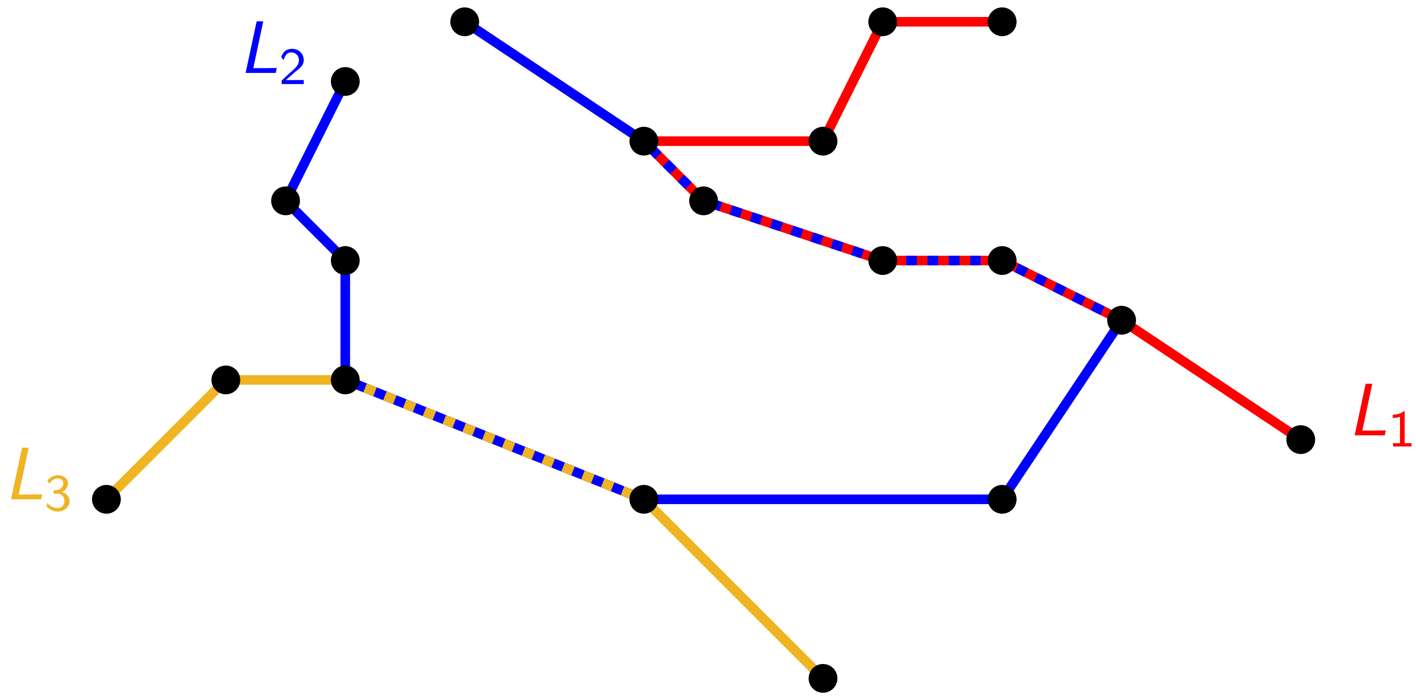
We call the union of their bends B and $n := |B|$



Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ
- Goal: find a $B^* \subseteq B$

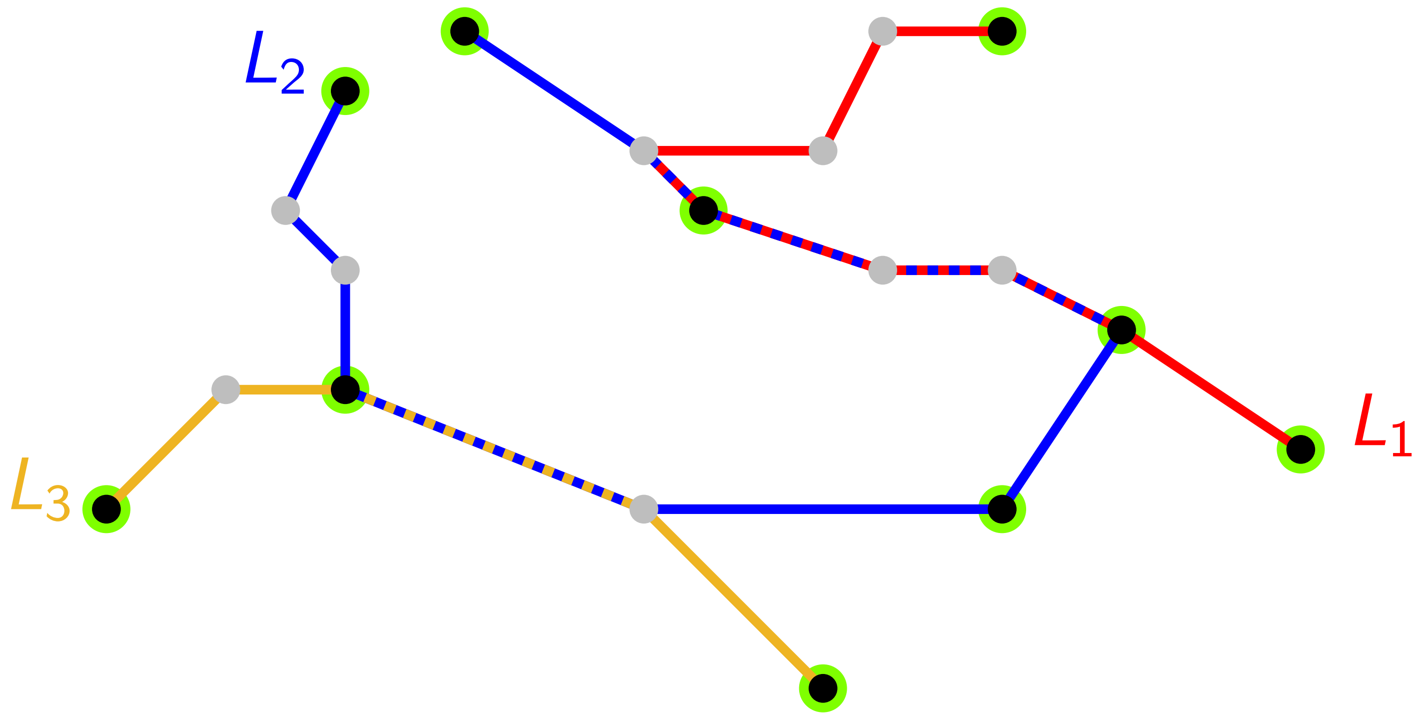
We call the union of their bends B and $n := |B|$



Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ
- Goal: find a $B^* \subseteq B$

We call the union of their bends B and $n := |B|$

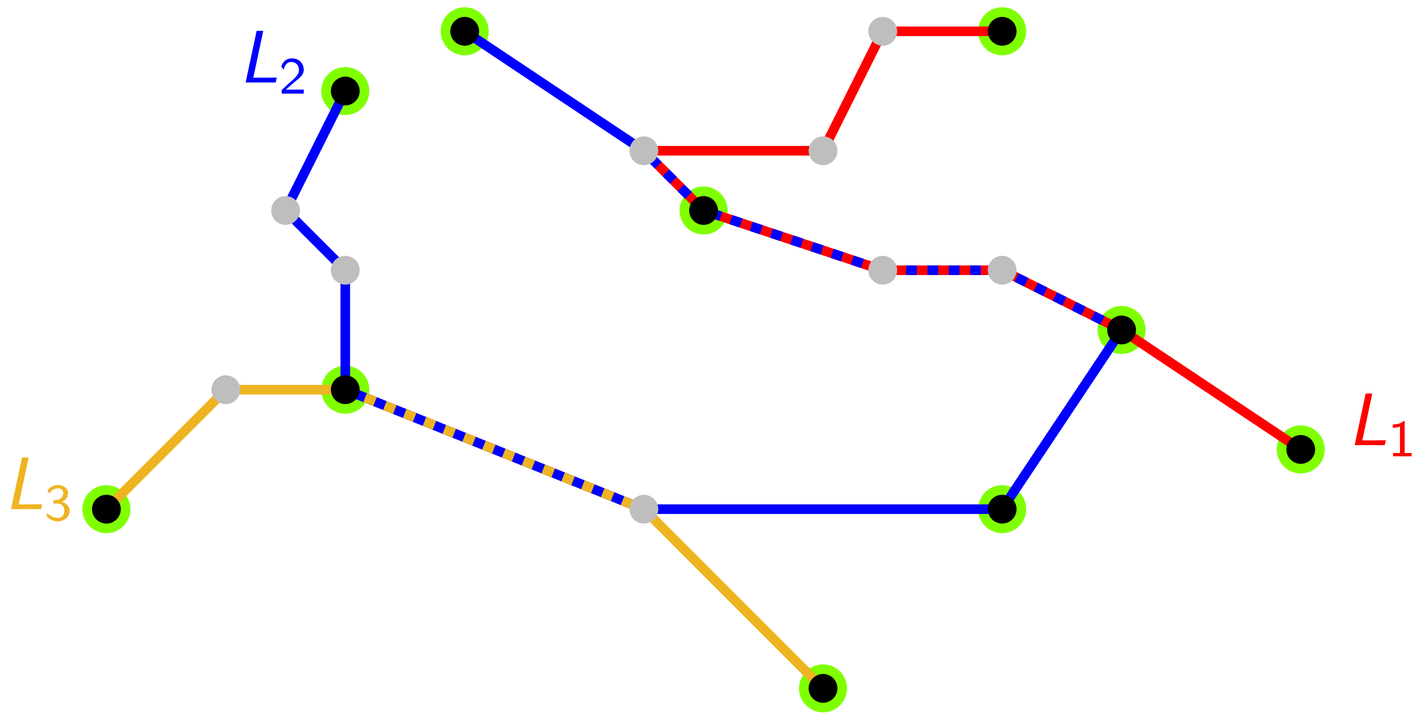


Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

We call the union of their bends B and $n := |B|$

- Goal: find a B^* $\subseteq B$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, s.t.

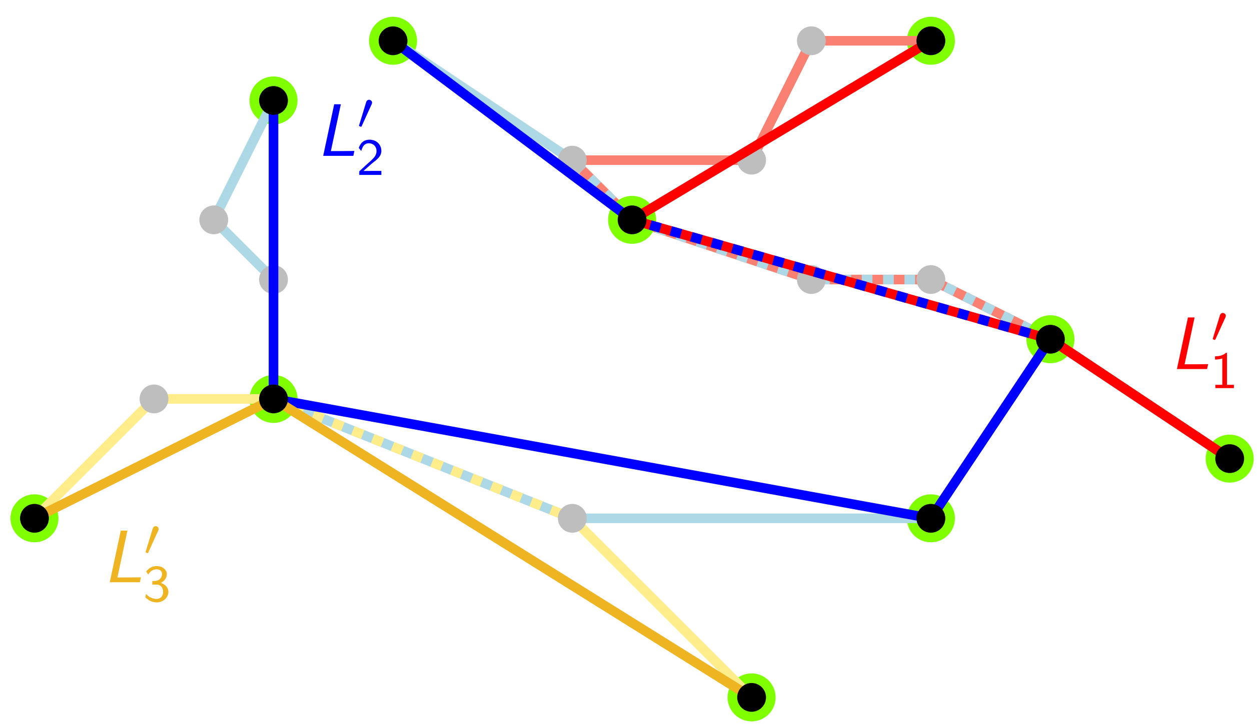


Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

We call the union of their bends B and $n := |B|$

- Goal: find a $B^* \subseteq B$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, s.t.

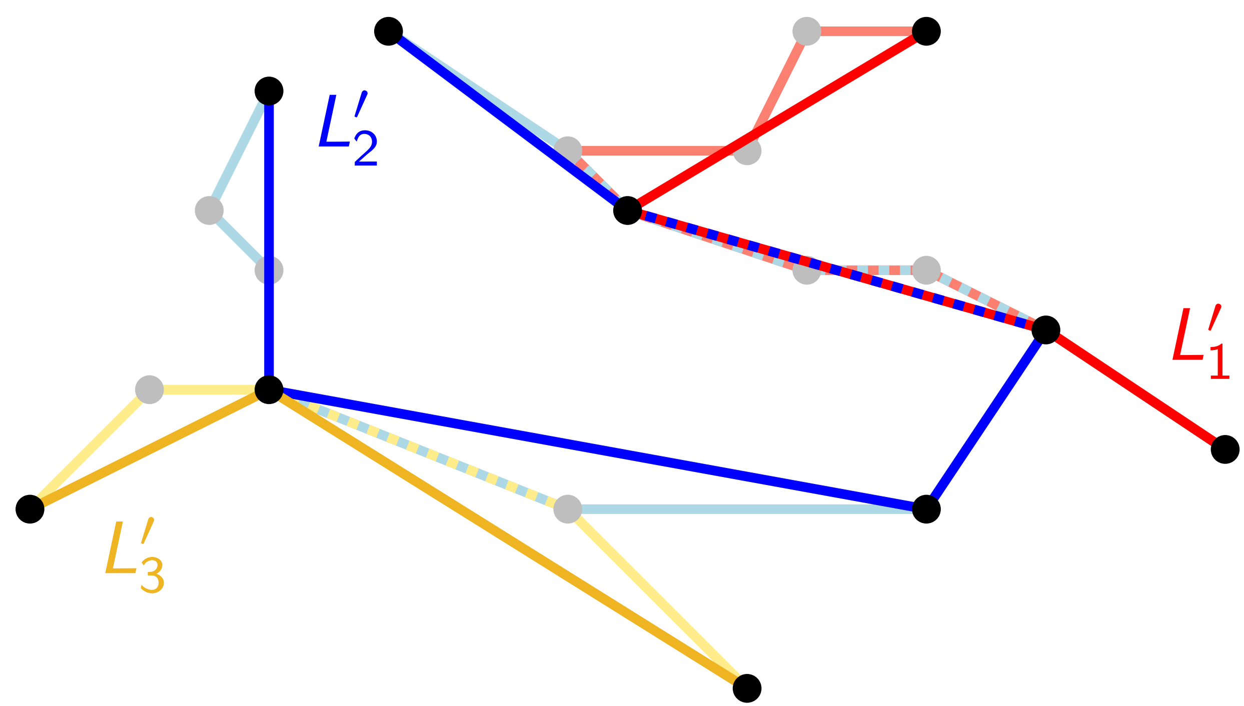


Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

We call the union of their bends B and $n := |B|$

- Goal: find a $B^* \subseteq B$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, s.t.
  – all start and end points of polylines are preserved,

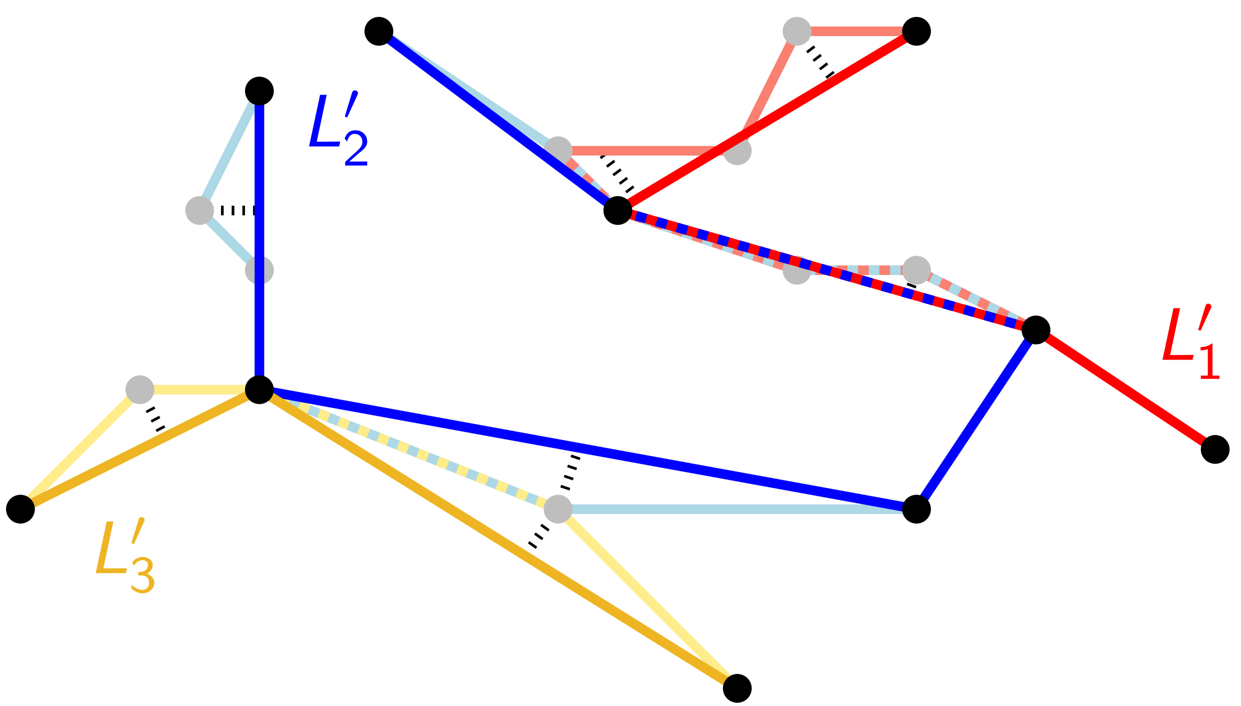


Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

We call the union of their bends B and $n := |B|$

- Goal: find a $B^* \subseteq B$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, s.t.
  - all start and end points of polylines are preserved,
  - there is no L'_i and L_i with $d_{\text{Fréchet}} > \delta$, and

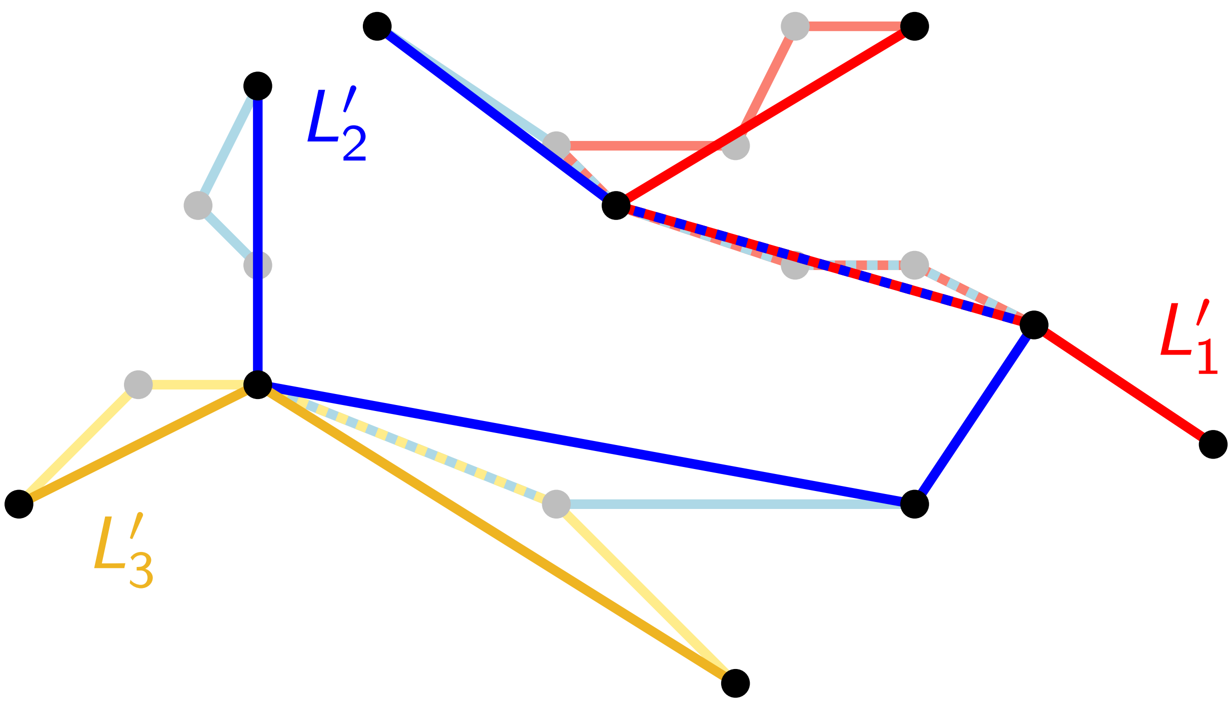


Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

> We call the union of their bends B and $n := |B|$

- Goal: find a $B^* \subseteq B$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, s.t.
  - all start and end points of polylines are preserved,
  - there is no L'_i and L_i with $d_{\text{Fréchet}} > \delta$, and
  - $|B^*|$ is minimum.

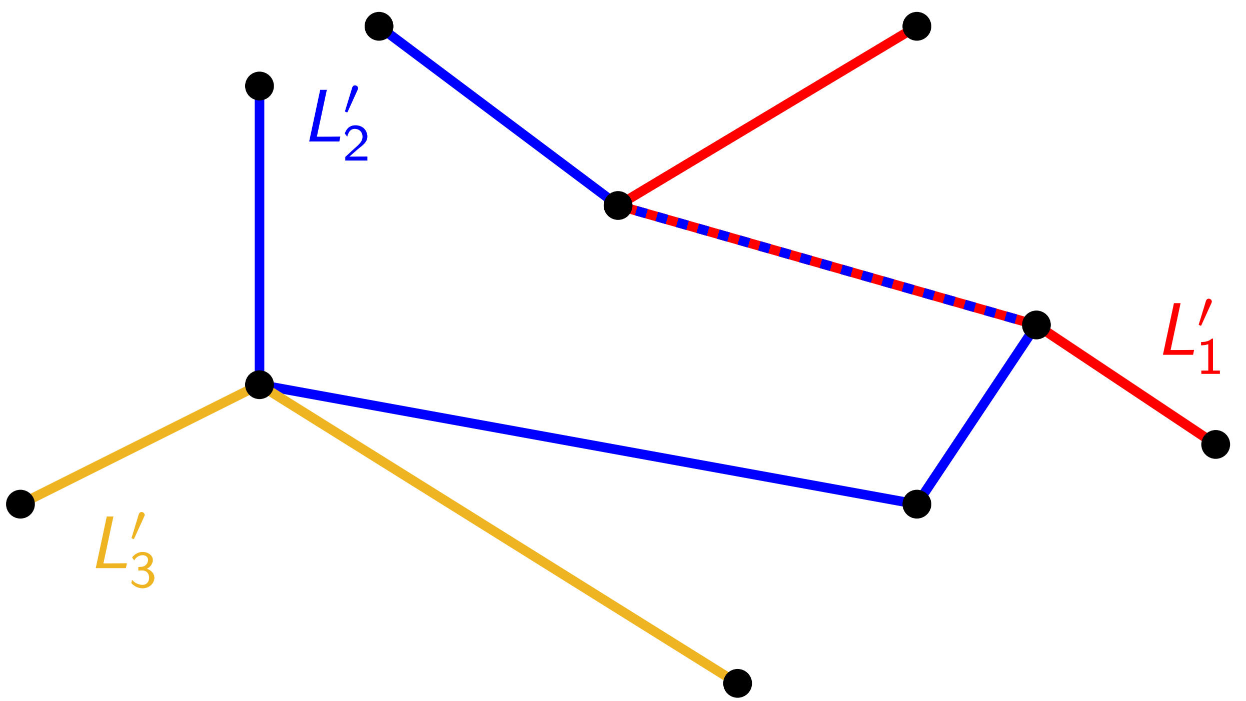


Our Generalization

- Given: set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing bends and segments, distance threshold δ

We call the union of their bends B and $n := |B|$

- Goal: find a $B^* \subseteq B$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, s.t.
  - all start and end points of polylines are preserved,
  - there is no L'_i and L_i with $d_{\text{Fréchet}} > \delta$, and
  - $|B^*|$ is minimum.



Agenda

1. Motivation and Introduction
2. Problem Definition
3. **Hardness of Approximation (+ Proof Sketch)**
4. Bi-Criteria Approximation (+ Proof Sketch)
5. Summary

Hardness

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

$\Rightarrow$ not FPT in # polylines ℓ

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

$\Rightarrow$ not FPT in # polylines ℓ

Proof Sketch:

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

$\Rightarrow$ not FPT in # polylines ℓ

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS)
  (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [Halldórsson '93])

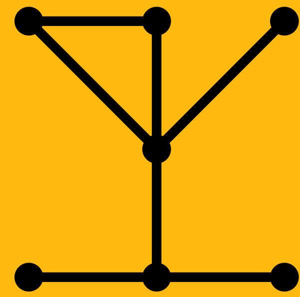
Hardness

Theorem 1:
Simplifying a bun
within a factor of

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS)
  (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$
  [Halldórsson '93])

Hardness



Theorem 1:
Simplifying a bun
within a factor of

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS)
  (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$
  [Halldórsson '93])

Hardness

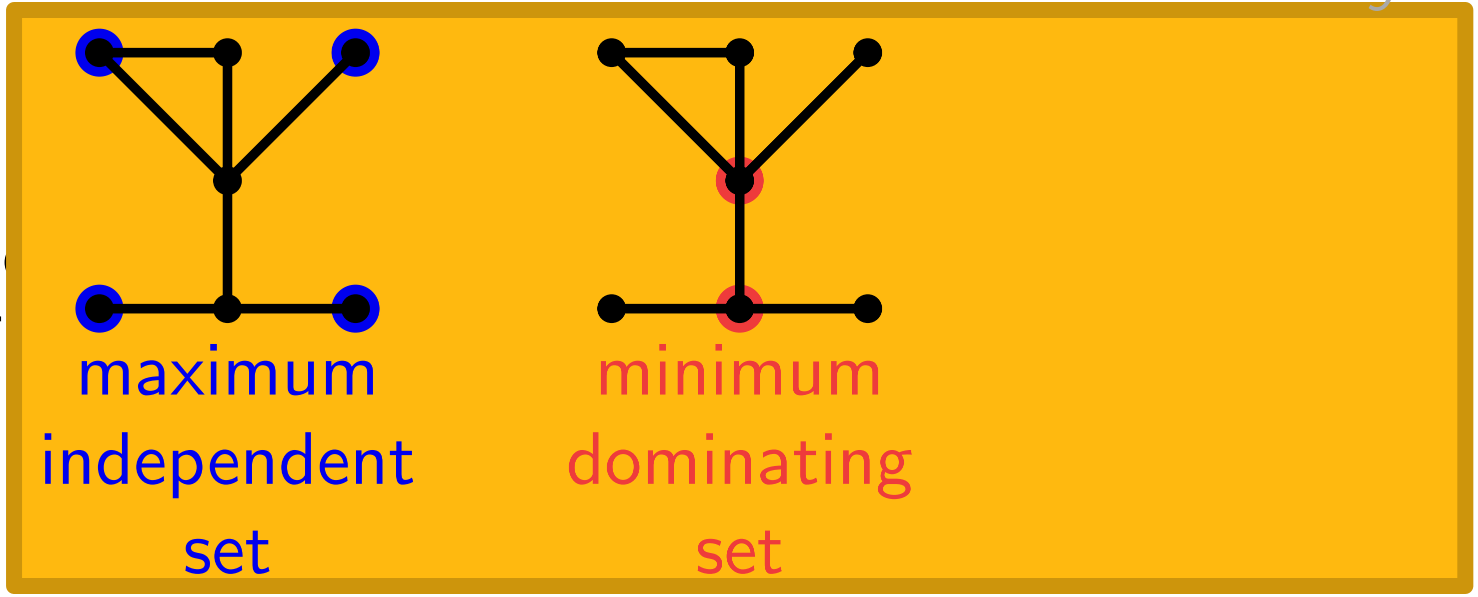
Theorem 1:
Simplifying a bun
within a factor of

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS)
  (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$
  [Halldórsson '93])

Hardness

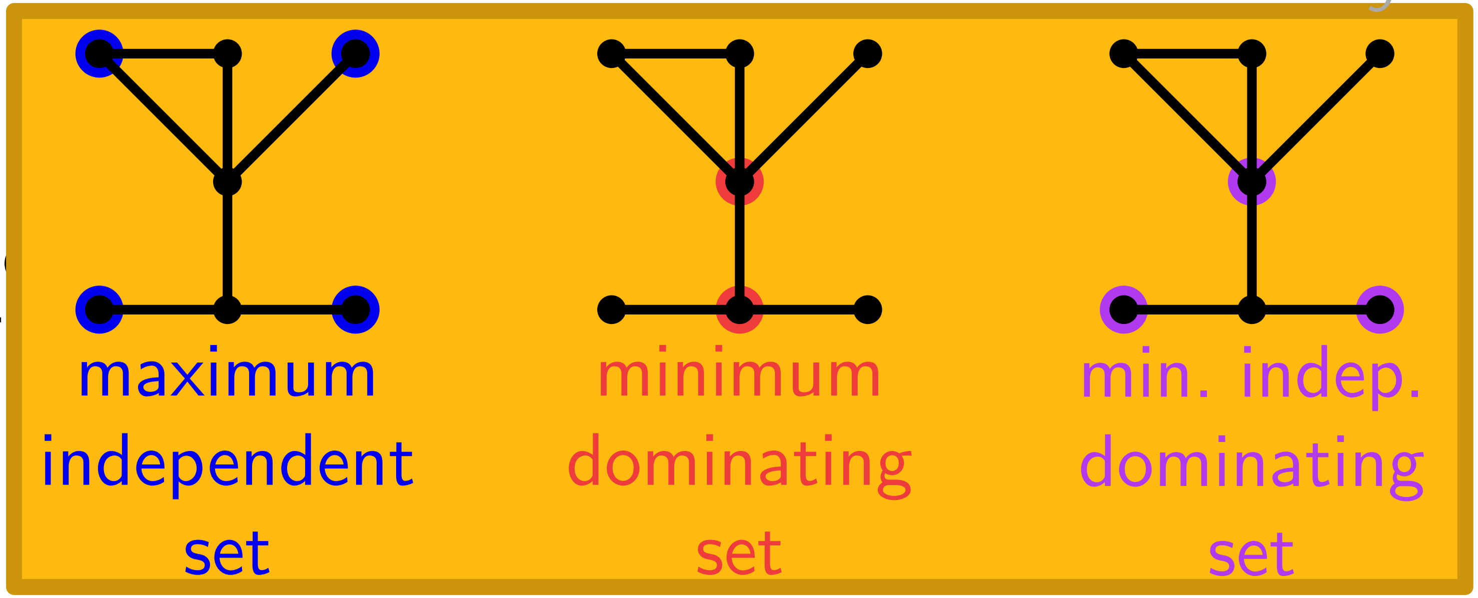
Theorem 1:
Simplifying a bun
within a factor of



Proof Sketch:

- Reduce from minimum independent dominating set (MIDS) (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [Halldórsson '93])

Hardness



Theorem 1:
Simplifying a bun
within a factor of

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS) (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [Halldórsson '93])

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

$\Rightarrow$ not FPT in # polylines ℓ

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS) (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [Halldórsson '93])
- We use vertex gadgets to indicate if a vertex is in the set V^*.

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

$\Rightarrow$ not FPT in # polylines ℓ

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS) (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [Halldórsson '93])
- We use vertex gadgets to indicate if a vertex is in the set V^*.
- We use edge and neighborhood gadgets to make V^* independent and dominating, respectively.

Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$ for any $\varepsilon > 0$ even for 2 polylines.

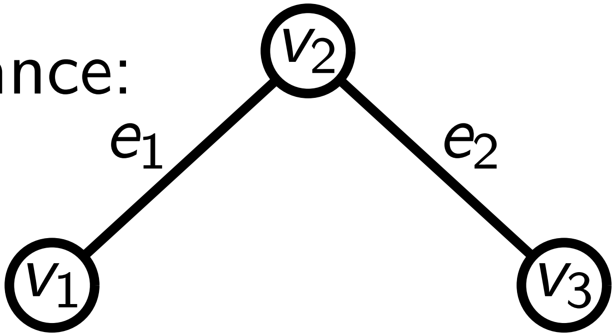
$\Rightarrow$ not FPT in # polylines ℓ

Proof Sketch:

- Reduce from minimum independent dominating set (MIDS) (NP-h to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [Halldórsson '93])
- We use vertex gadgets to indicate if a vertex is in the set V^*.
- We use edge and neighborhood gadgets to make V^* independent and dominating, respectively.
- Connecting all vertex gadgets and connecting all edge and neighborhood gadgets gives us 2 polylines

Full Example

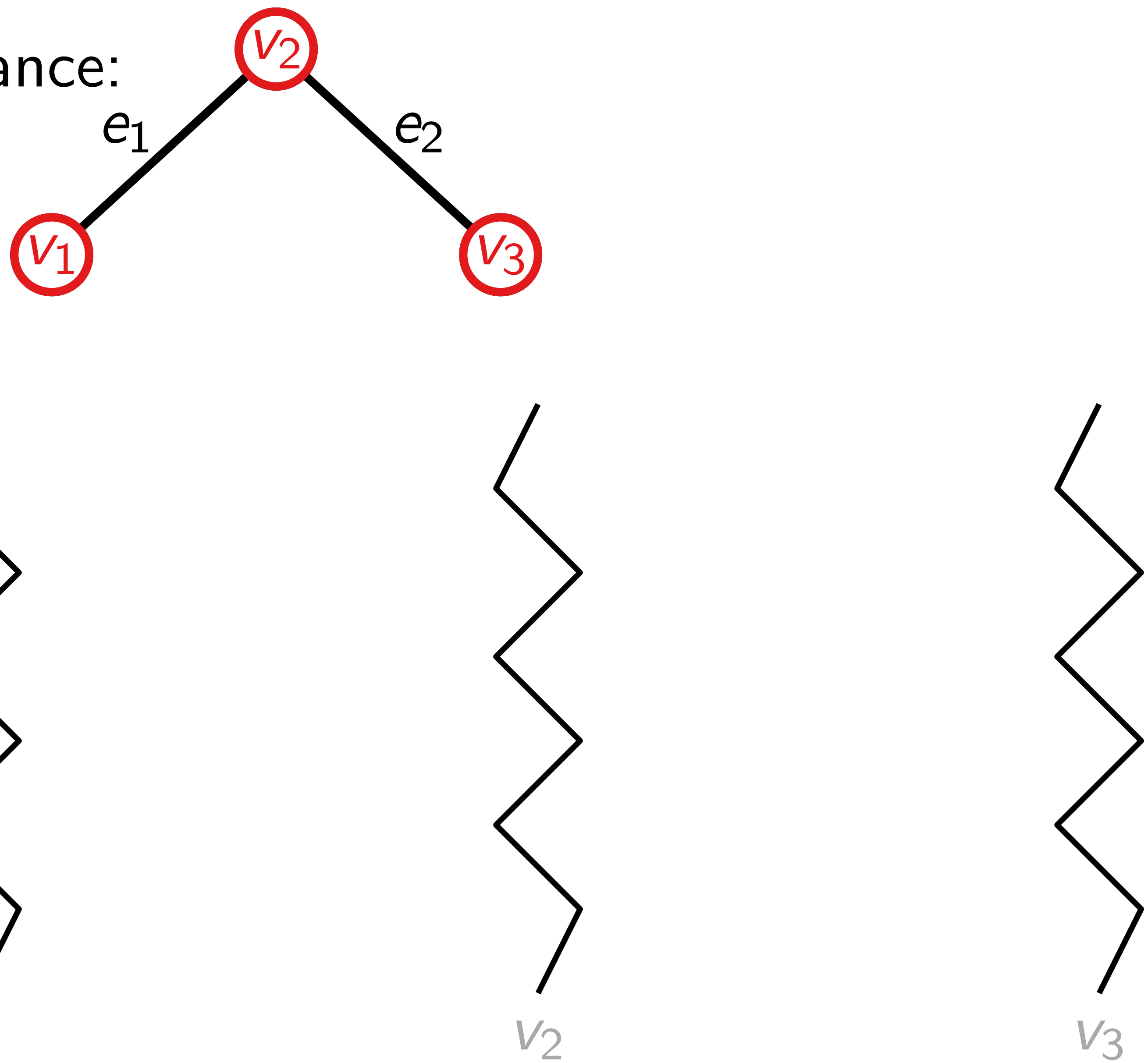
MIDS instance:



$\hat{n} = 3$

Full Example

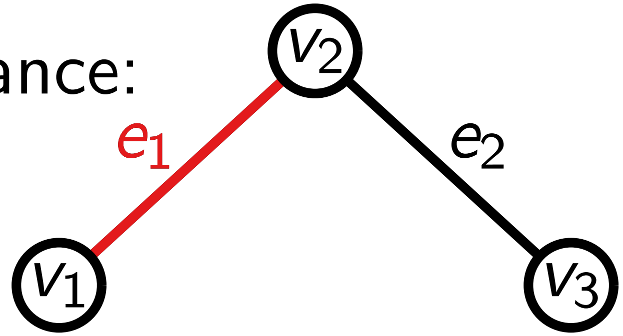
MIDS instance:



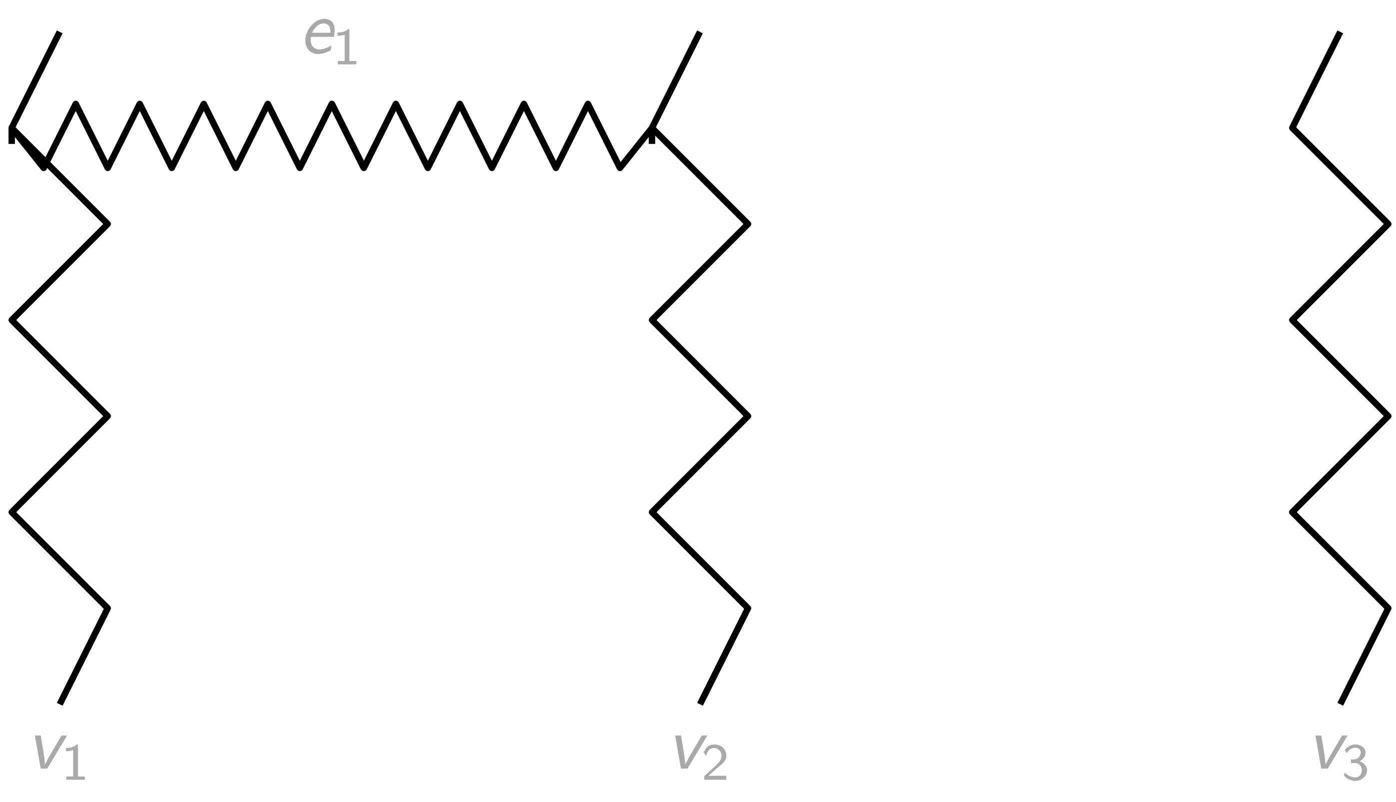
$$\hat{n} = 3$$
$$2\hat{n} = 6$$

Full Example

MIDS instance:

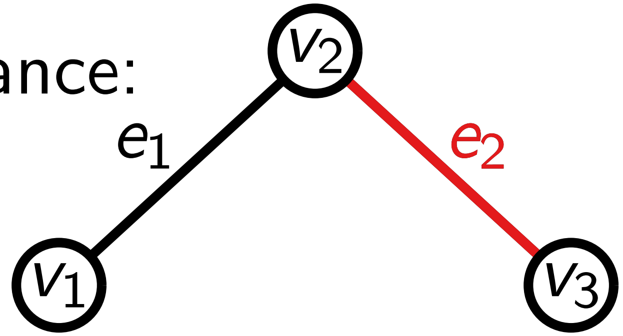


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

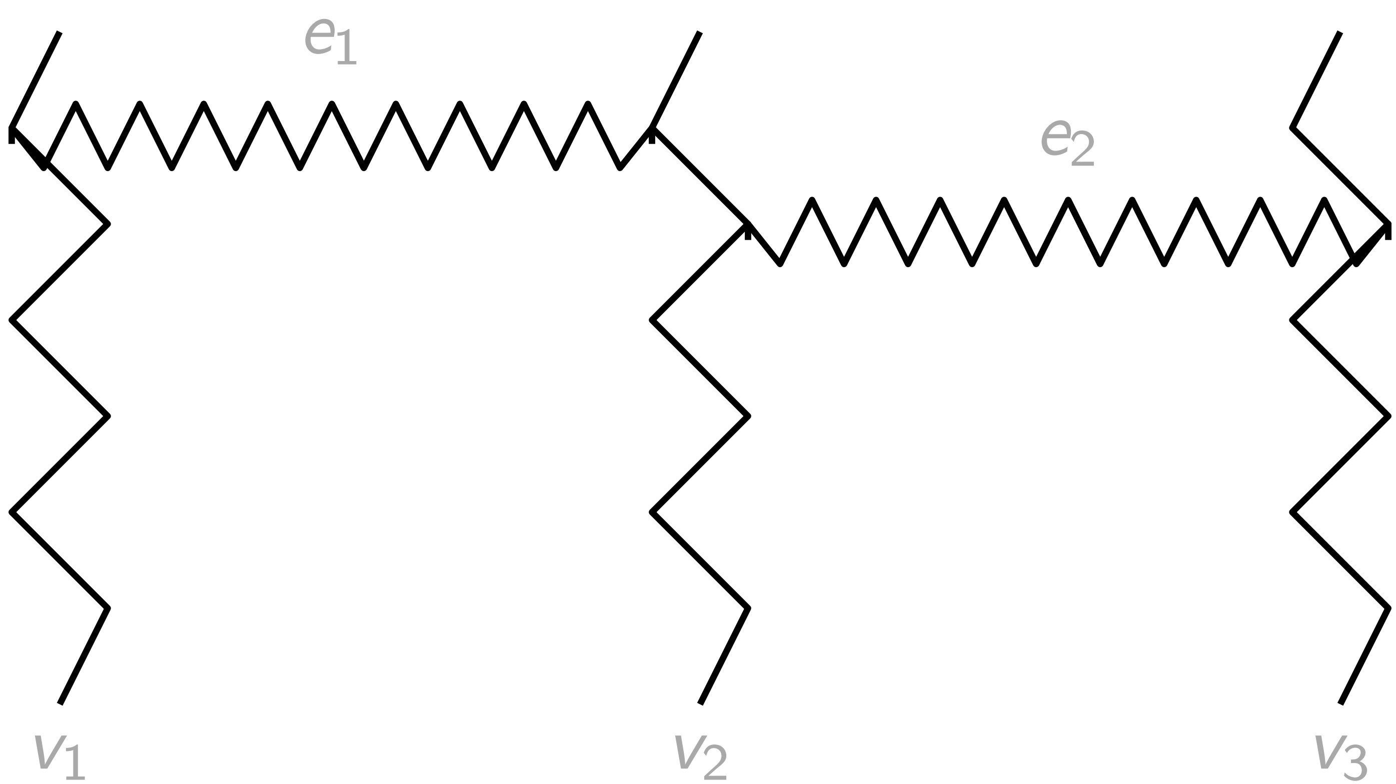


Full Example

MIDS instance:

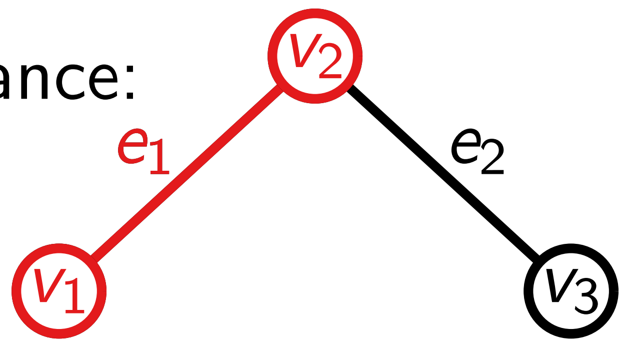


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

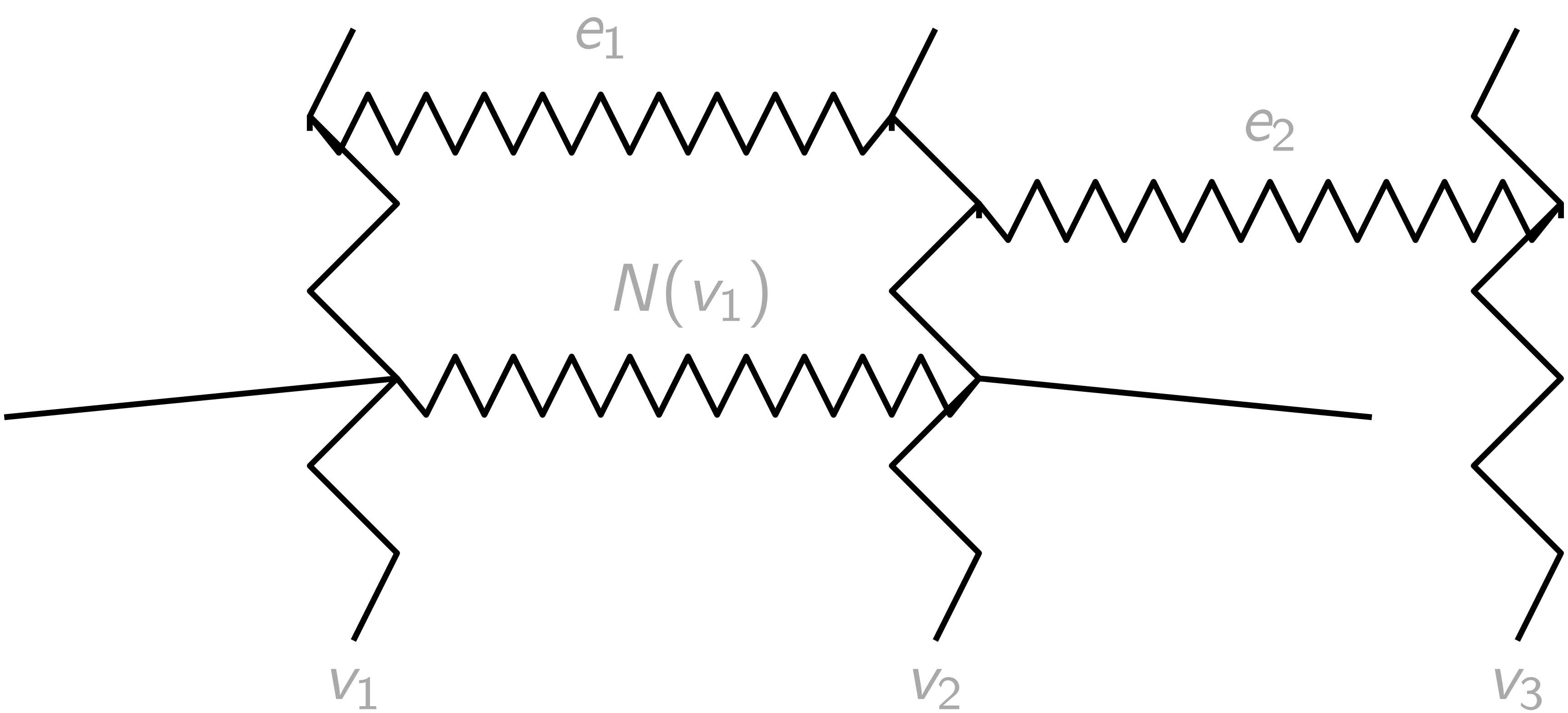


Full Example

MIDS instance:

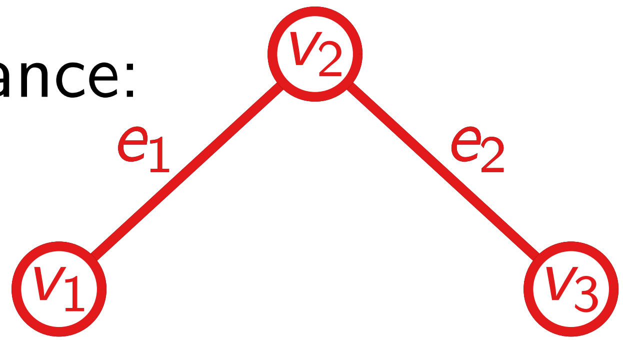


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

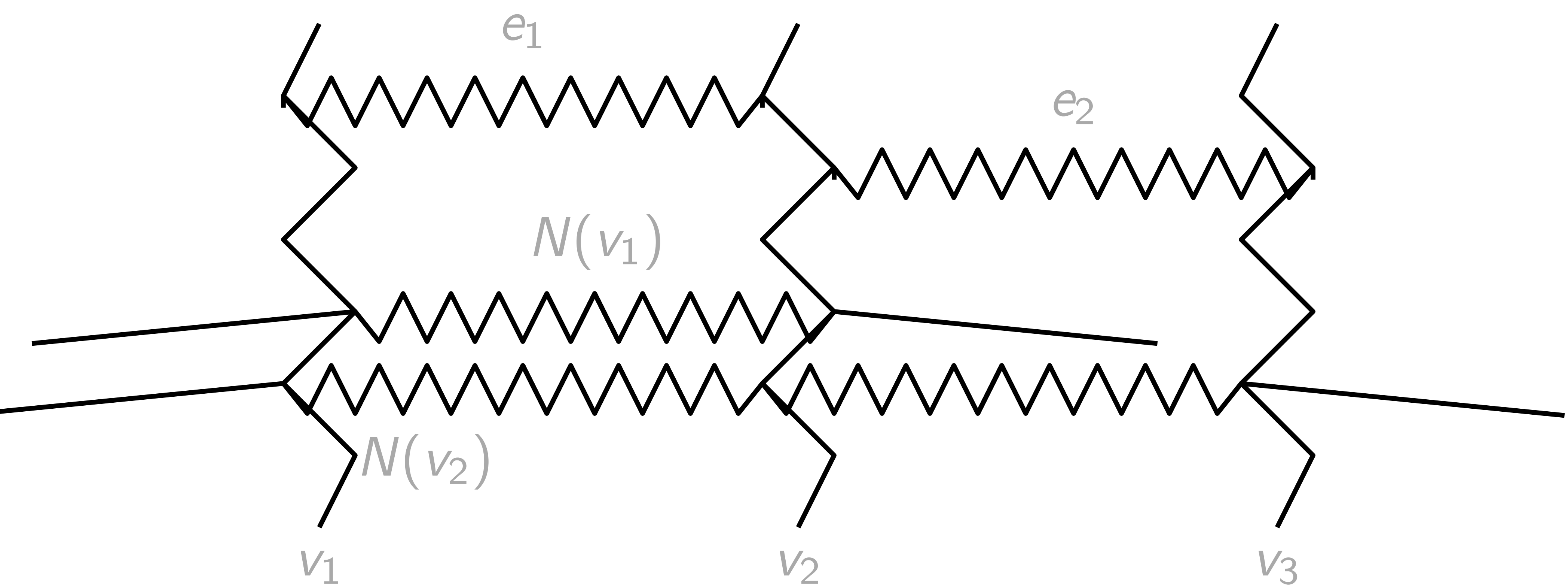


Full Example

MIDS instance:

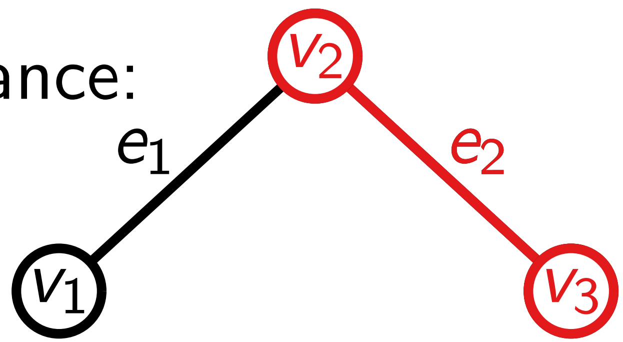


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

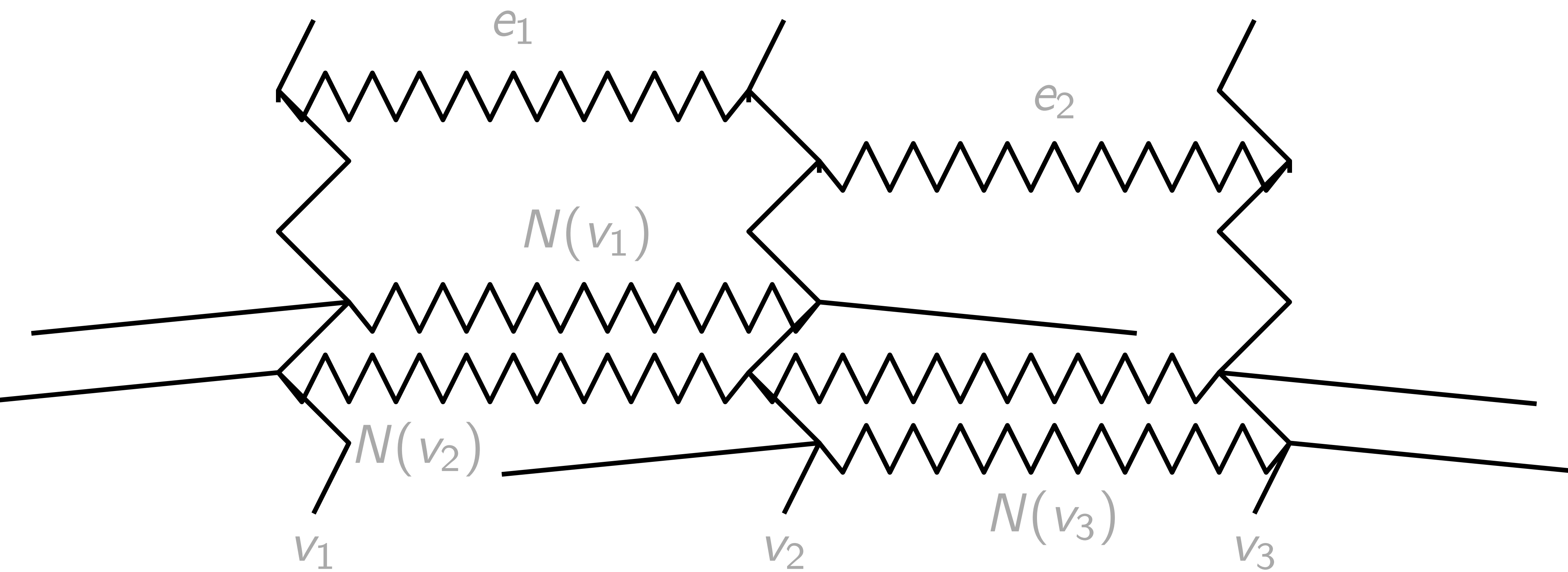


Full Example

MIDS instance:

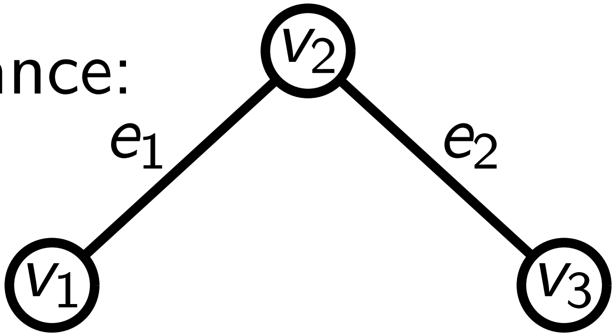


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

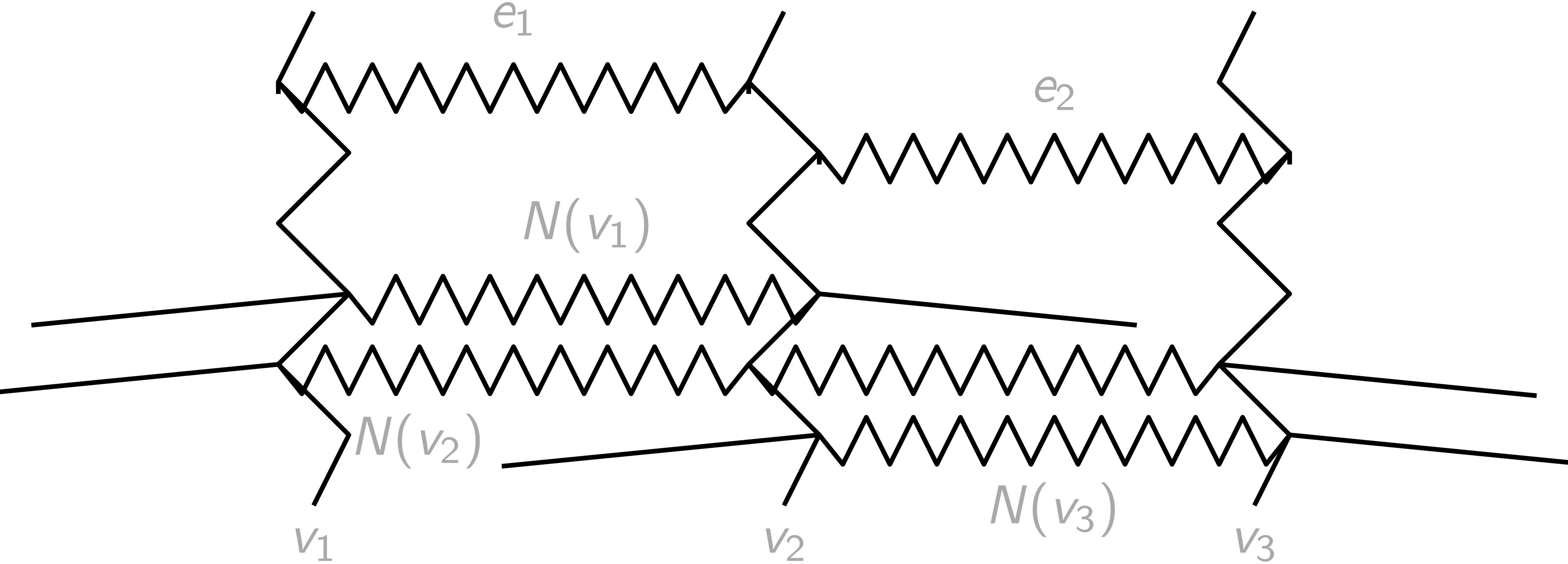


Full Example

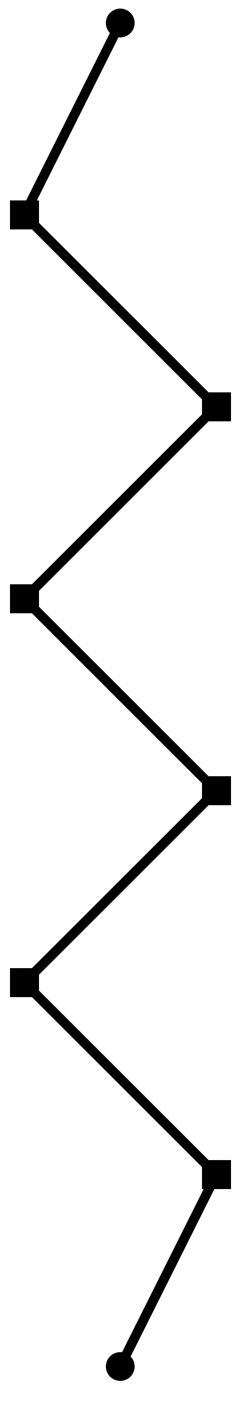
MIDS instance:



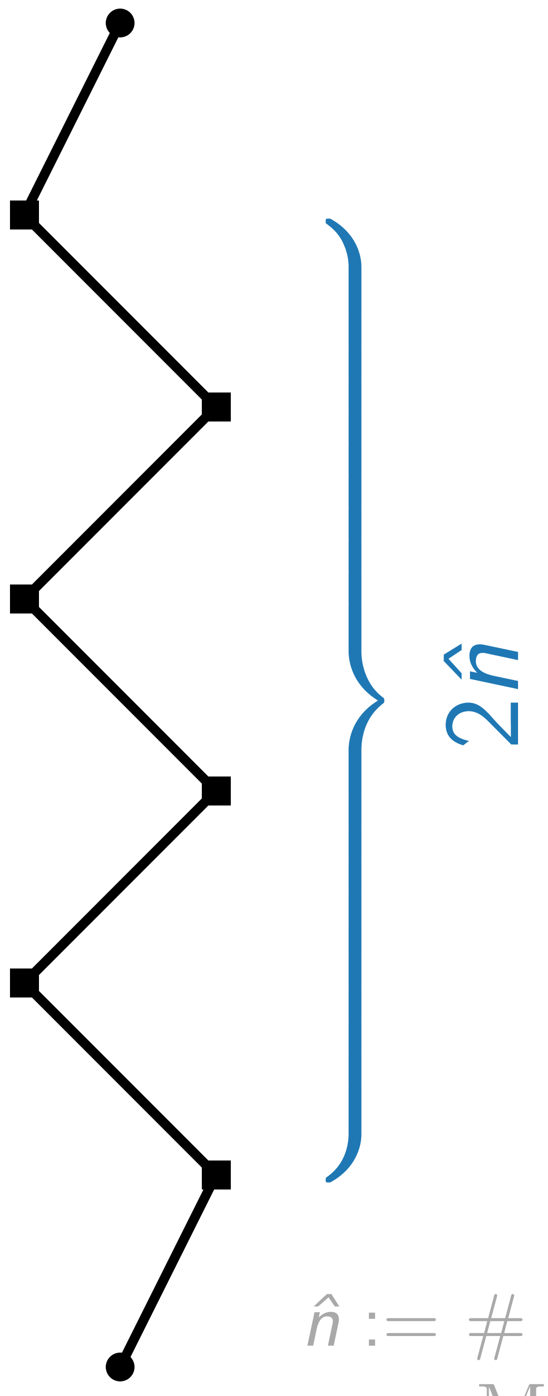
$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$



Vertex Gadget

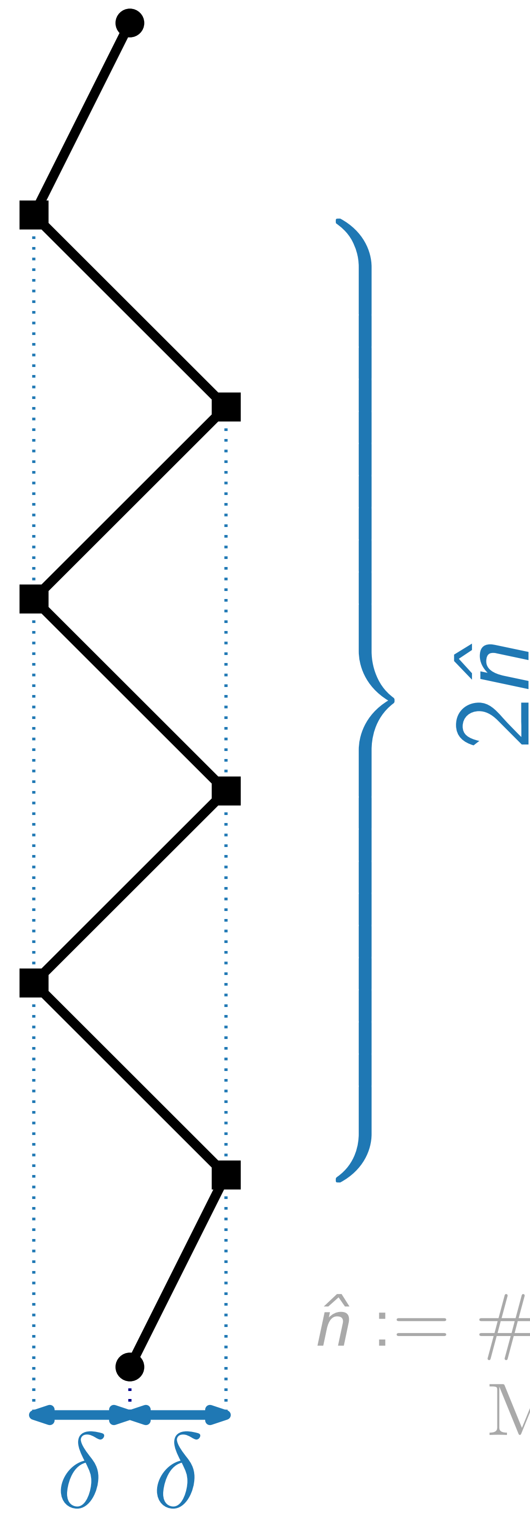


Vertex Gadget



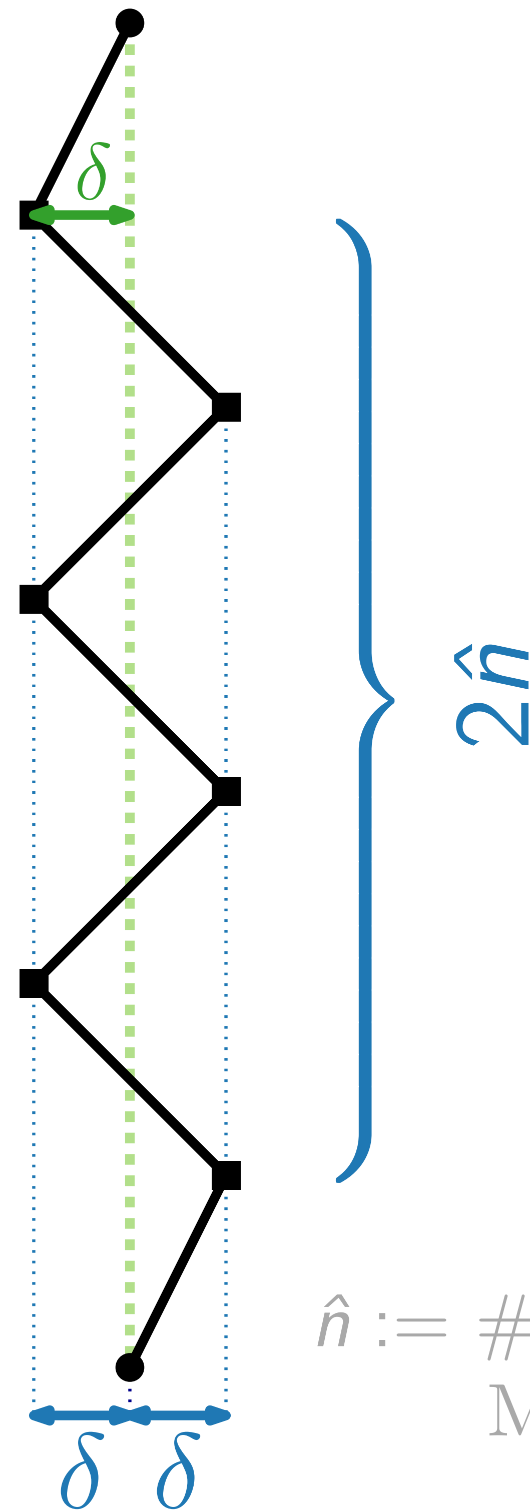
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



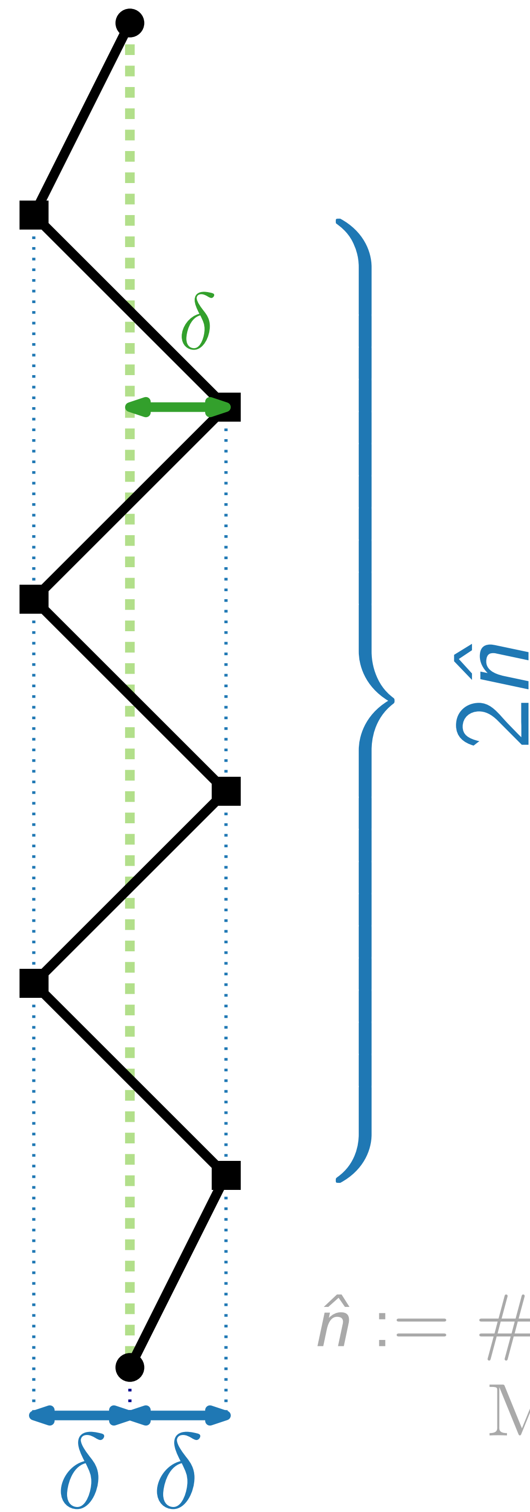
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



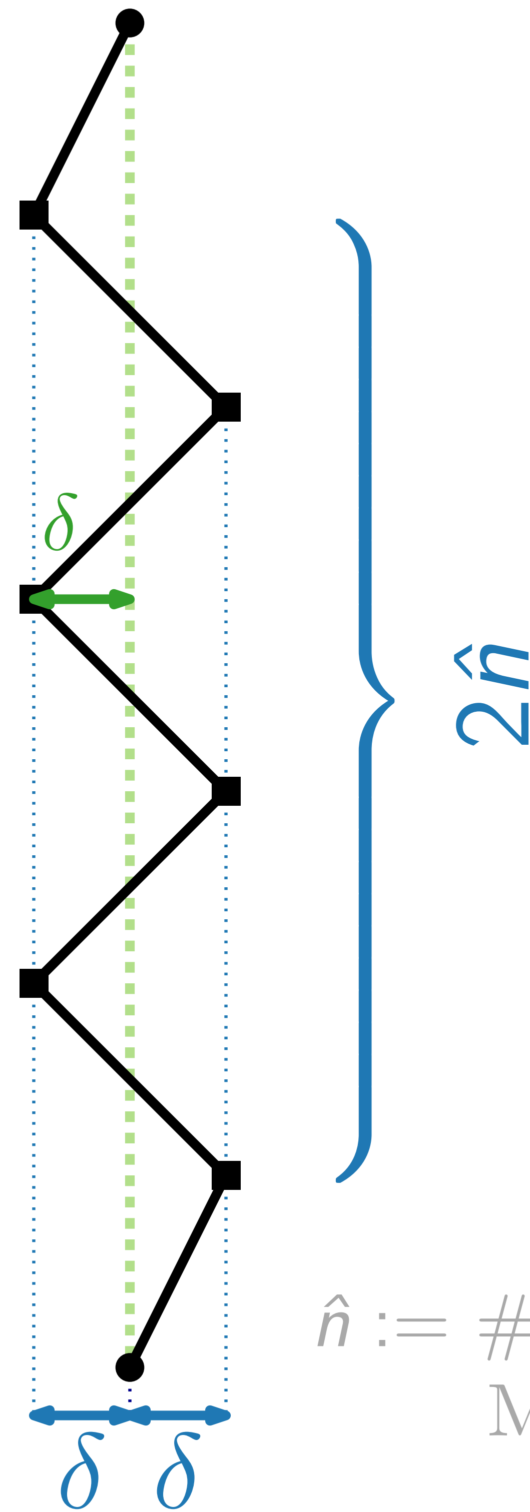
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



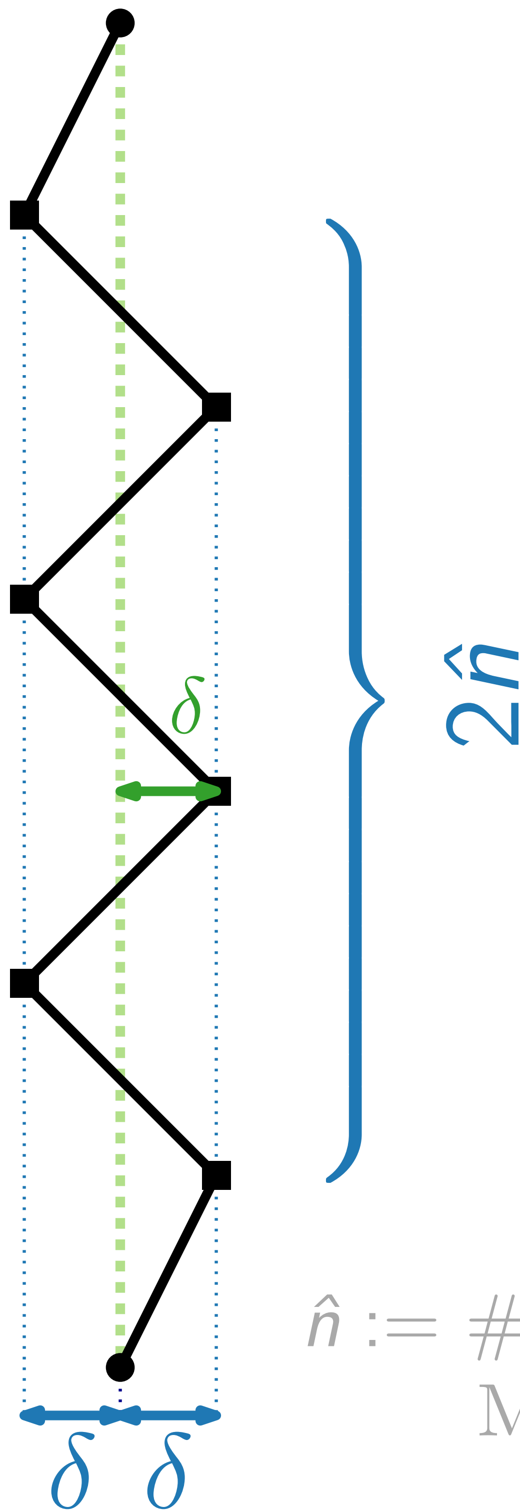
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



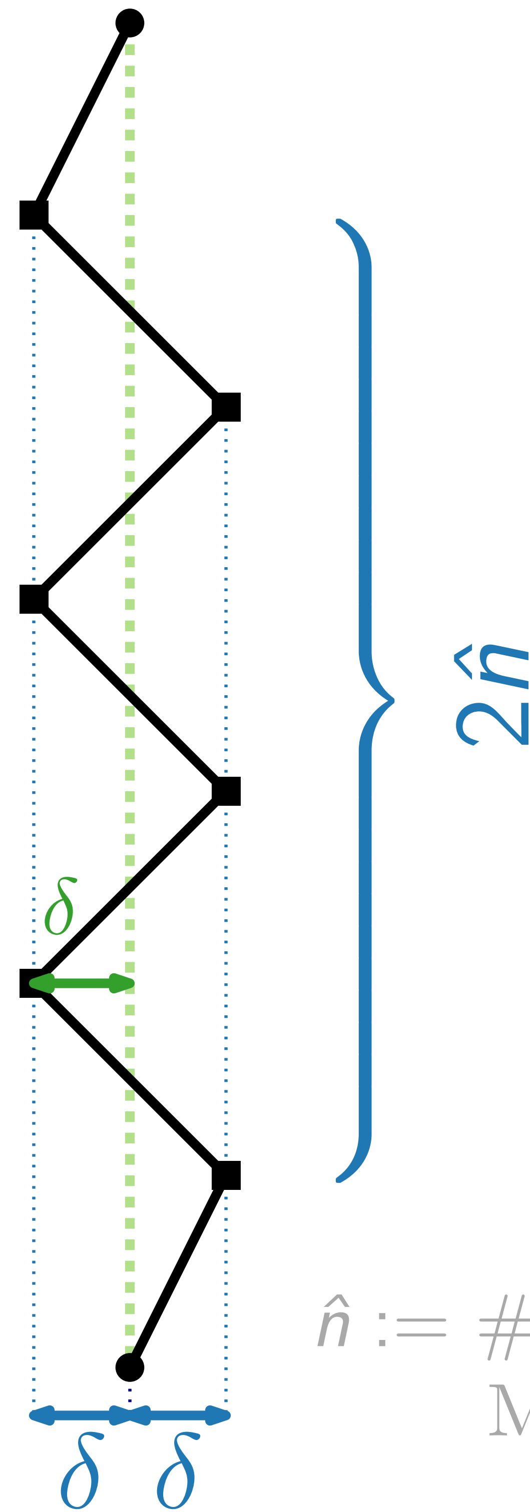
$\hat{n} :=$ # vertices in MIDS graph

Vertex Gadget



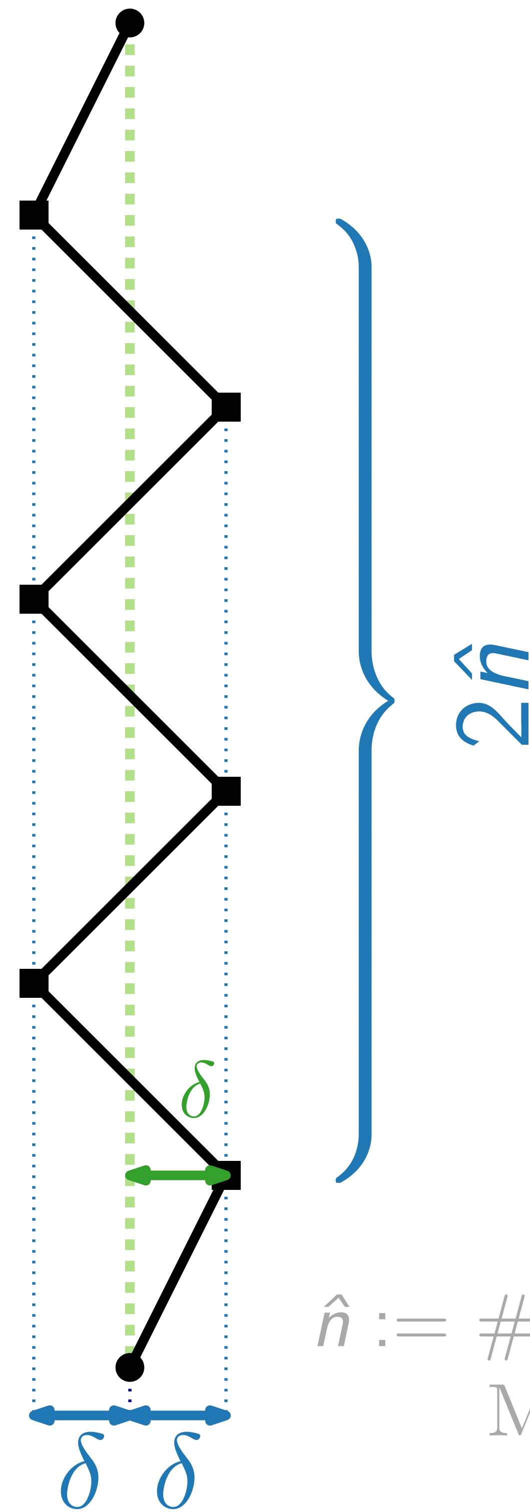
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



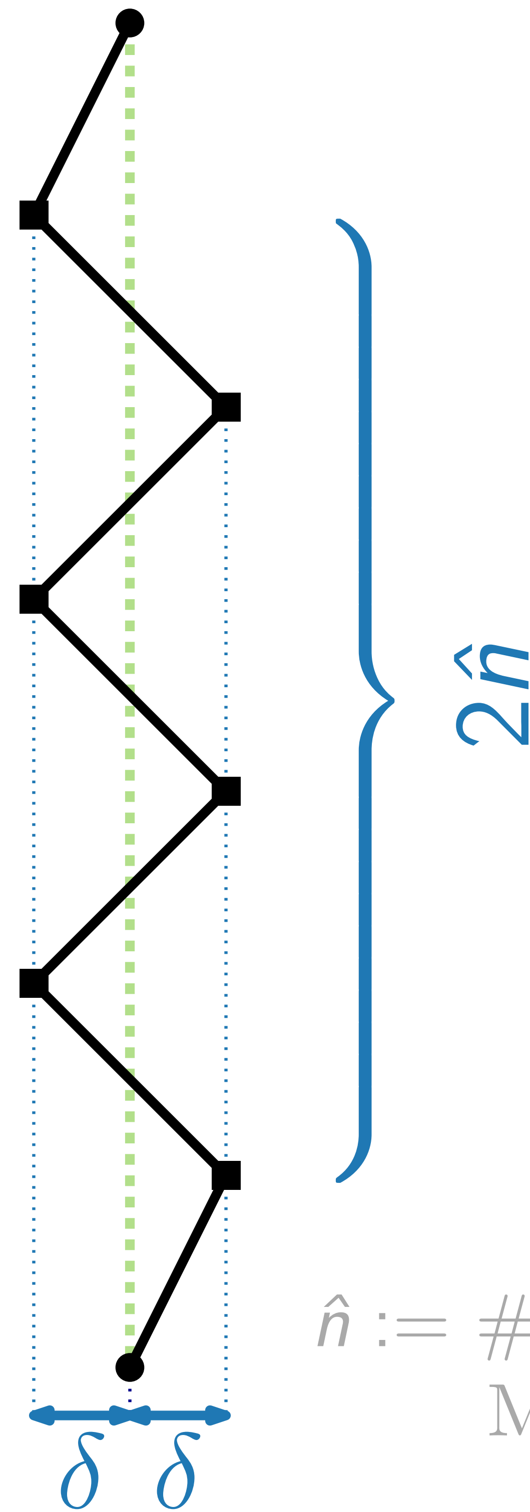
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



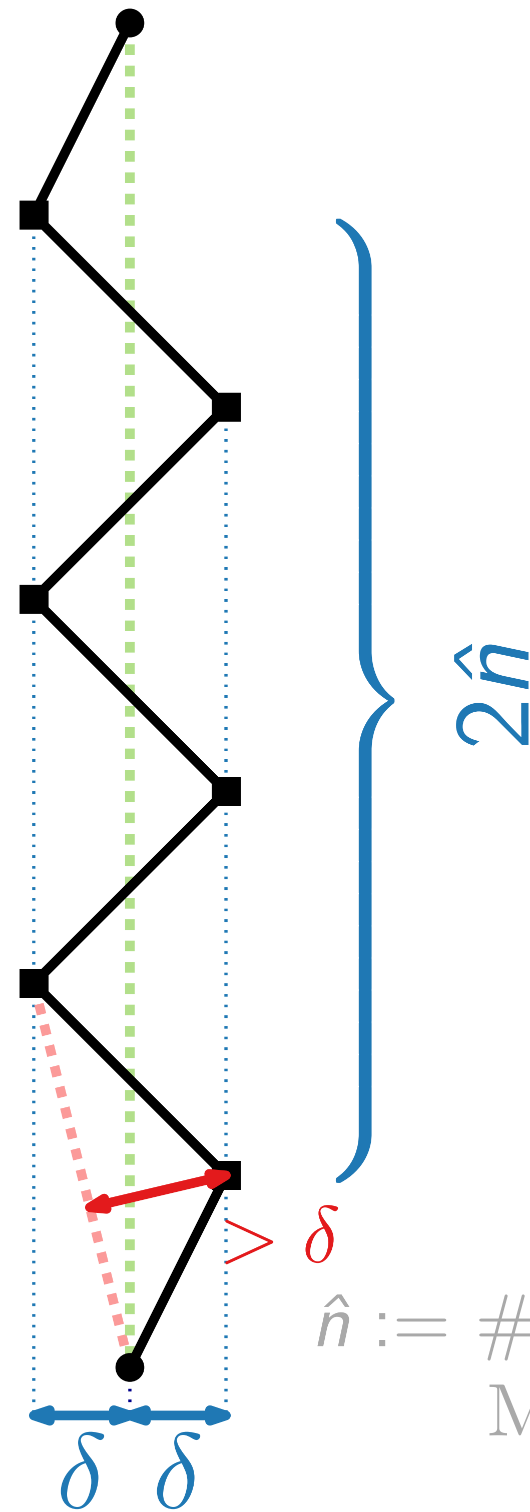
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



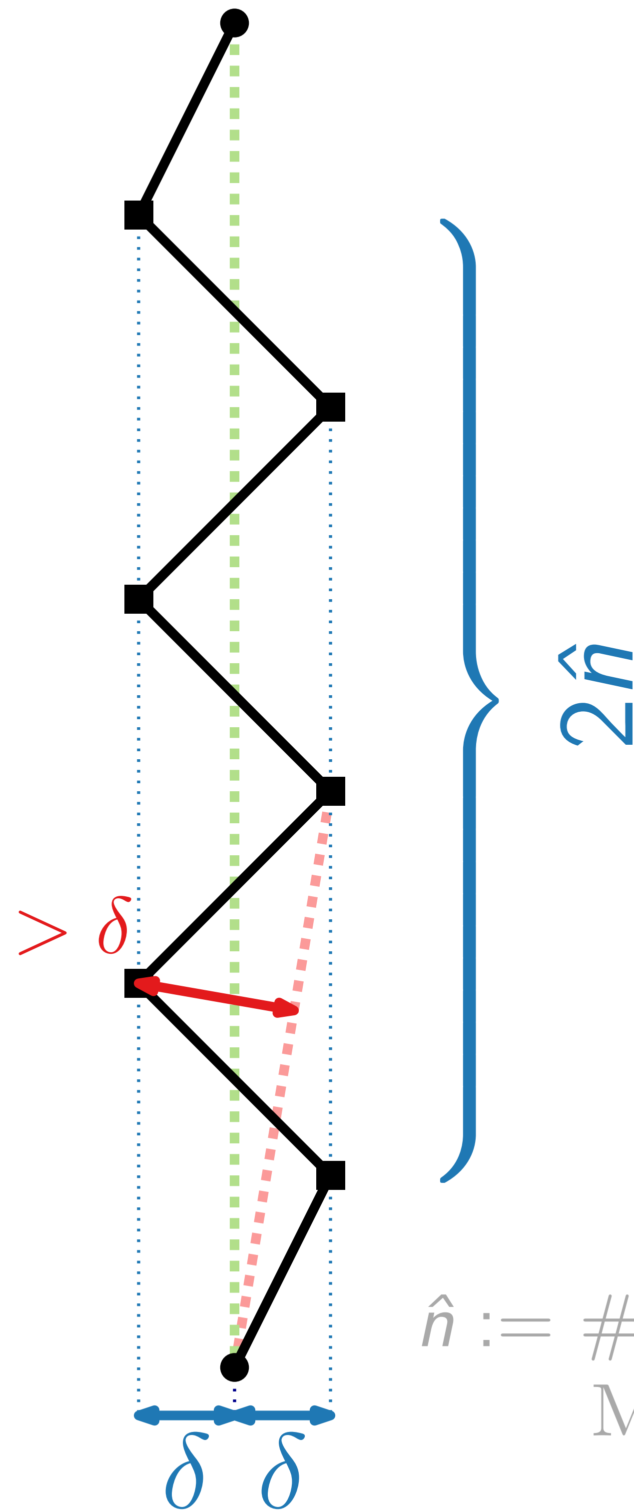
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



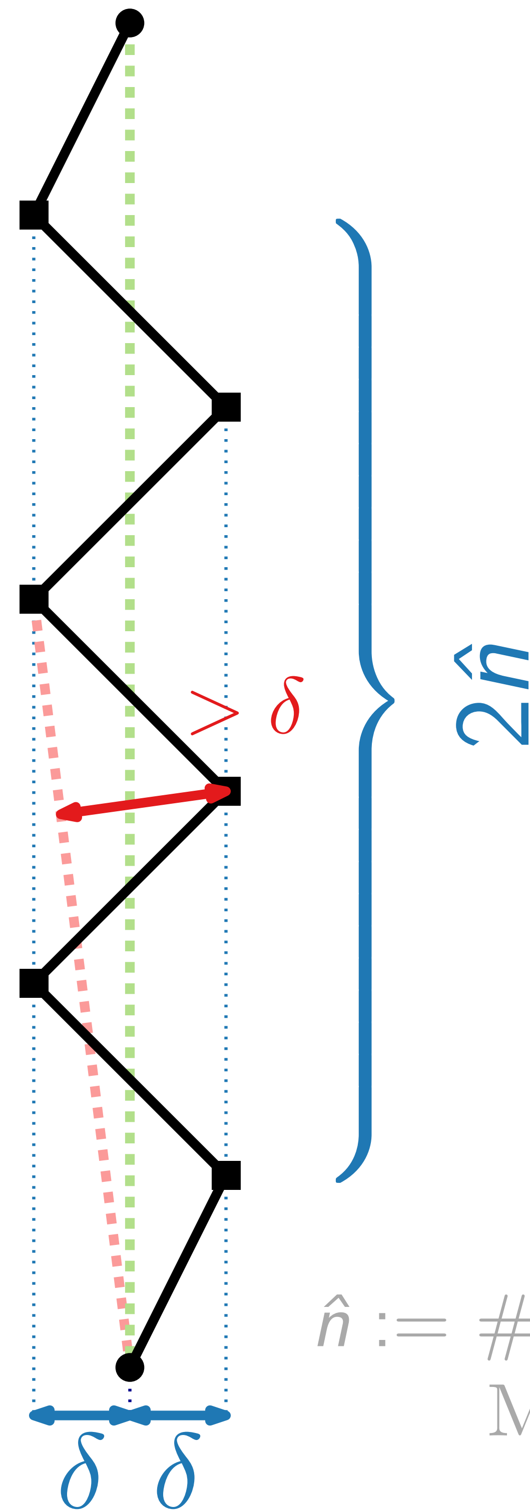
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



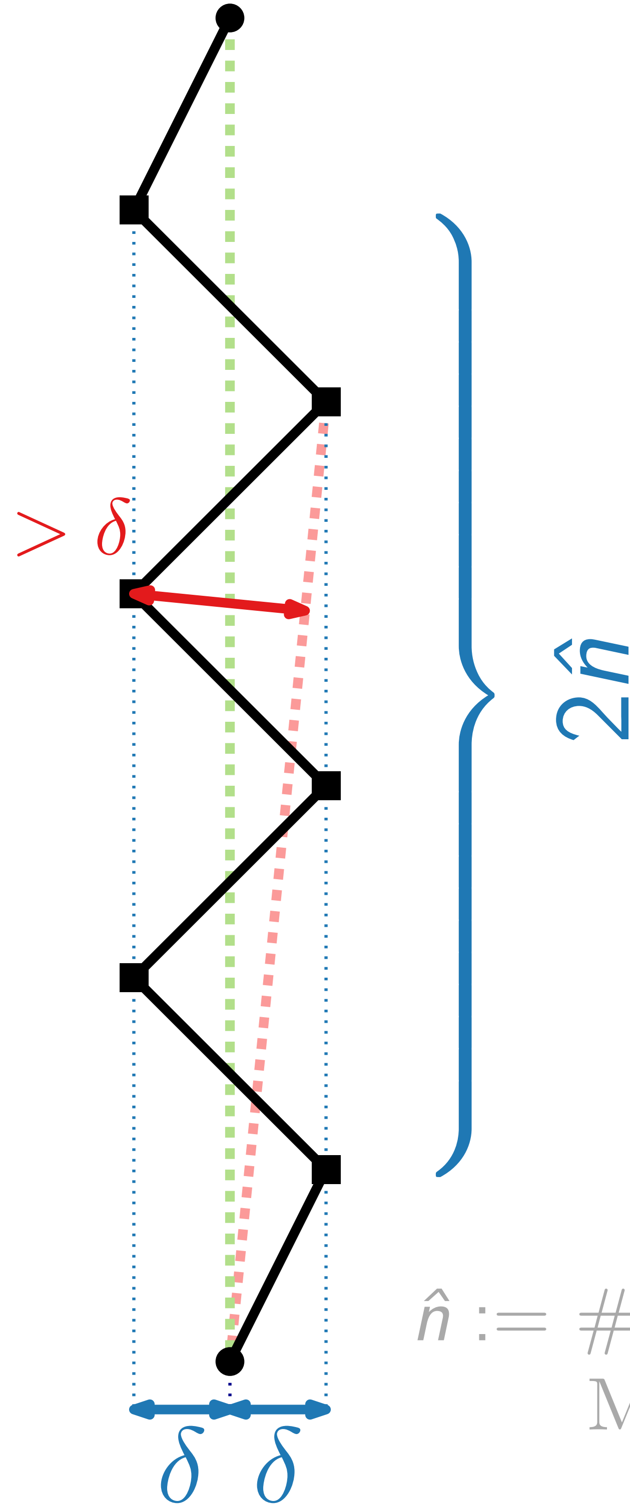
$\hat{n}$:= # vertices in
MIDS graph

Vertex Gadget



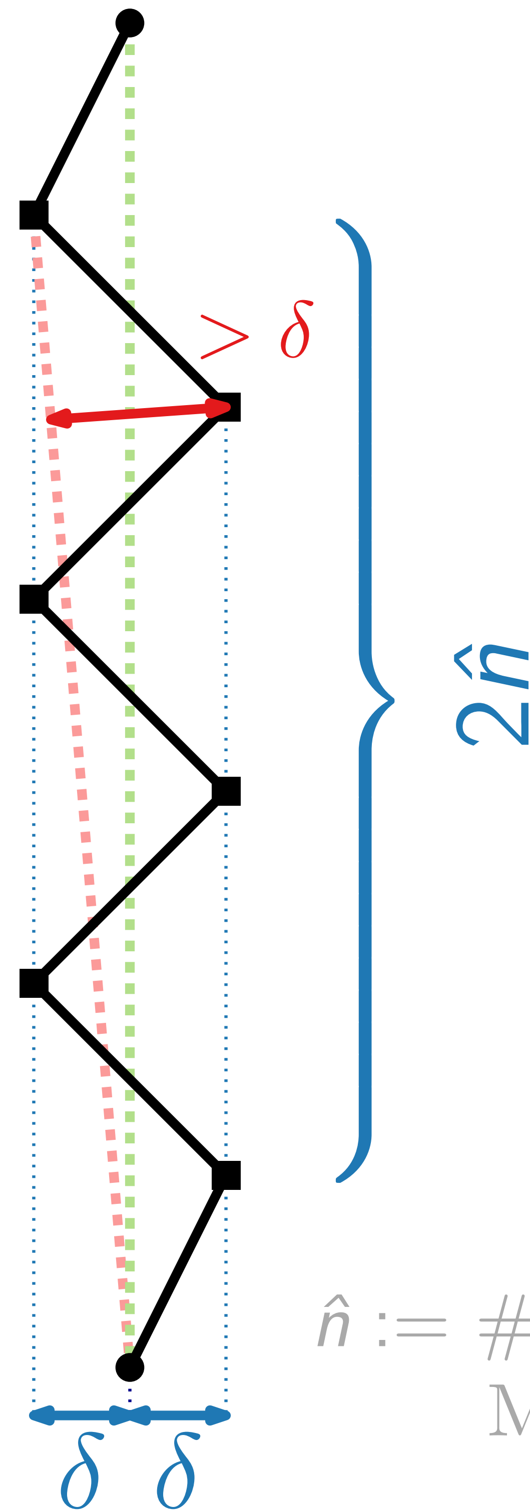
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget



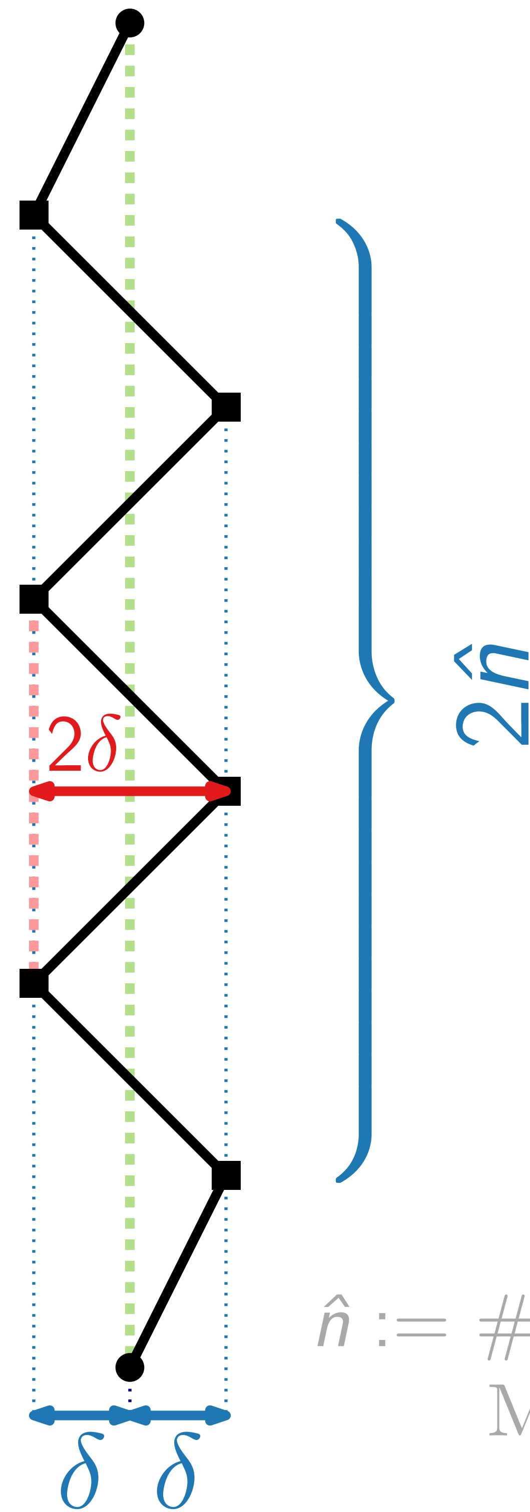
$\hat{n}$:= # vertices in MIDS graph

Vertex Gadget



$\hat{n} := \#$ vertices in MIDS graph

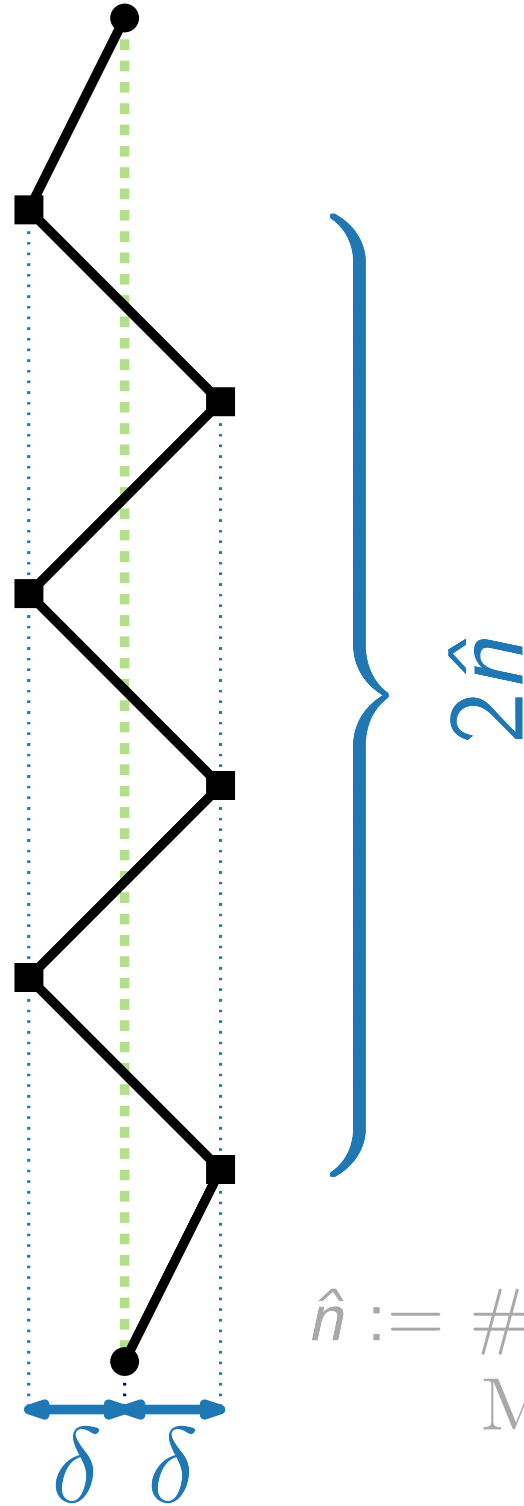
Vertex Gadget



$\hat{n}$:= # vertices in MIDS graph

Vertex Gadget

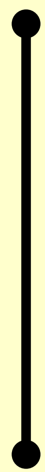
Interpretation:



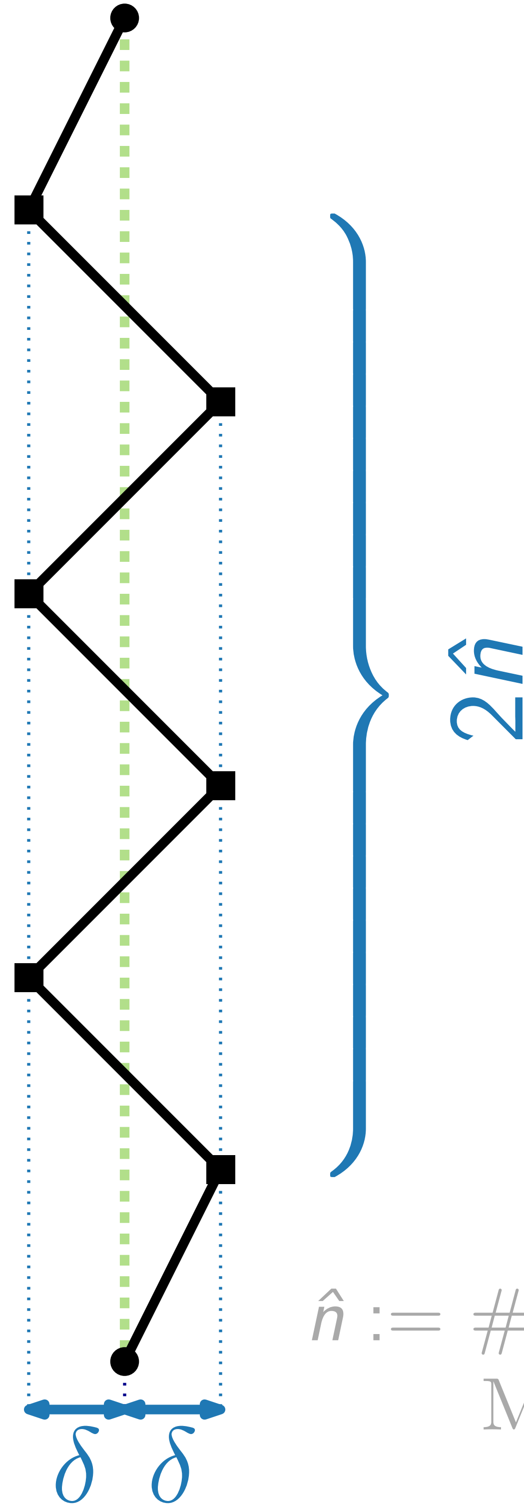
$\hat{n} := \#$ vertices in MIDS graph

Vertex Gadget

Interpretation:



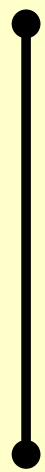
vertex is **not** in the MIDS



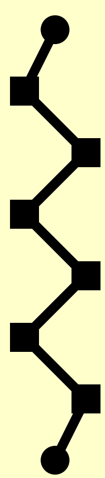
$\hat{n}$:= # vertices in MIDS graph

Vertex Gadget

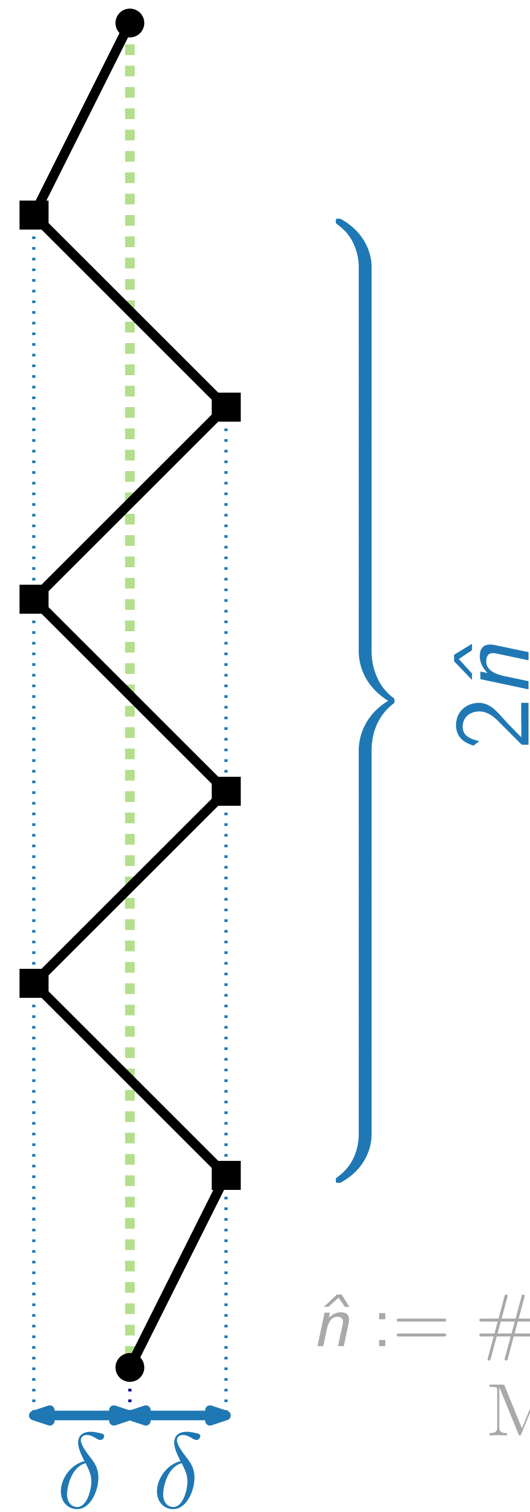
Interpretation:



vertex is **not** in the MIDS

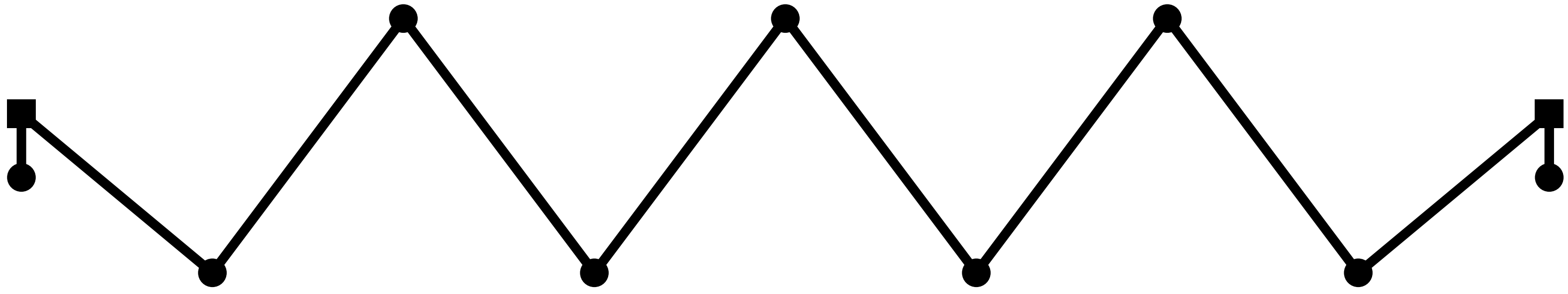


vertex is in the MIDS

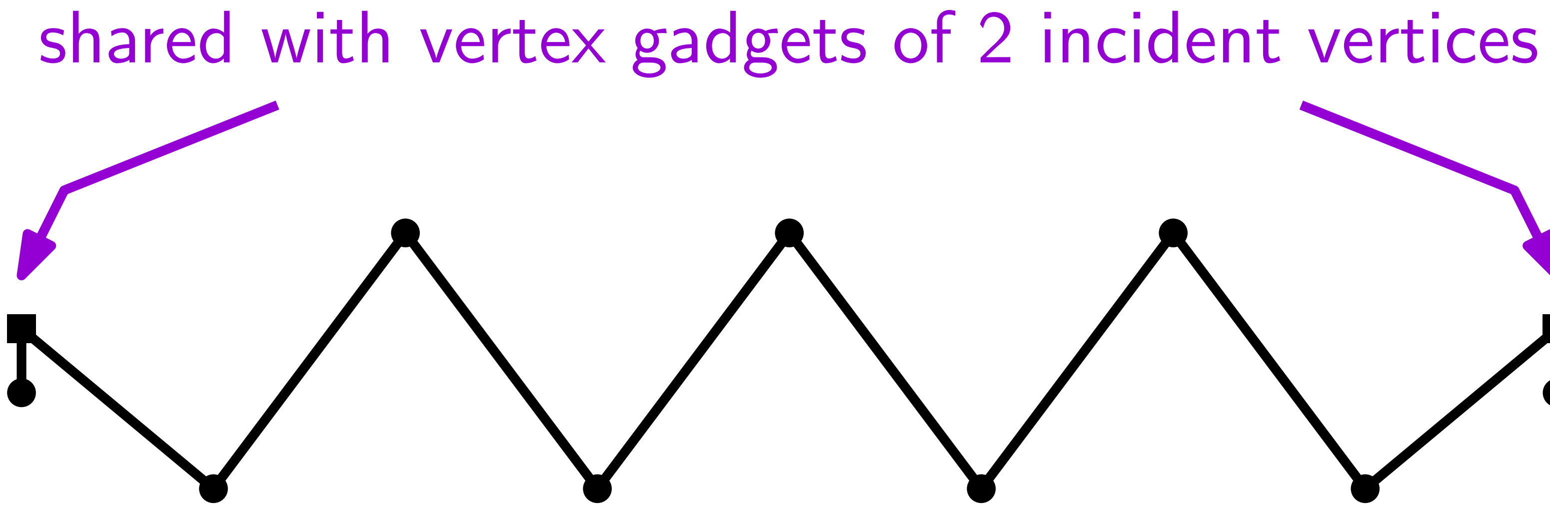


$\hat{n}$:= # vertices in MIDS graph

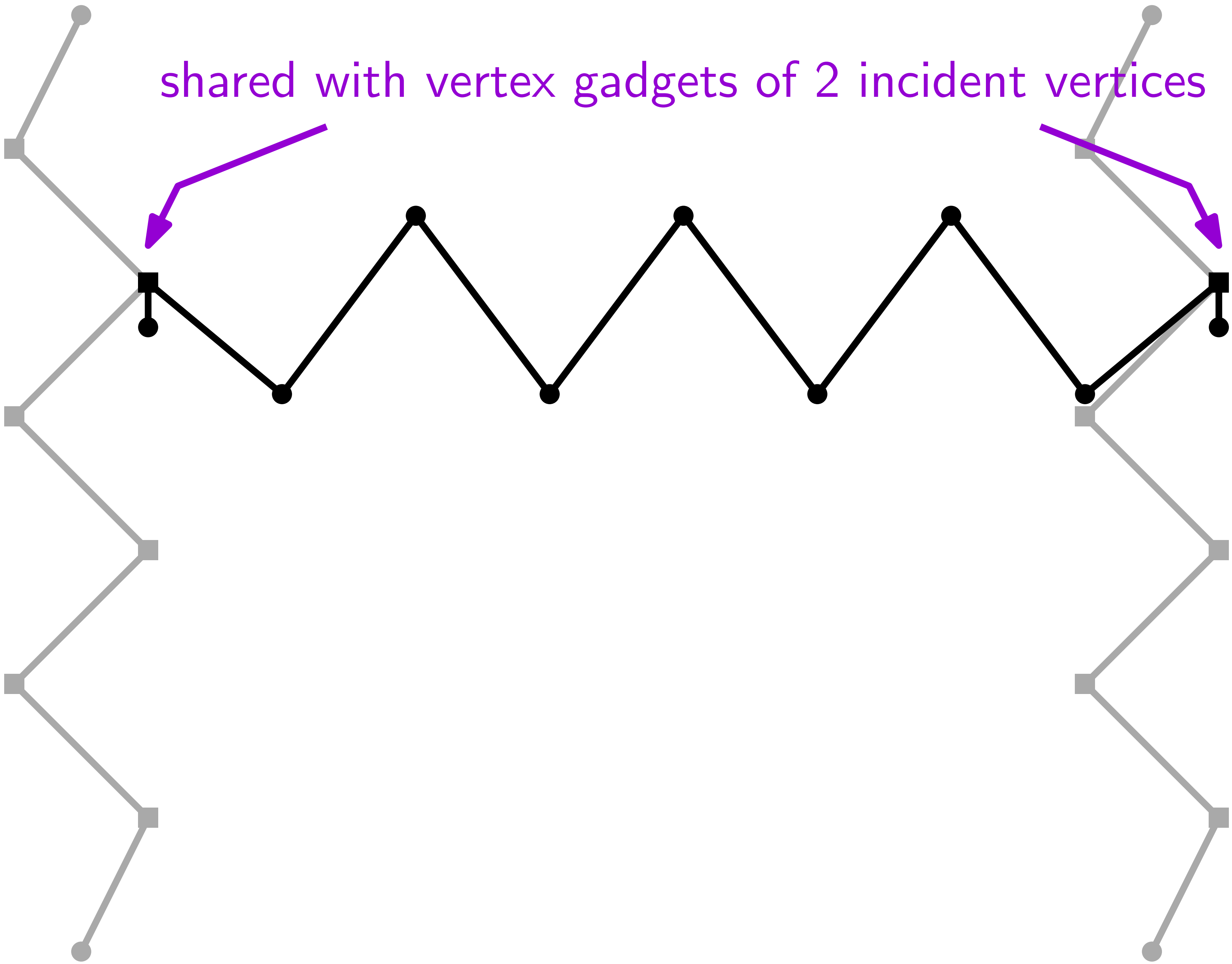
Edge Gadget



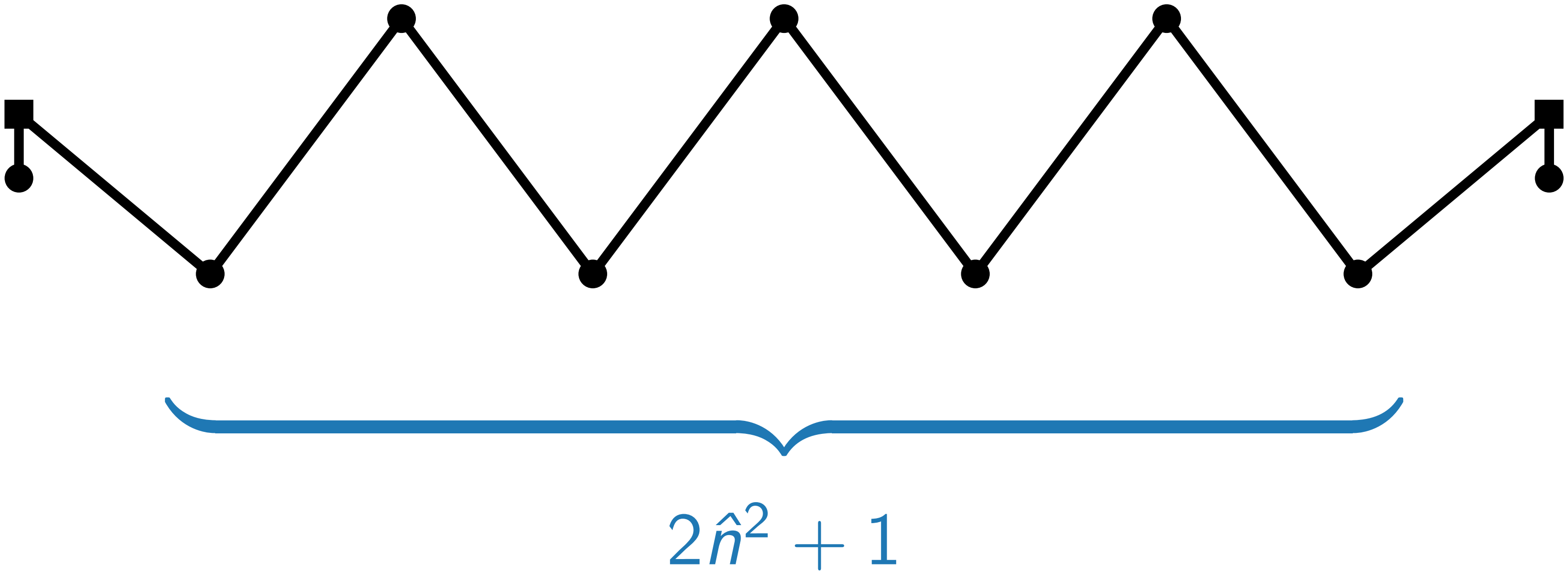
Edge Gadget



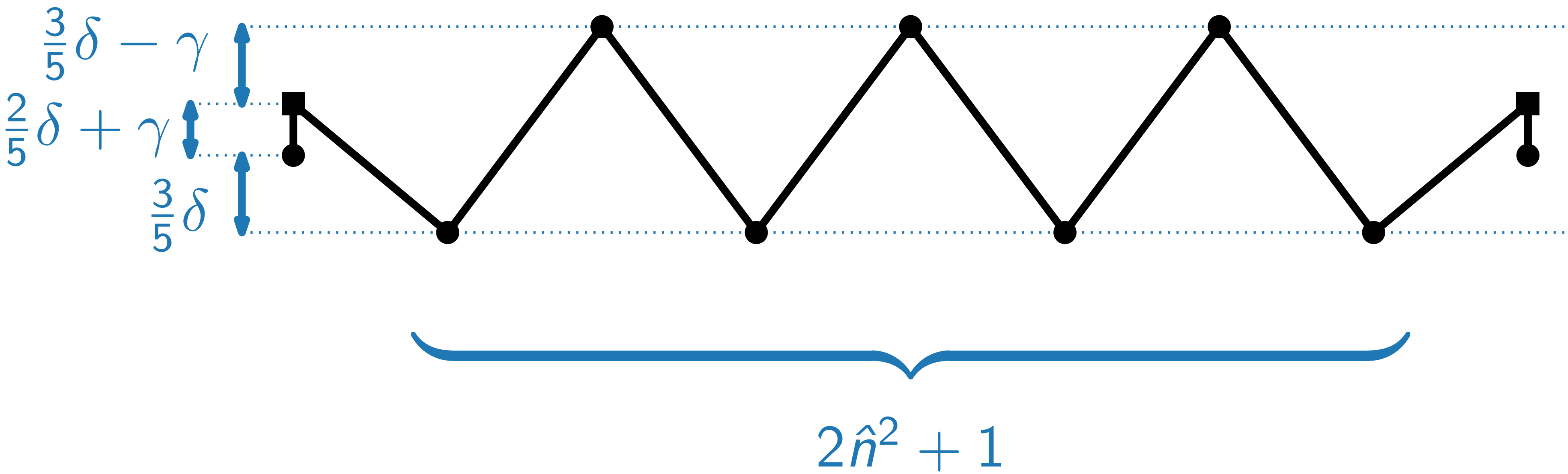
Edge Gadget



Edge Gadget

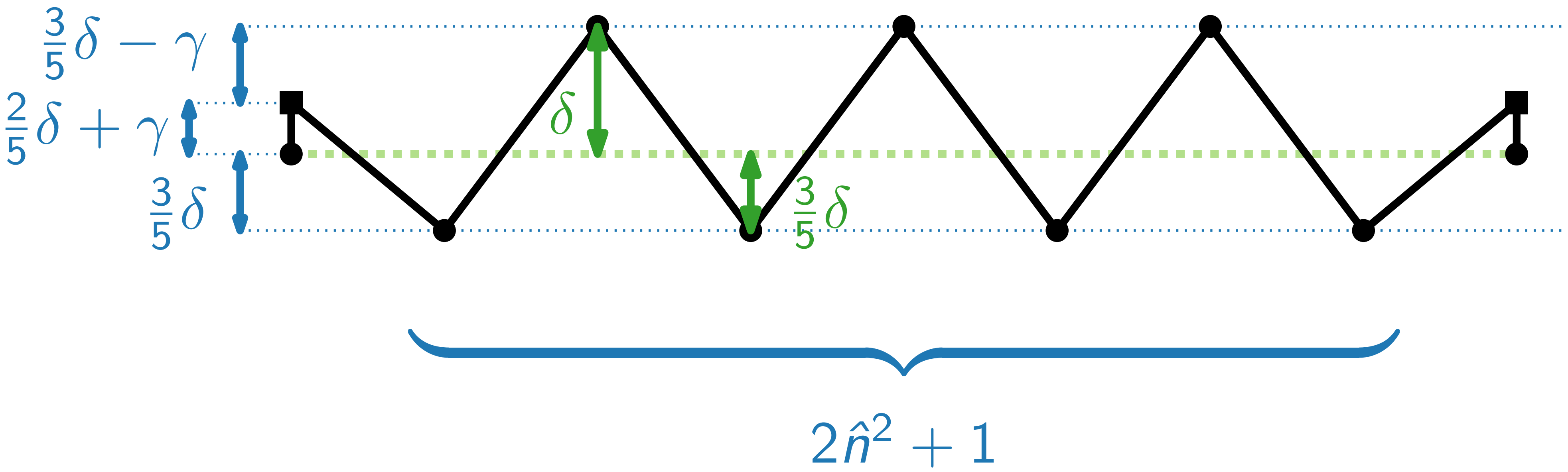


Edge Gadget



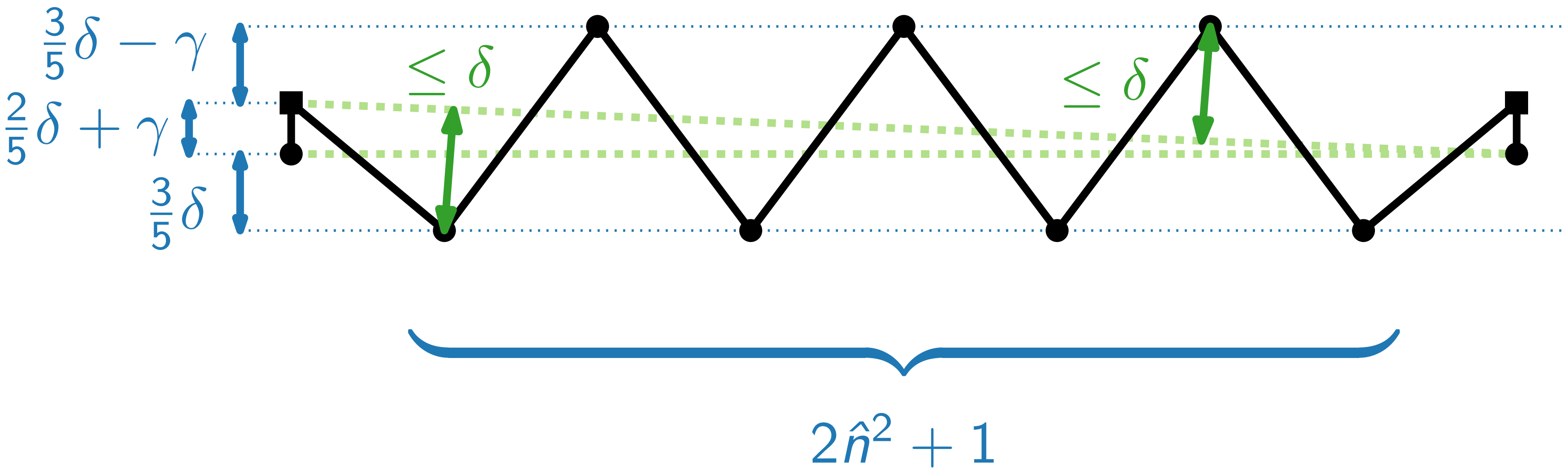
$\frac{3}{5}\delta - \gamma$
$\frac{2}{5}\delta + \gamma$
$\frac{3}{5}\delta$
$2\hat{n}^2 + 1$

Edge Gadget



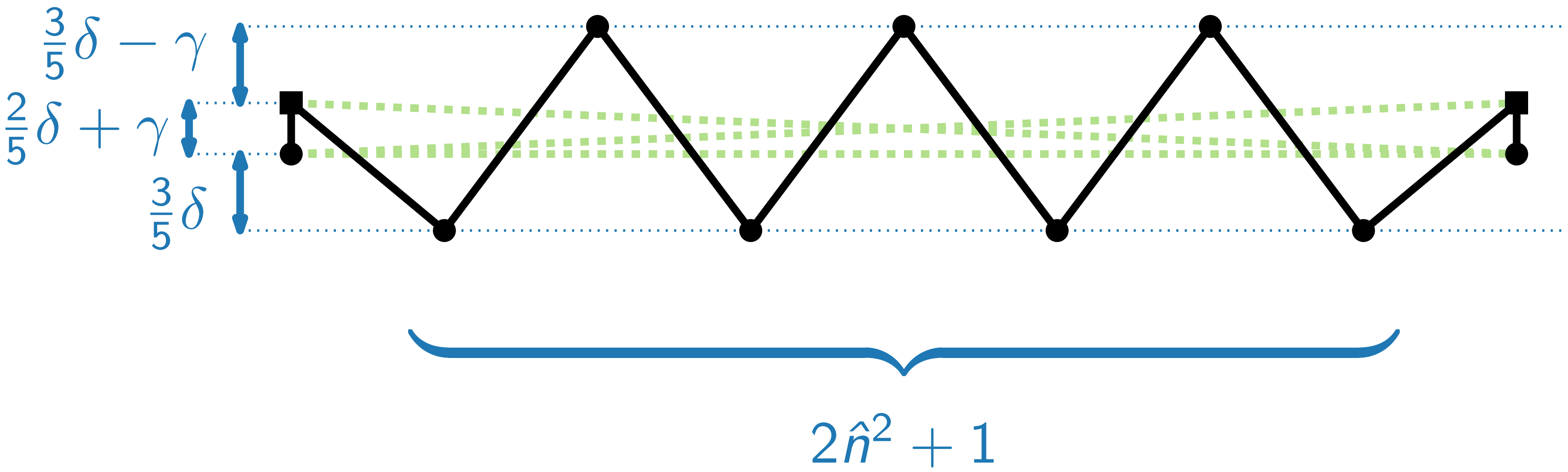
$\frac{3}{5}\delta - \gamma$
$\frac{2}{5}\delta + \gamma$
$\frac{3}{5}\delta$
δ
$\frac{3}{5}\delta$
$2\hat{n}^2 + 1$

Edge Gadget



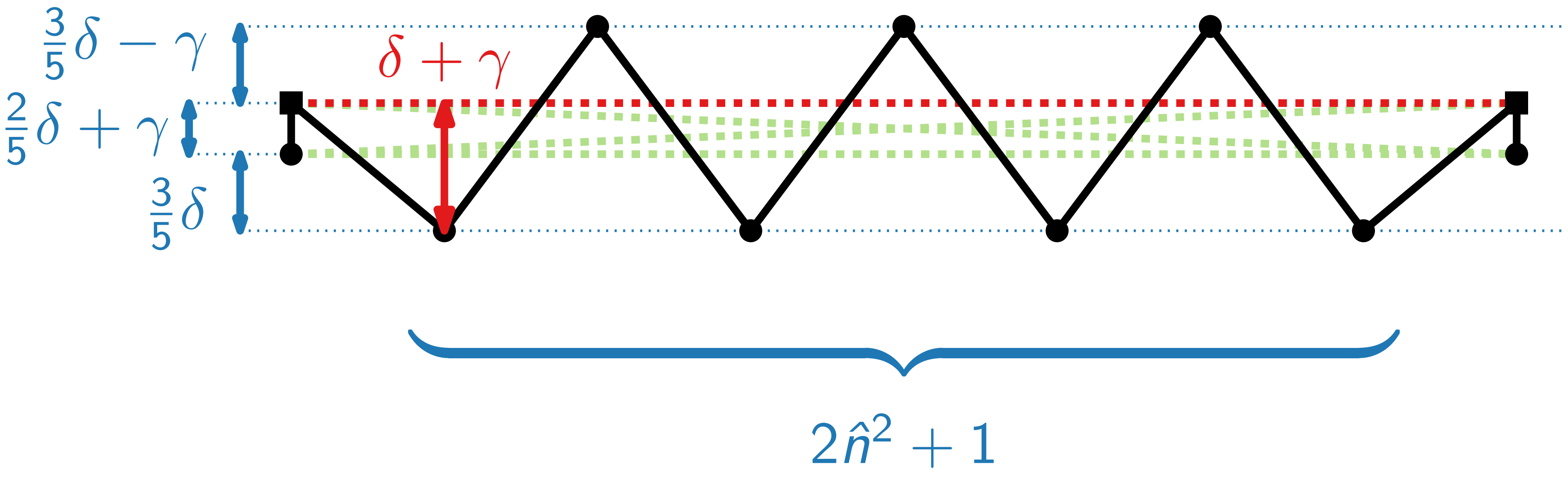
$\frac{3}{5}\delta - \gamma$
$\frac{2}{5}\delta + \gamma$
$\frac{3}{5}\delta$
$\leq \delta$
$\leq \delta$
$2\hat{n}^2 + 1$

Edge Gadget



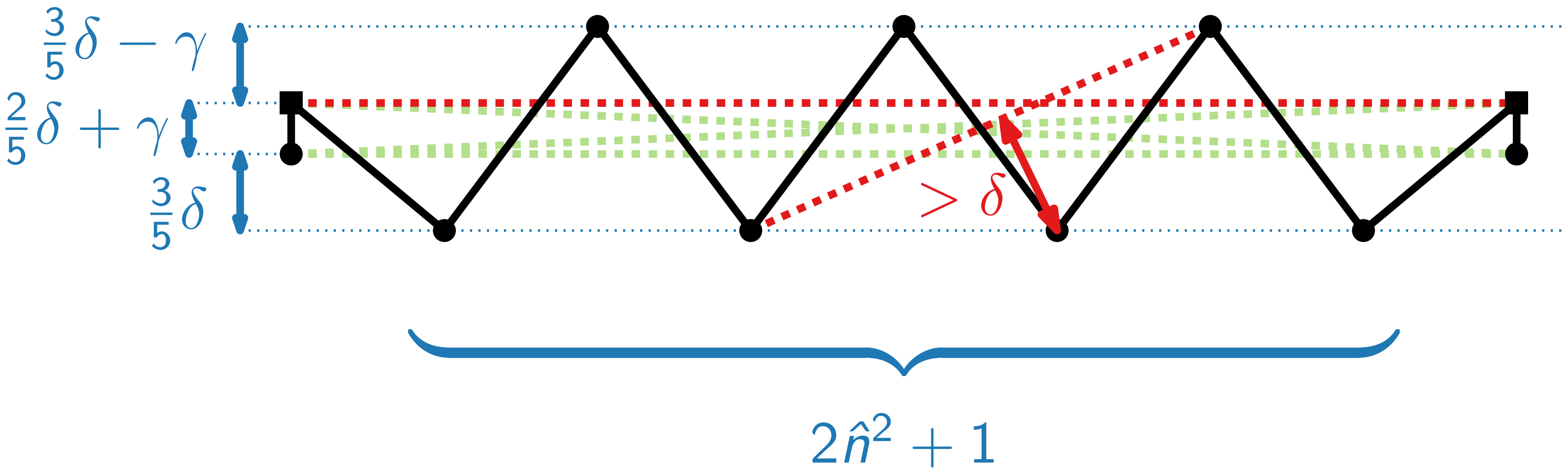
$\frac{3}{5}\delta - \gamma$
$\frac{2}{5}\delta + \gamma$
$\frac{3}{5}\delta$
$2\hat{n}^2 + 1$

Edge Gadget



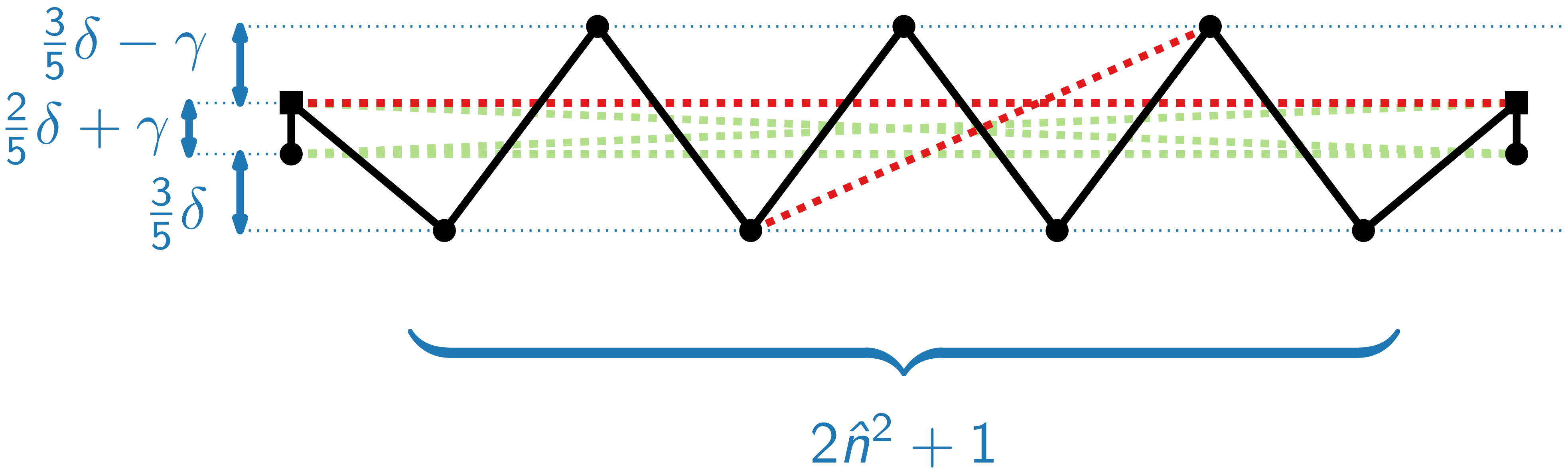
$\frac{3}{5}\delta - \gamma$
$\frac{2}{5}\delta + \gamma$
$\frac{3}{5}\delta$
$\delta + \gamma$
$2\hat{n}^2 + 1$

Edge Gadget



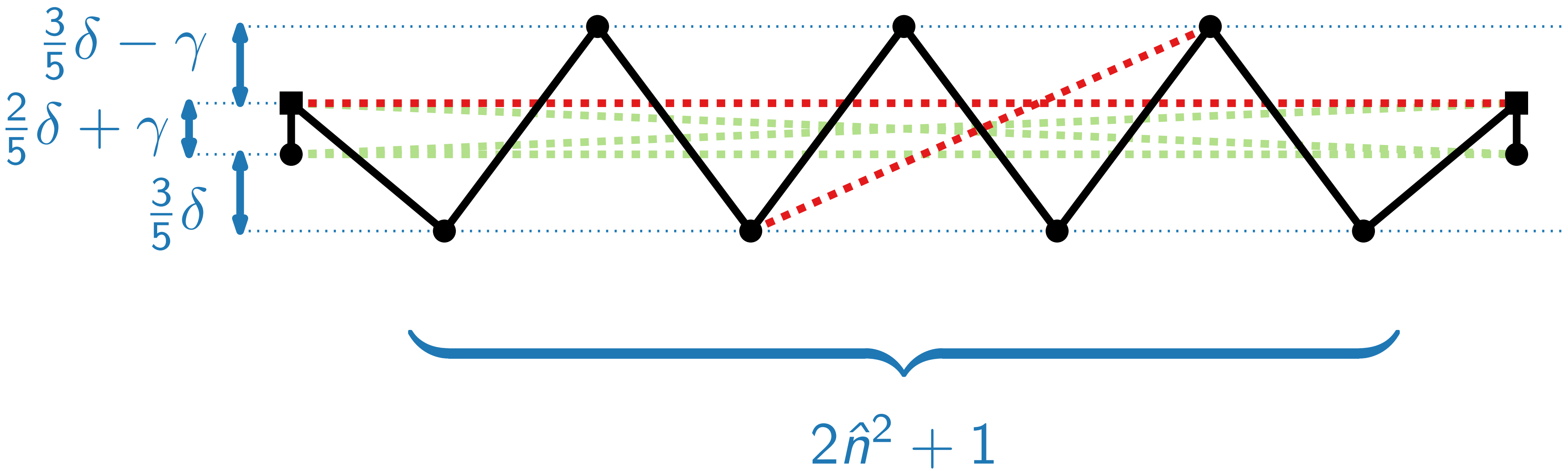
$\frac{3}{5}\delta - \gamma$
$\frac{2}{5}\delta + \gamma$
$\frac{3}{5}\delta$
$> \delta$
$2\hat{n}^2 + 1$

Edge Gadget



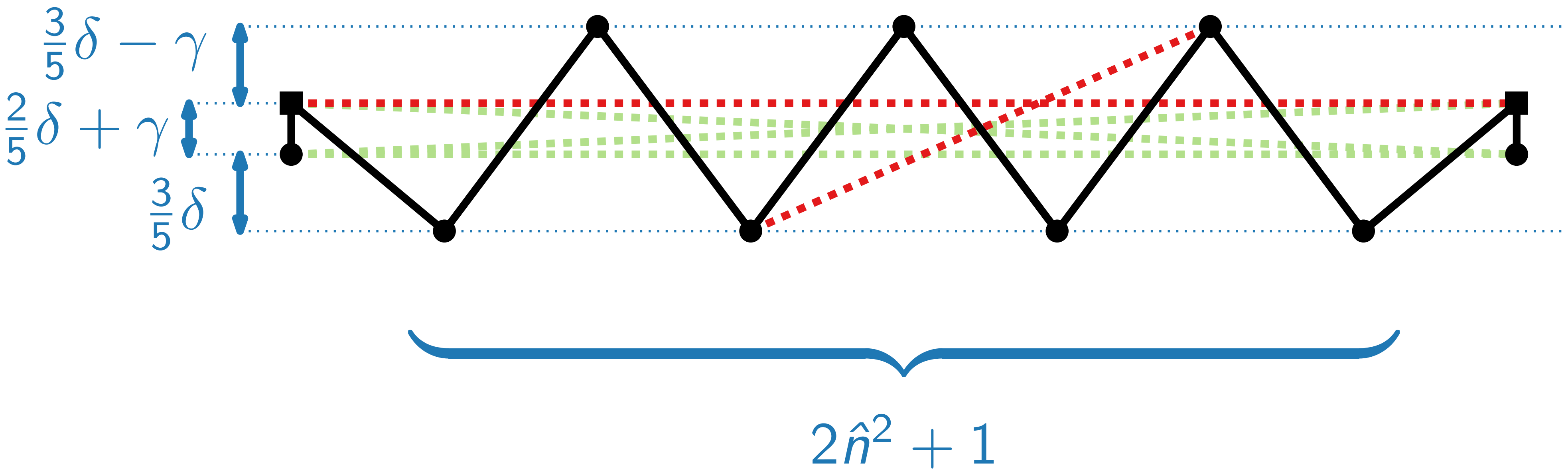
- not taking a "long" shortcut is very expensive

Edge Gadget



- not taking a “long” shortcut is very expensive
- take a “long” shortcut $\Rightarrow$ take a shortcut of a vertex gadget

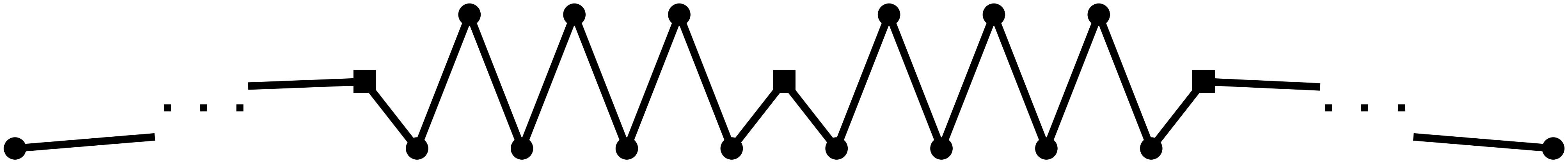
Edge Gadget



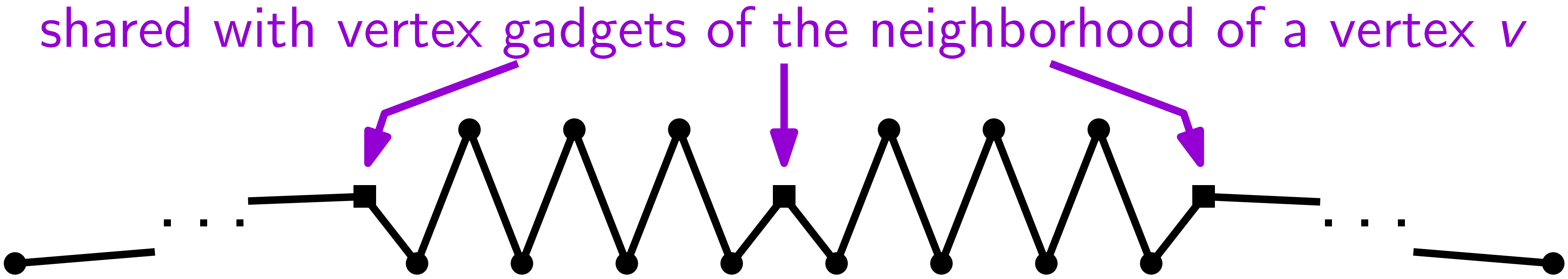
- not taking a “long” shortcut is very expensive
- take a “long” shortcut $\Rightarrow$ take a shortcut of a vertex gadget

$\Rightarrow$ guarantees independent set property

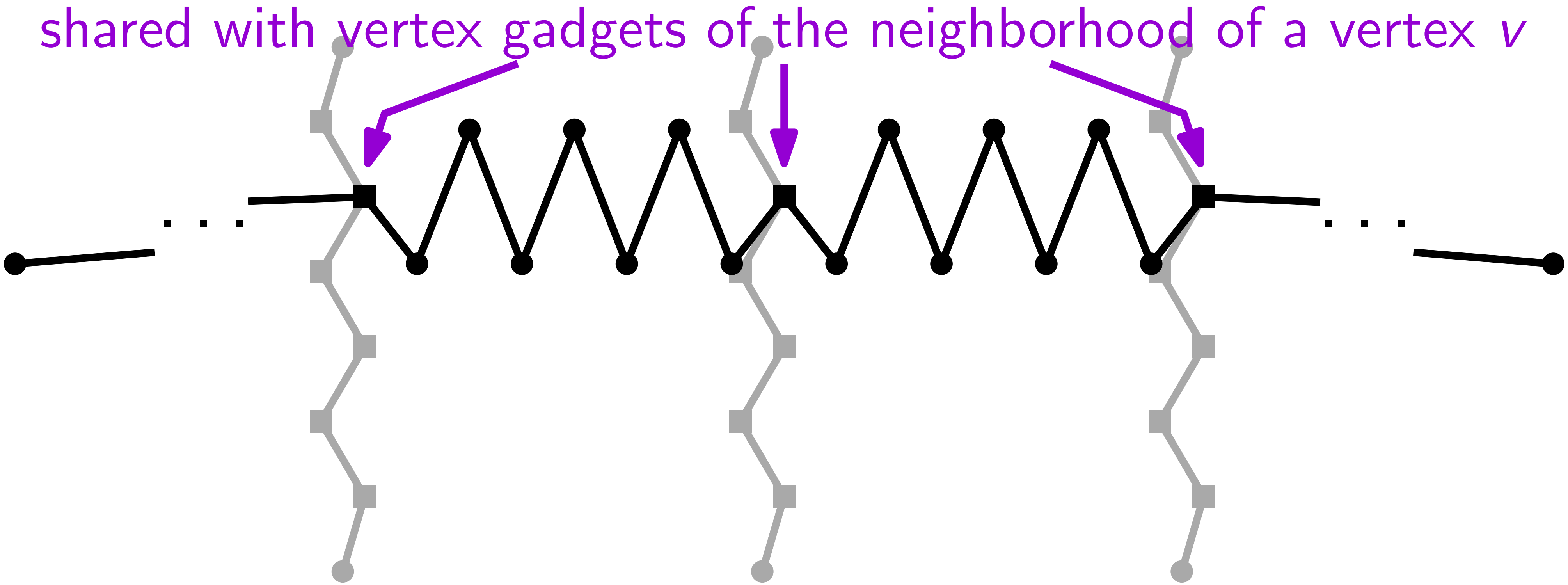
Neighborhood Gadget



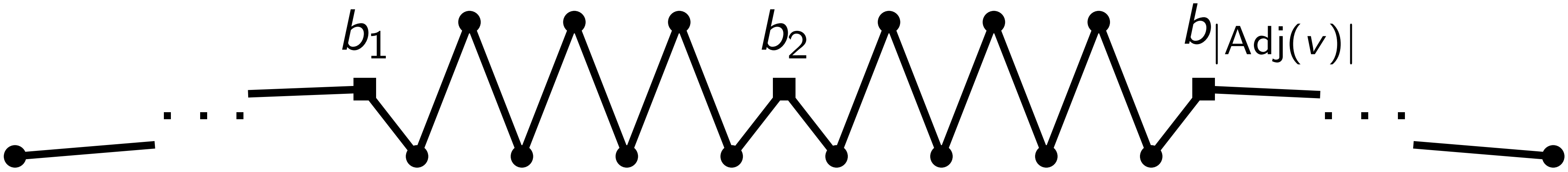
Neighborhood Gadget



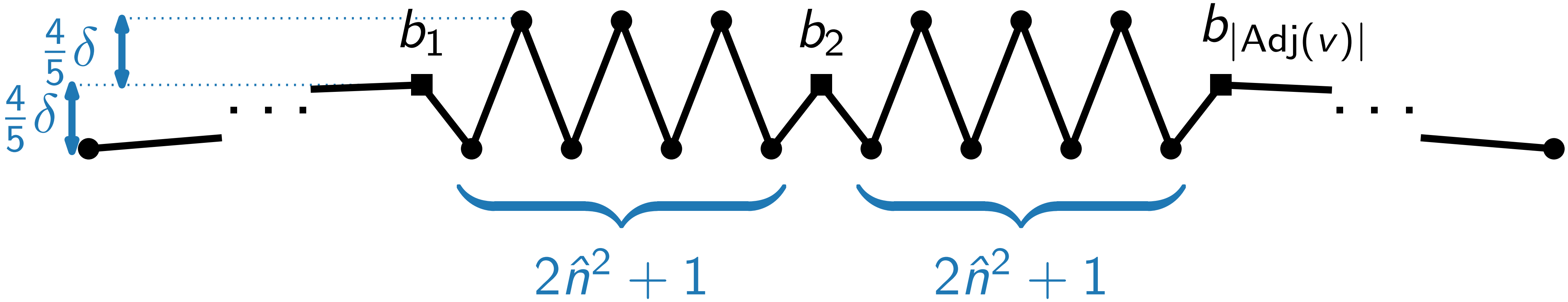
Neighborhood Gadget



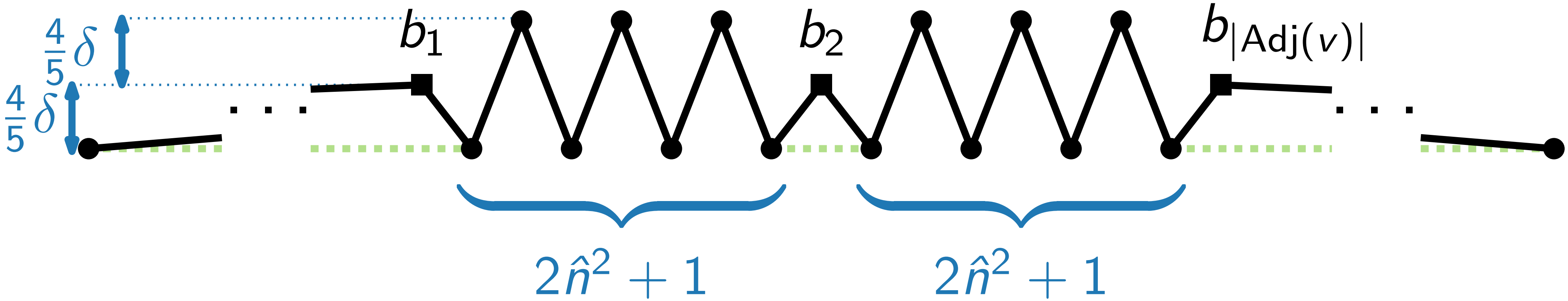
Neighborhood Gadget



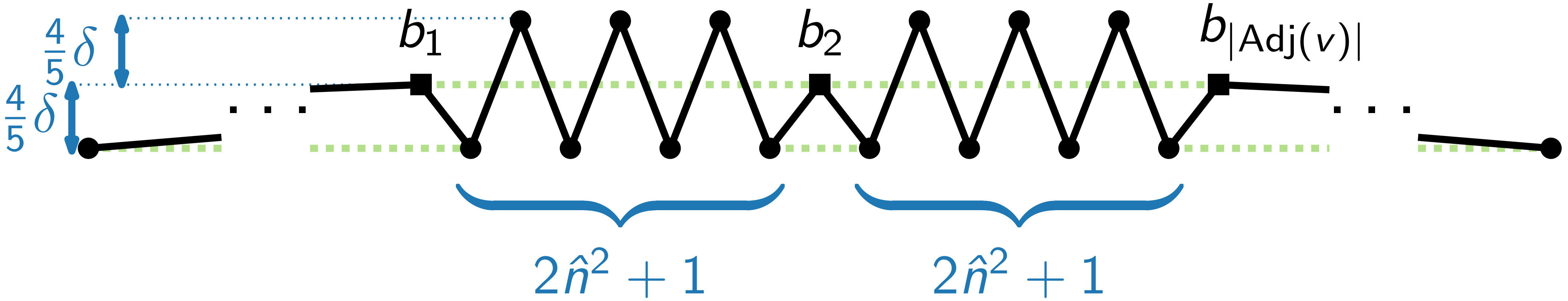
Neighborhood Gadget



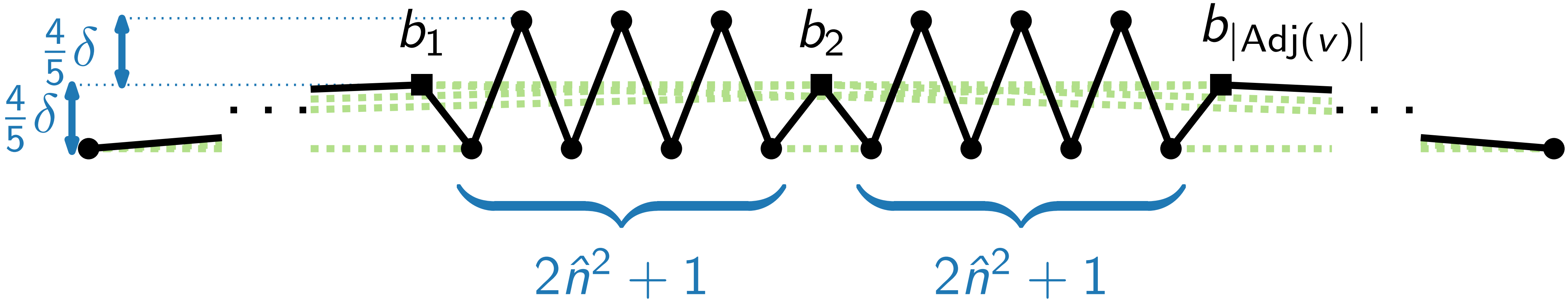
Neighborhood Gadget



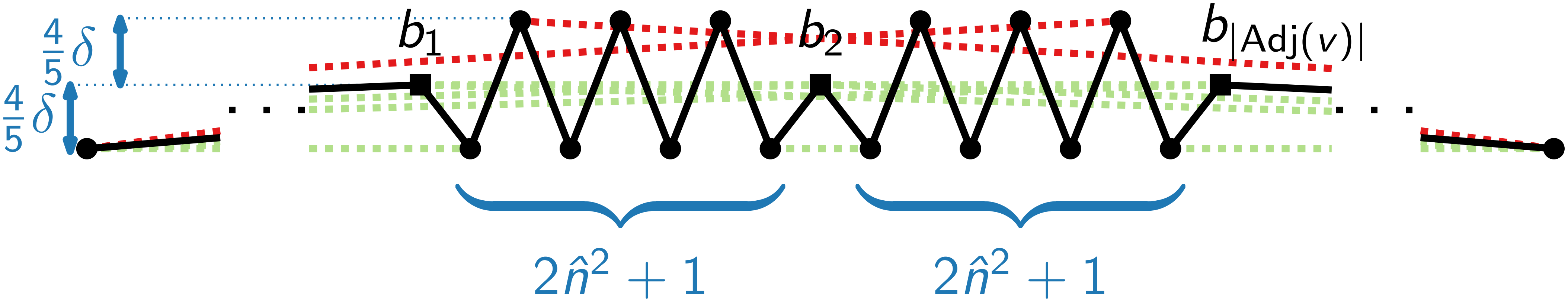
Neighborhood Gadget



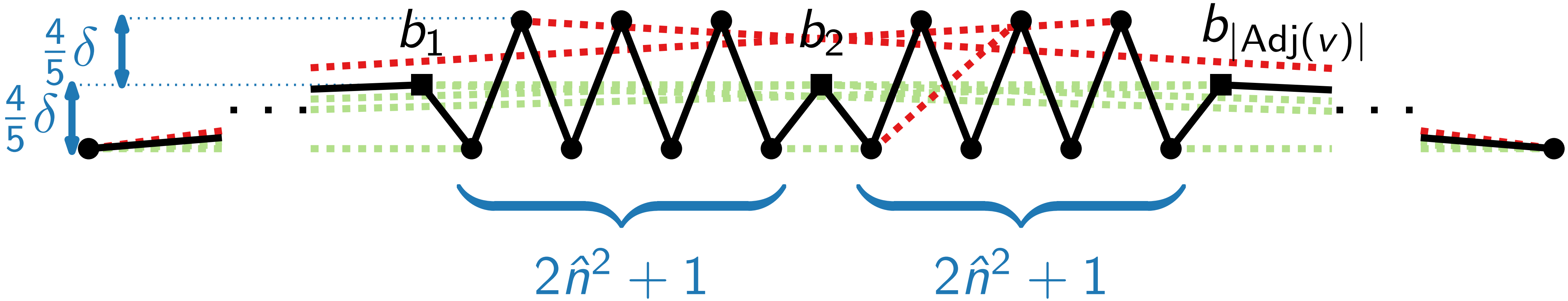
Neighborhood Gadget



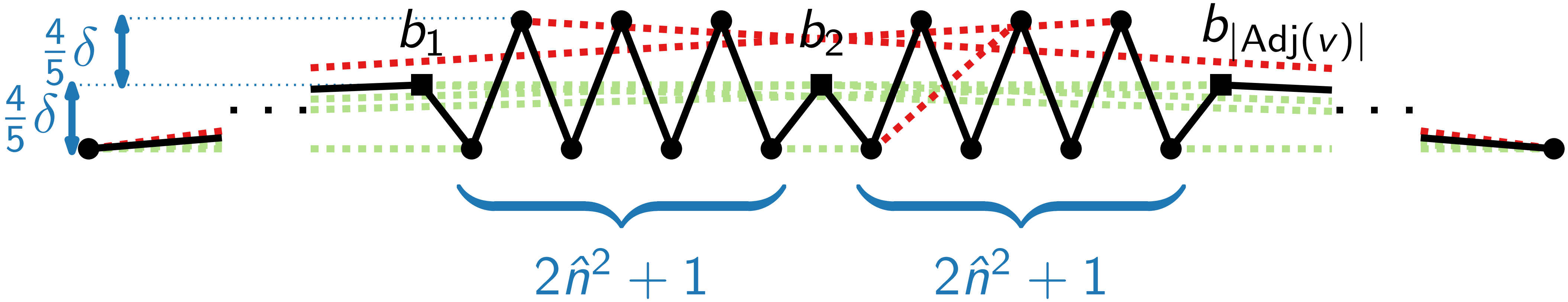
Neighborhood Gadget



Neighborhood Gadget

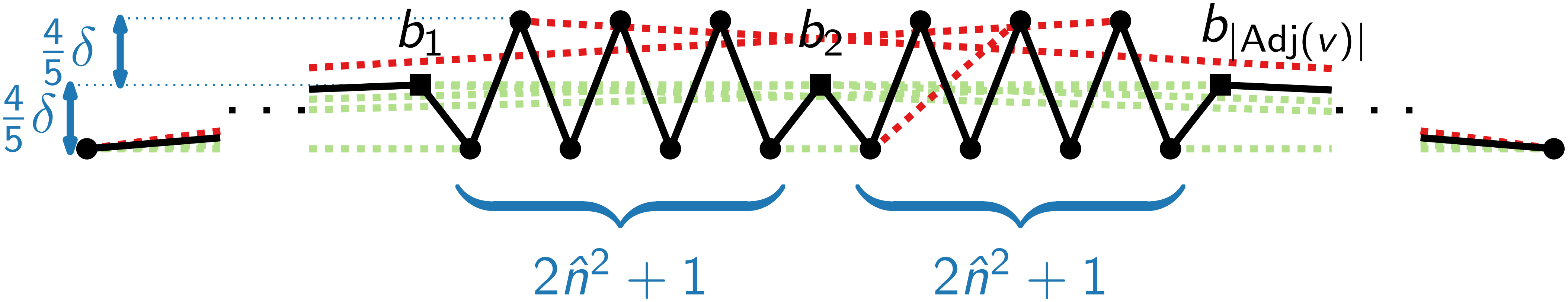


Neighborhood Gadget



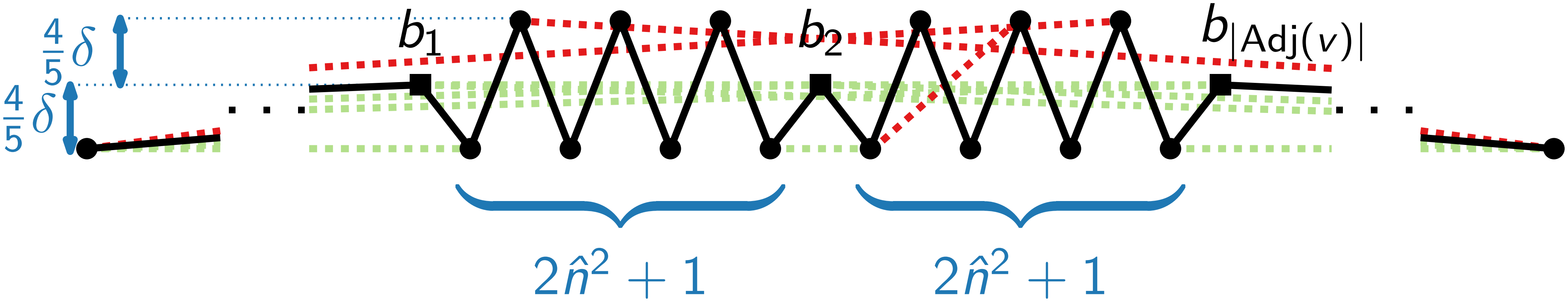
- not taking the “long” shortcuts is very expensive

Neighborhood Gadget



- not taking the “long” shortcuts is very expensive
- take a “long” shortcut $\Rightarrow$ do not take a shortcut of at least one vertex gadget

Neighborhood Gadget

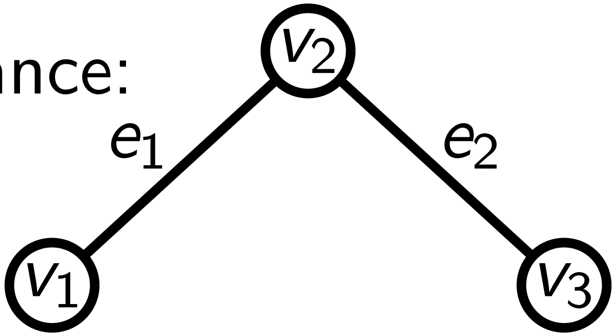


- not taking the "long" shortcuts is very expensive
- take a "long" shortcut $\Rightarrow$ do not take a shortcut of at least one vertex gadget

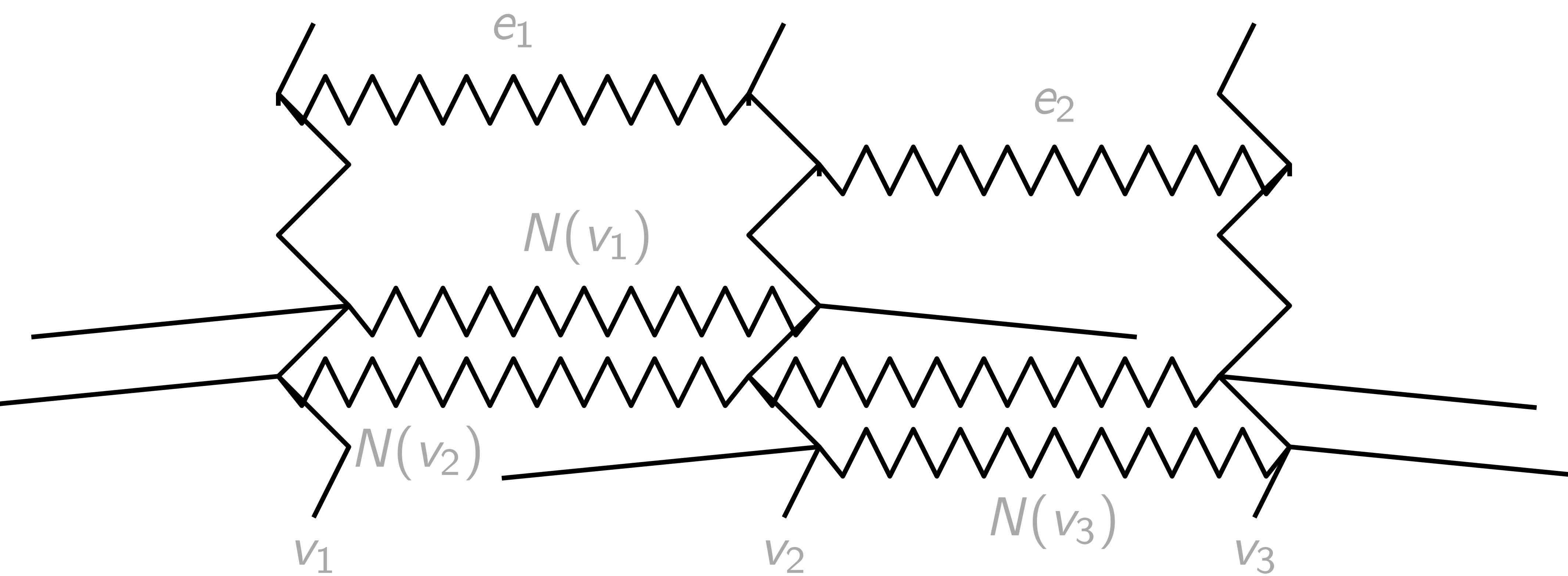
$\Rightarrow$ guarantees dominating set property

Full Example

MIDS instance:

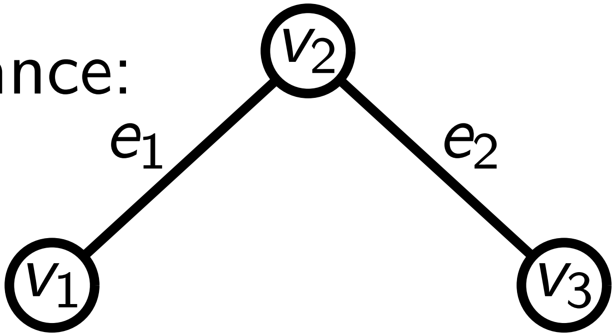


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

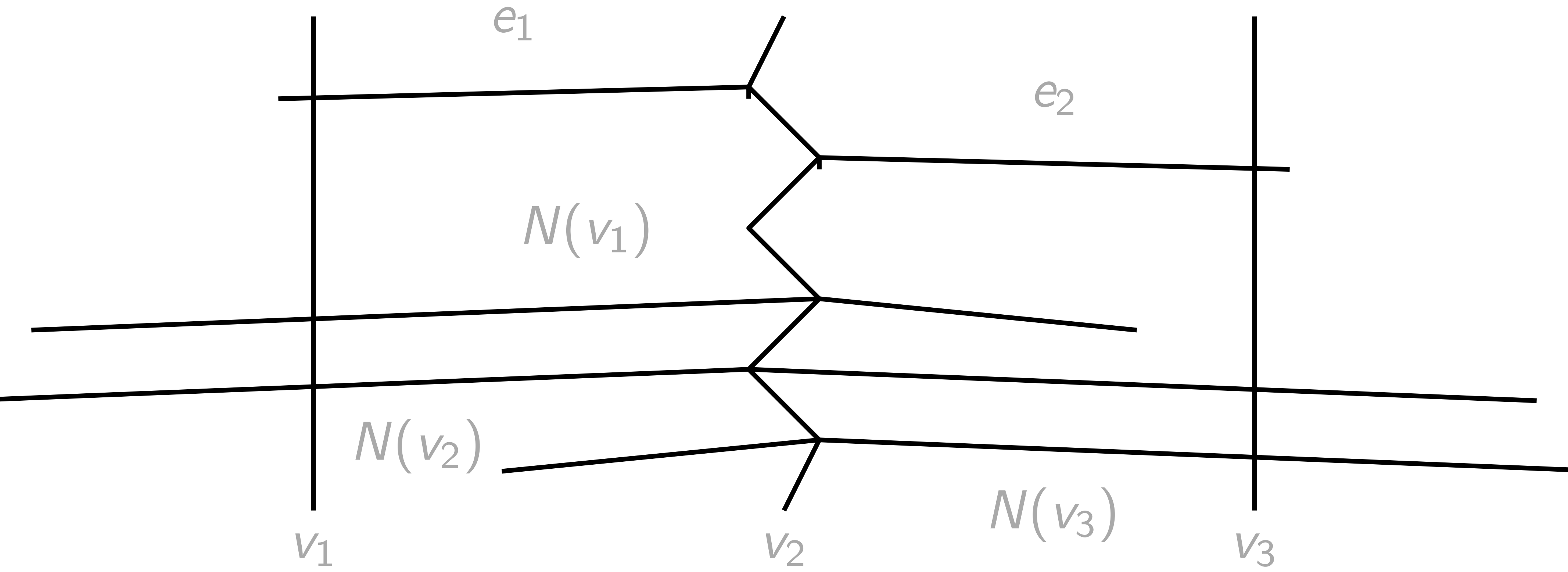


Full Example

MIDS instance:

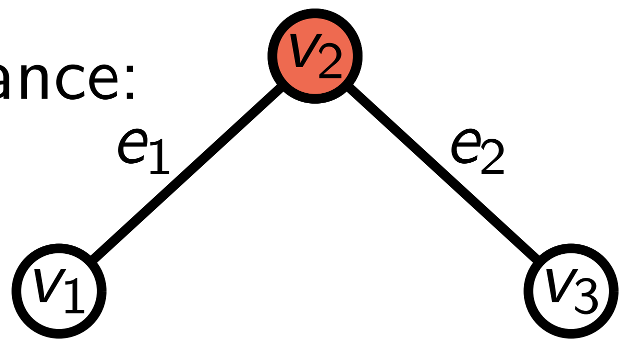


$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$

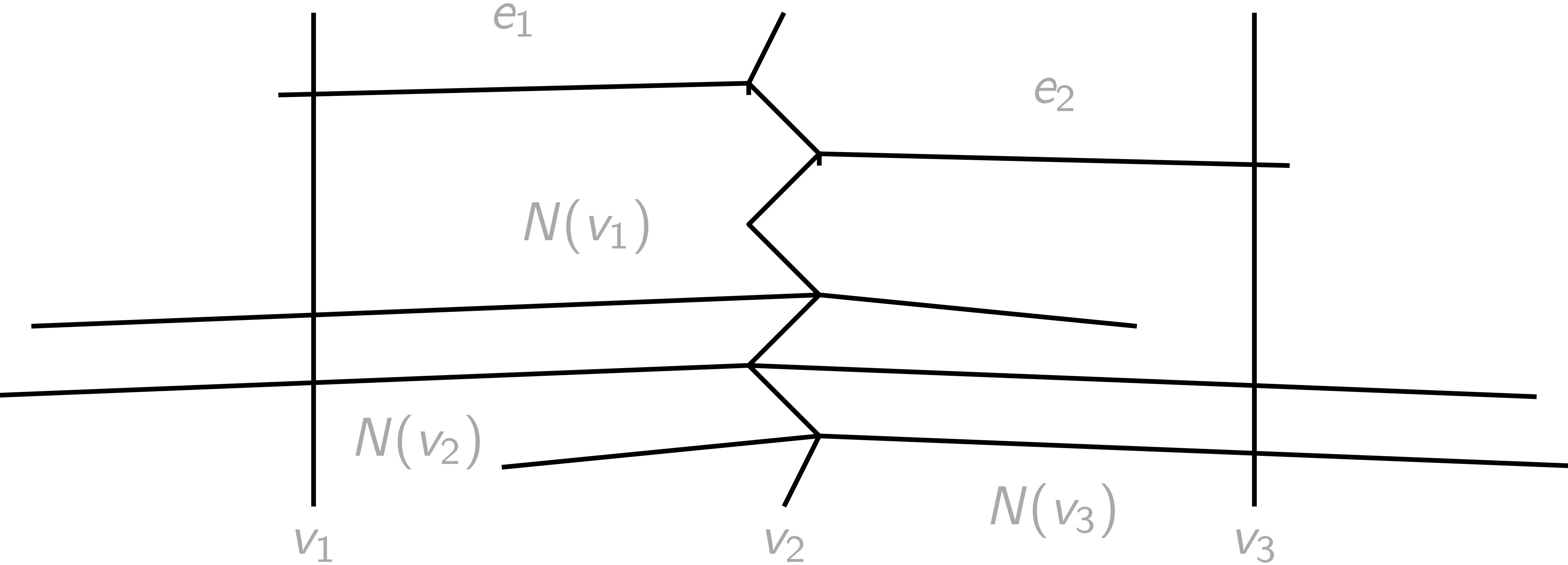


Full Example

MIDS instance:



$$\hat{n} = 3$$
$$2\hat{n} = 6$$
$$2\hat{n}^2 + 1 = 19$$



Agenda

1. Motivation and Introduction
2. Problem Definition
3. Hardness of Approximation (+ Proof Sketch)
4. **Bi-Criteria Approximation (+ Proof Sketch)**
5. Summary

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards “bad” choices for keeping or discarding a bend

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards “bad” choices for keeping or discarding a bend

  How to overcome this?

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards “bad” choices for keeping or discarding a bend

How to overcome this?

Relax distance constraint!

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards “bad” choices for keeping or discarding a bend

How to overcome this?

Relax distance constraint!

- A bi-criteria (α, β) approx. algo. yields a polyline bundle

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards "bad" choices for keeping or discarding a bend

How to overcome this?

Relax distance constraint!

- A bi-criteria (α, β) approx. algo. yields a polyline bundle
  - for distance threshold $\beta \cdot \delta$

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards “bad” choices for keeping or discarding a bend

How to overcome this?

Relax distance constraint!

- A bi-criteria (α, β) approx. algo. yields a polyline bundle
  - for distance threshold $\beta \cdot \delta$
  - with at most α times as many bends as an optimal solution for distance threshold δ, and

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards "bad" choices for keeping or discarding a bend

How to overcome this?

Relax distance constraint!

- A bi-criteria (α, β) approx. algo. yields a polyline bundle
  - for distance threshold $\beta \cdot \delta$
  - with at most α times as many bends as an optimal solution for distance threshold δ, and
  - runs in polynomial time.

Bi-criteria Approximation

- Reason for this strong inapproximability: high sensitivity towards “bad” choices for keeping or discarding a bend

How to overcome this?

Relax distance constraint!

- A bi-criteria (α, β) approx. algo. yields a polyline bundle
  - for distance threshold $\beta \cdot \delta$
  - with at most α times as many bends as an optimal solution for distance threshold δ, and
  - runs in polynomial time.

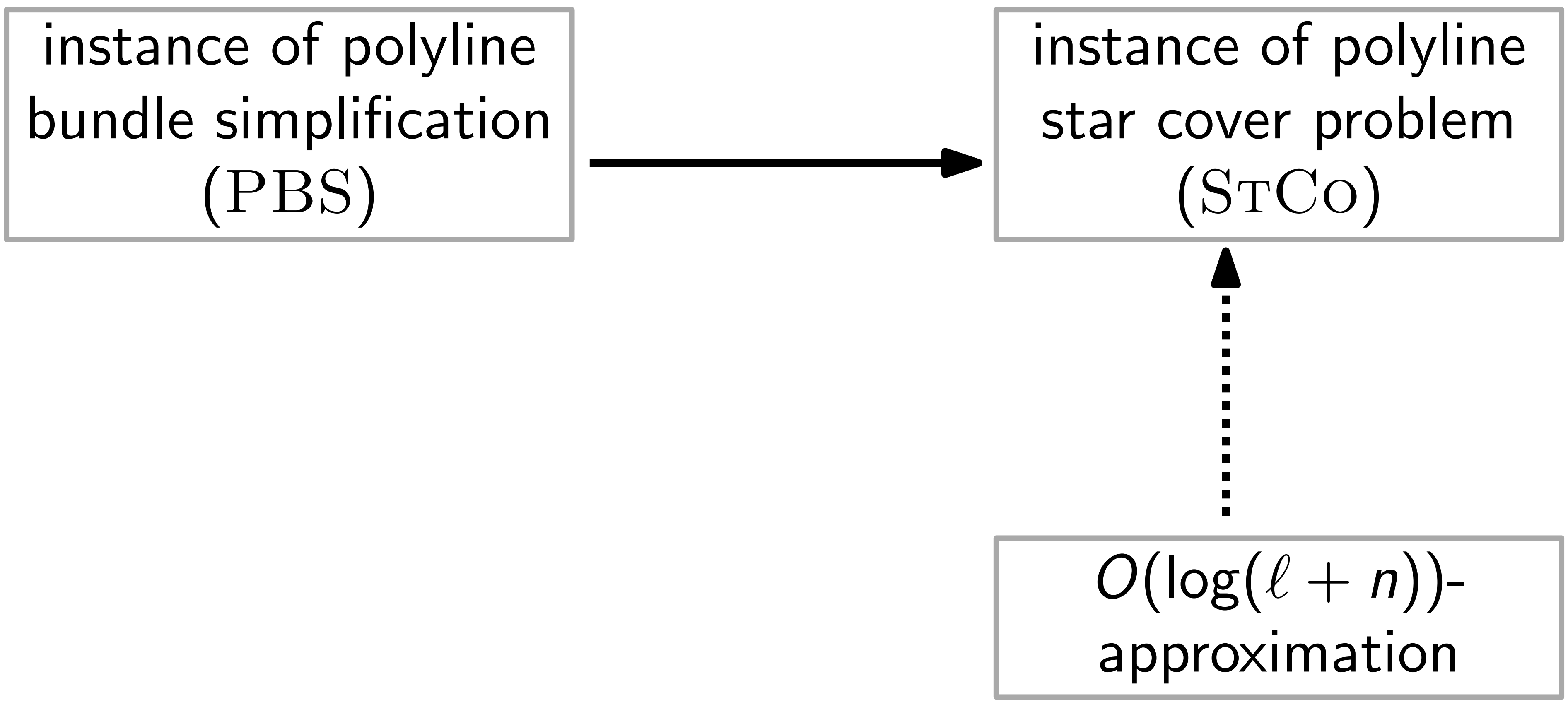
Theorem 2:
There is a bi-criteria $(O(\log(\ell + n)), 2)$-approximation algorithm for simplifying a bundle of polylines.
$\ell = \#$ polylines, $n = \#$ bends

Bi-criteria Approximation

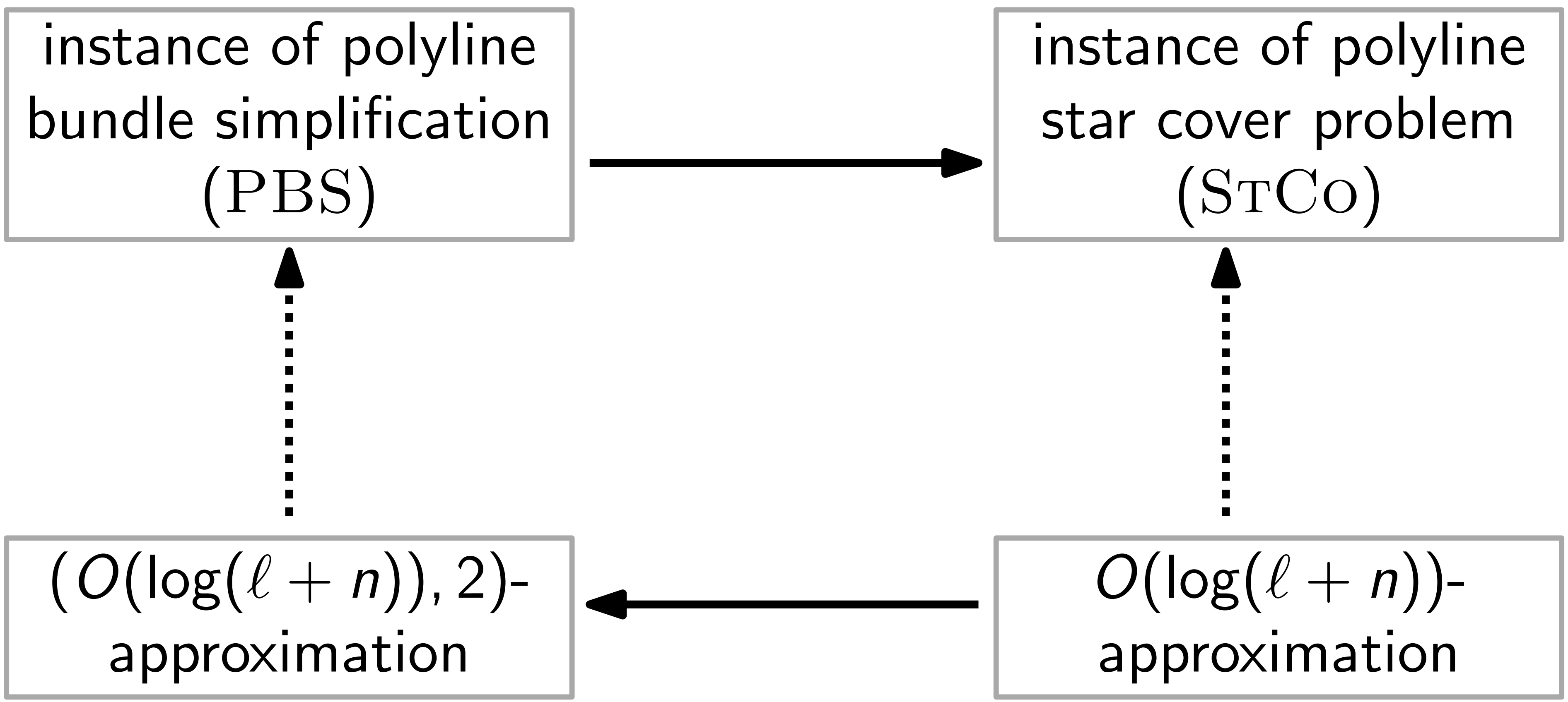
instance of polyline bundle simplification (PBS)

Bi-criteria Approximation

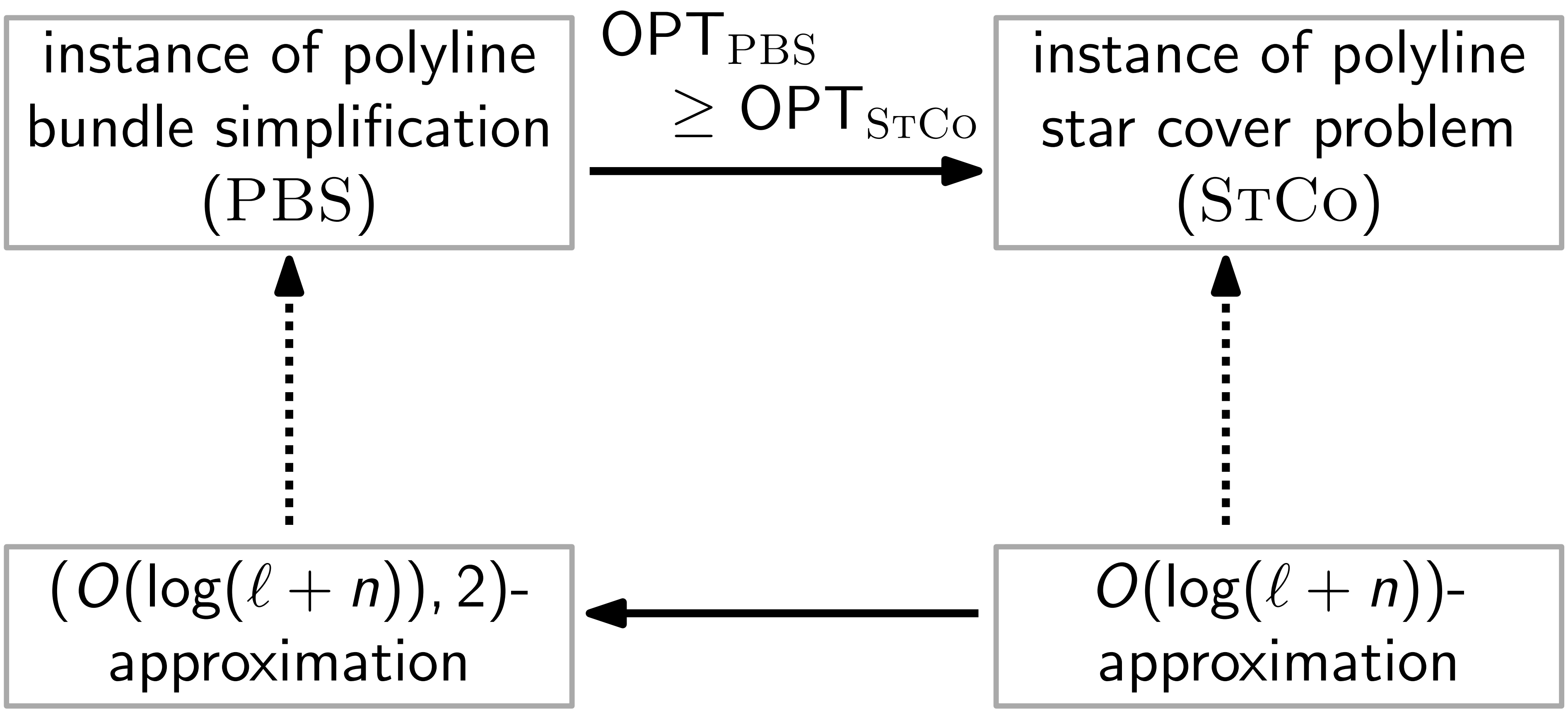
Bi-criteria Approximation



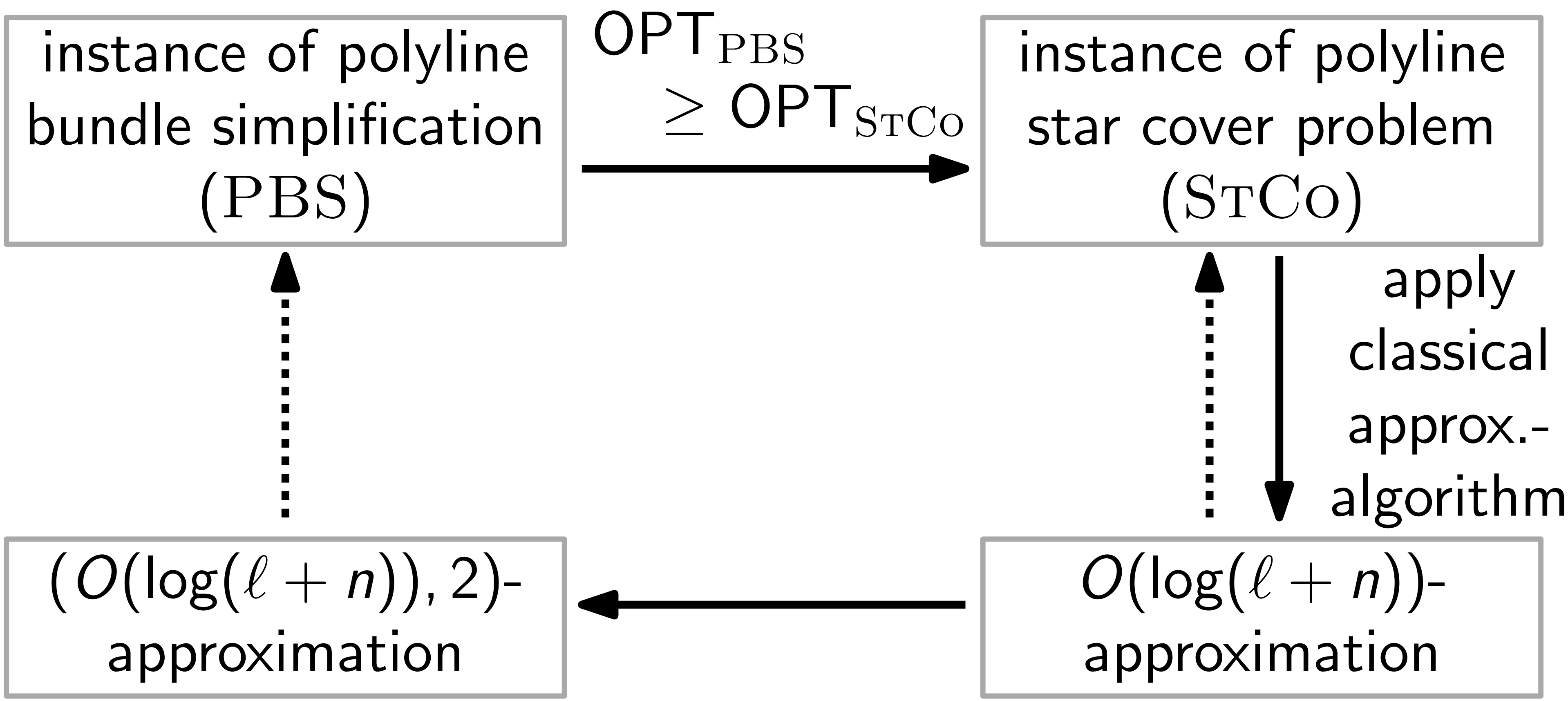
Bi-criteria Approximation



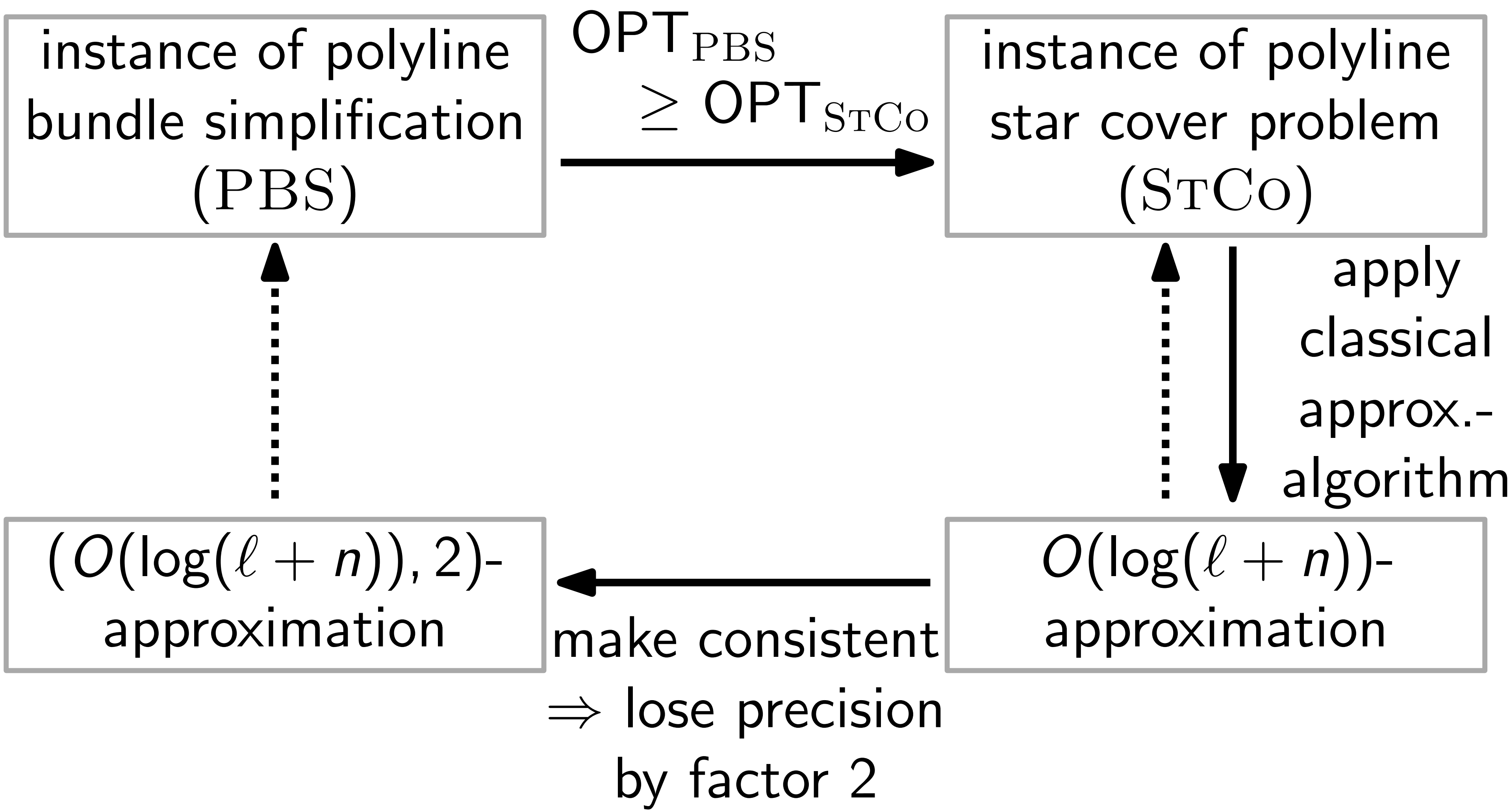
Bi-criteria Approximation



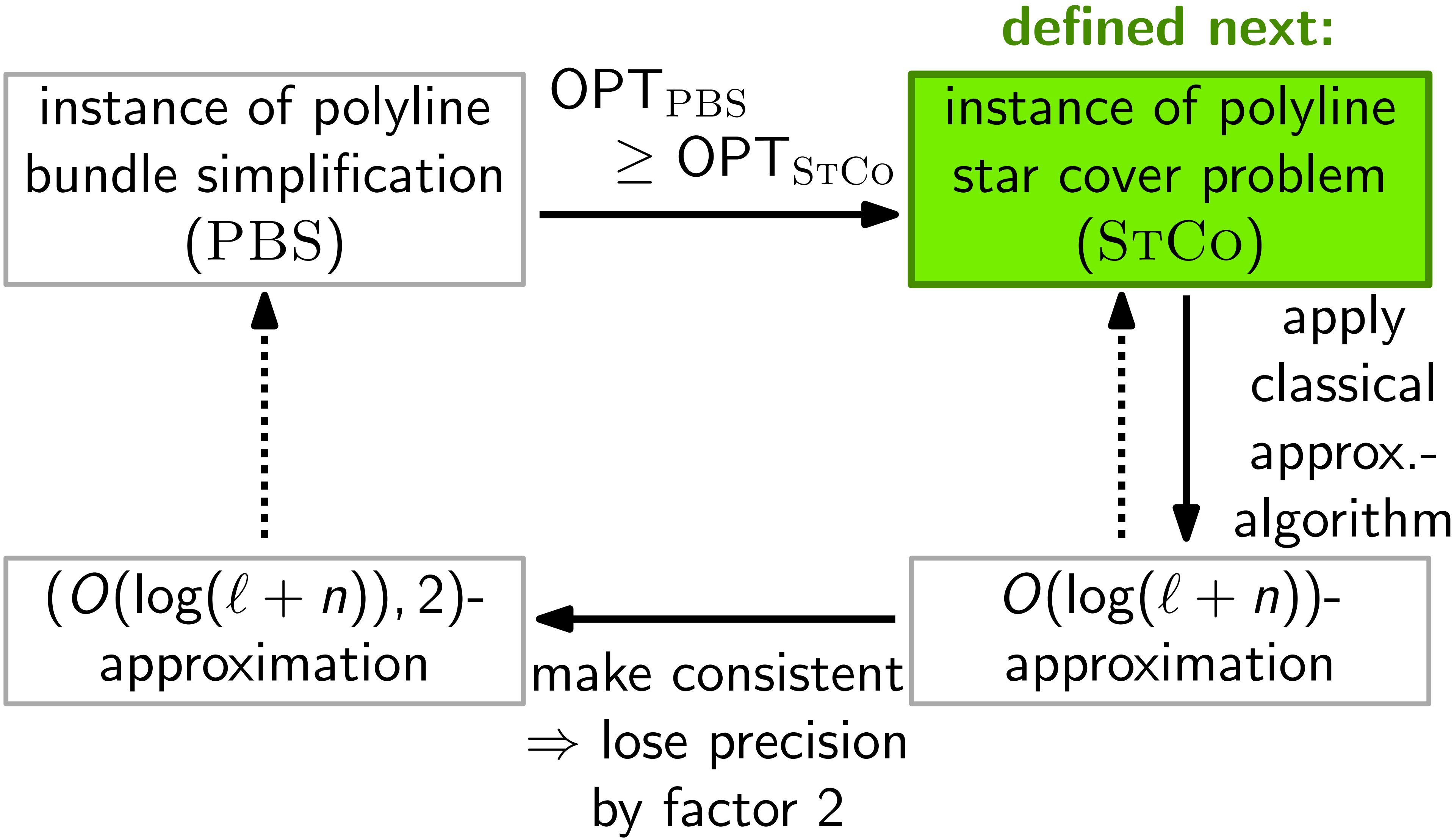
Bi-criteria Approximation



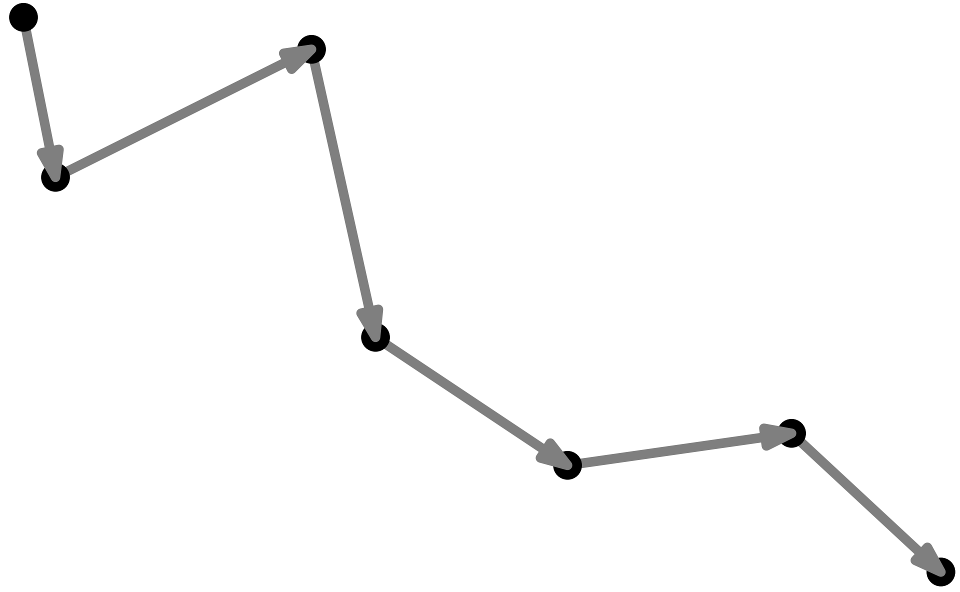
Bi-criteria Approximation



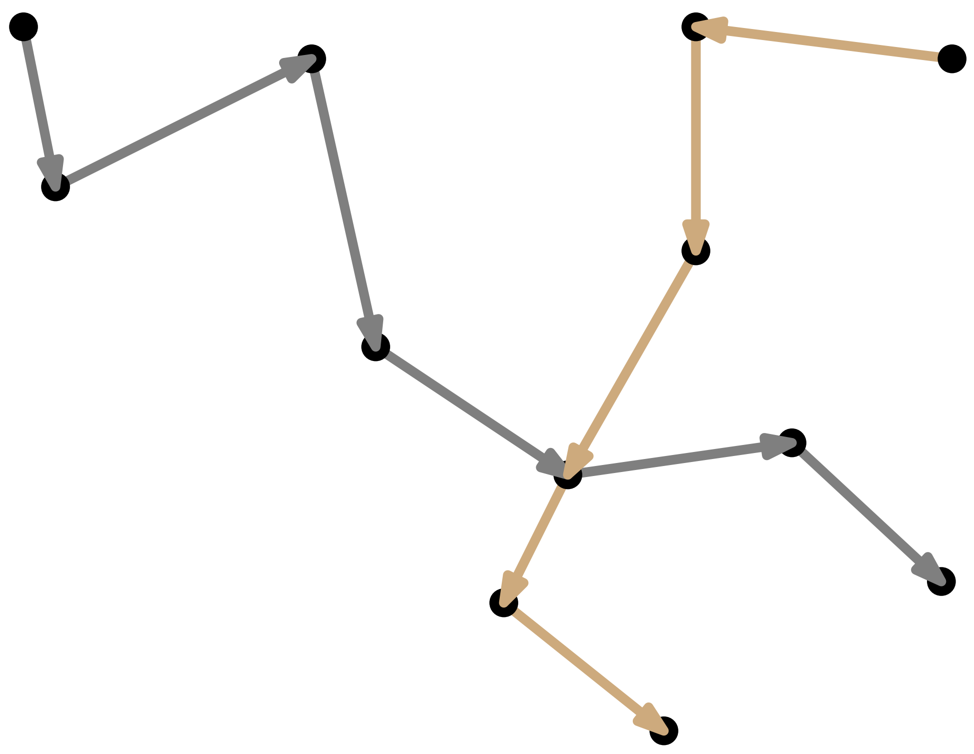
Bi-criteria Approximation



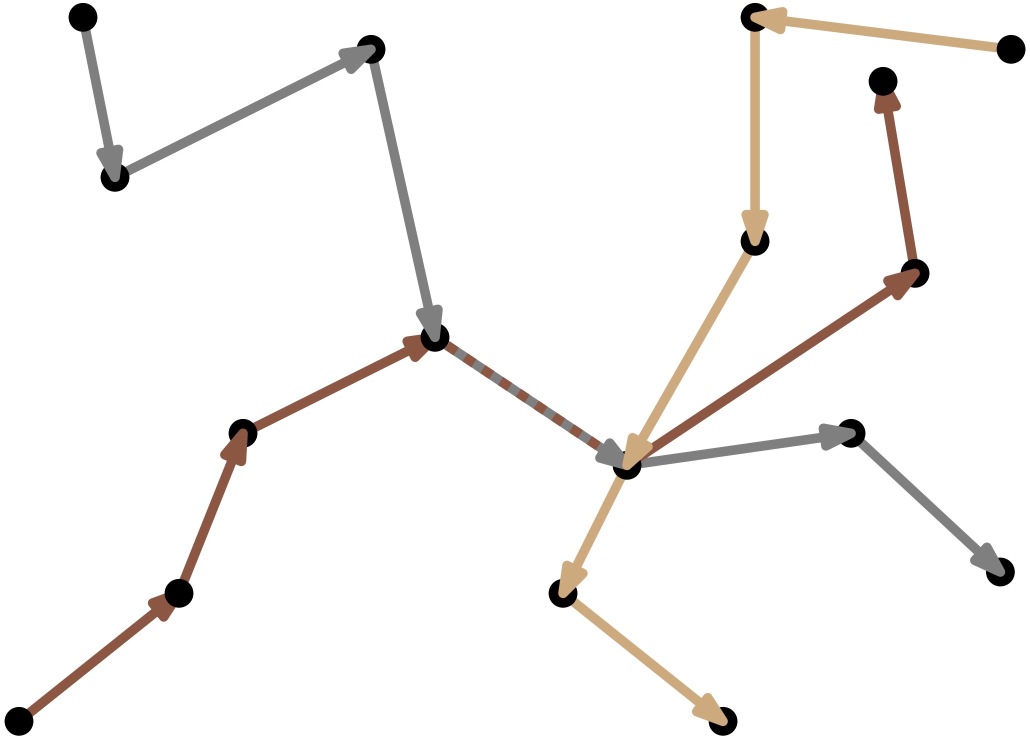
Polyline Star Cover Problem



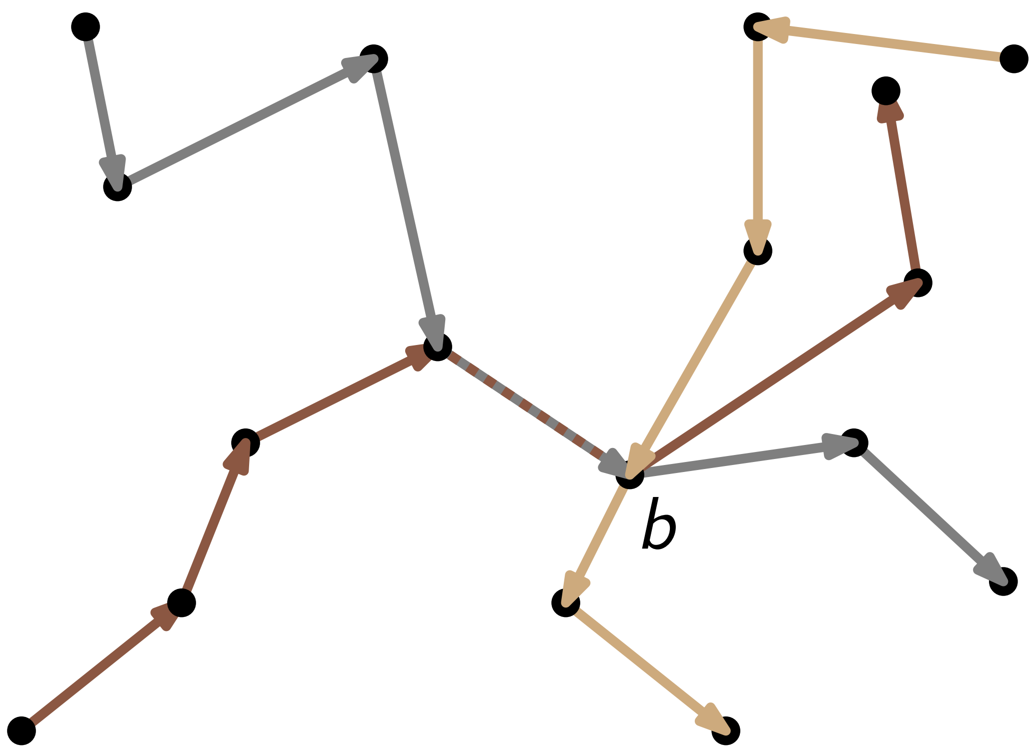
Polyline Star Cover Problem



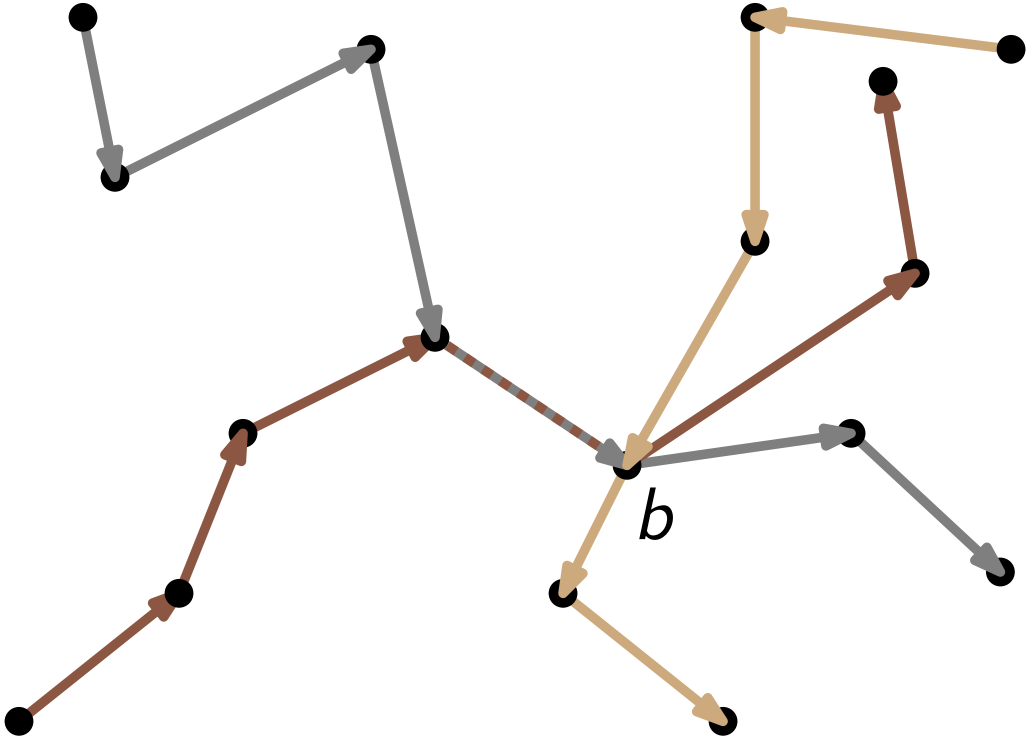
Polyline Star Cover Problem



Polyline Star Cover Problem

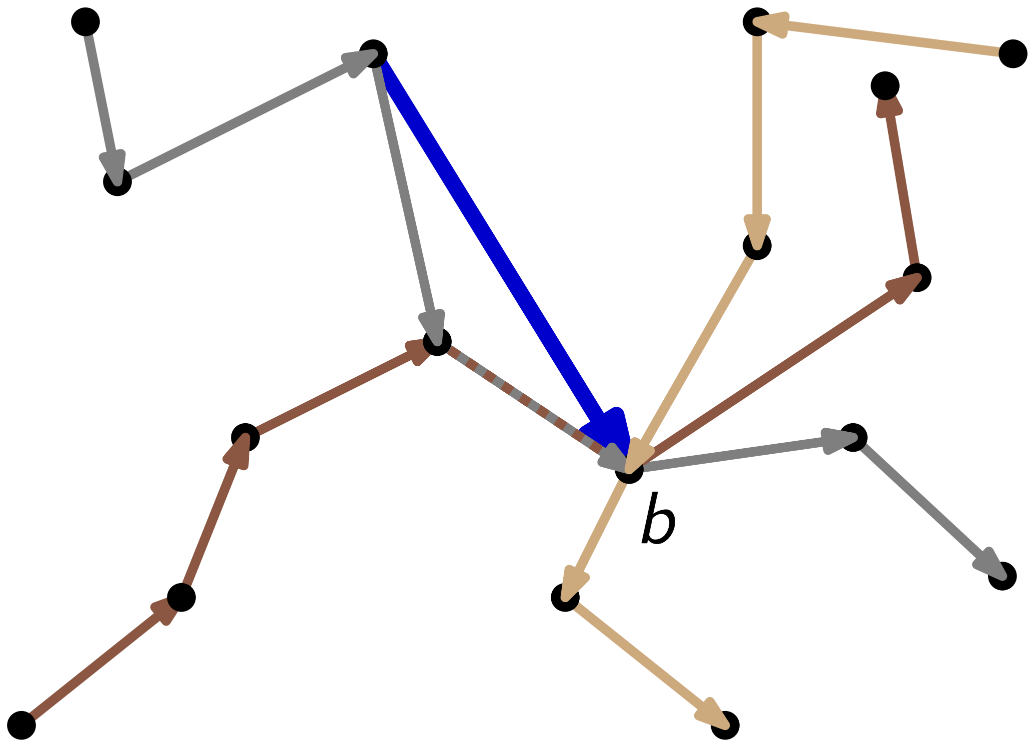


Polyline Star Cover Problem



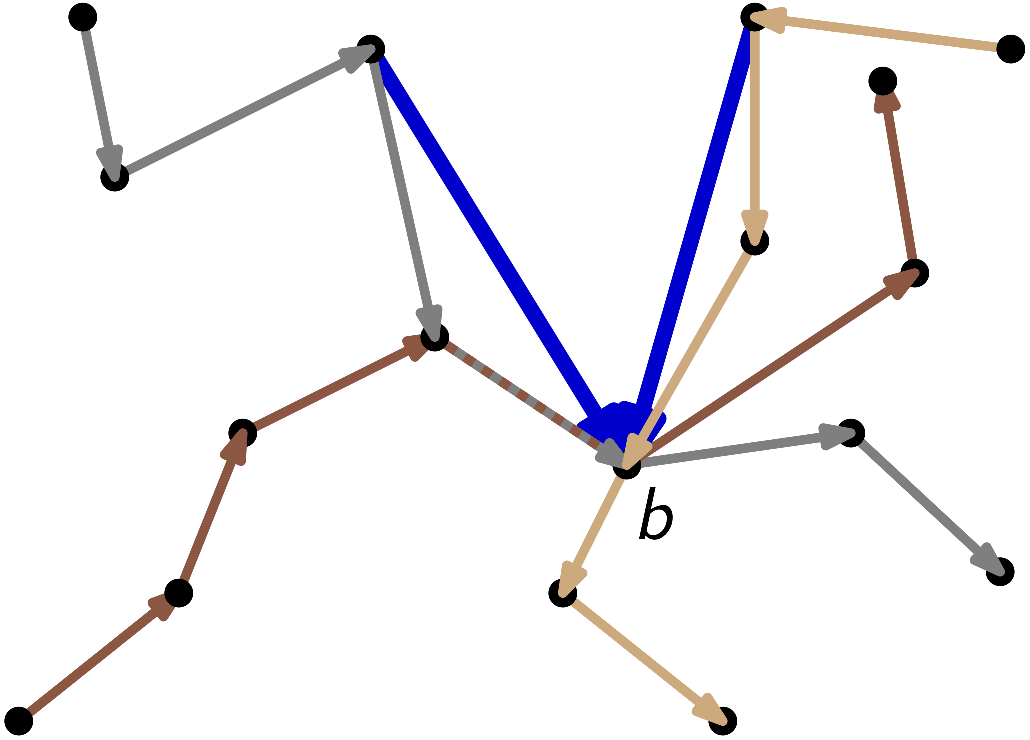
star around b

Polyline Star Cover Problem



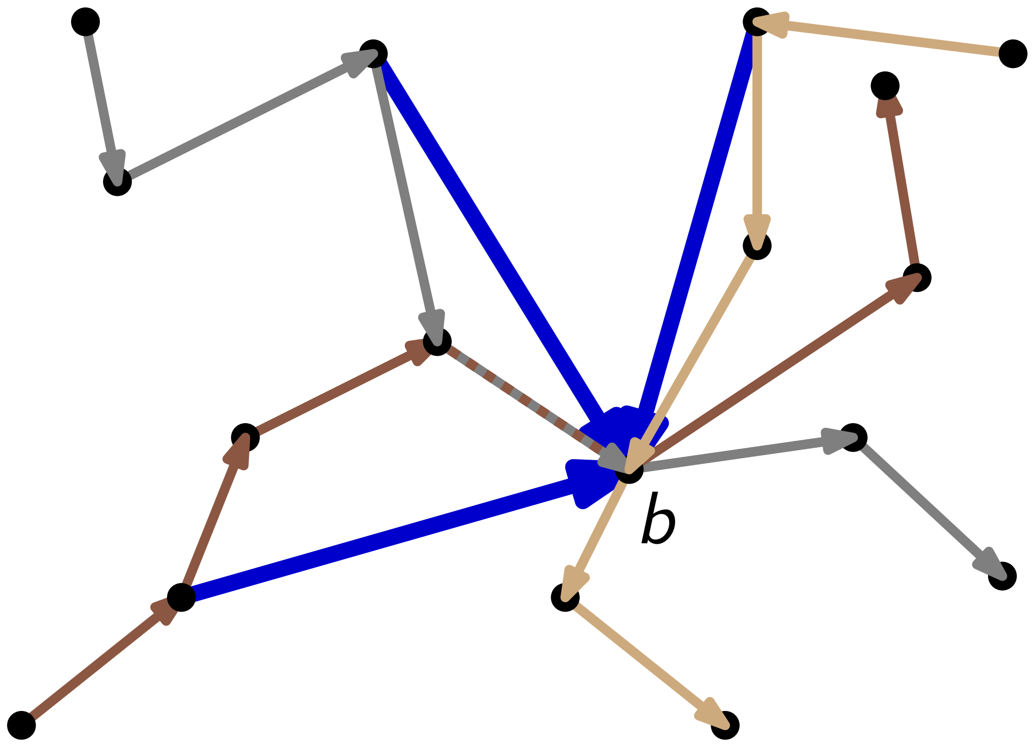
star around b

Polyline Star Cover Problem



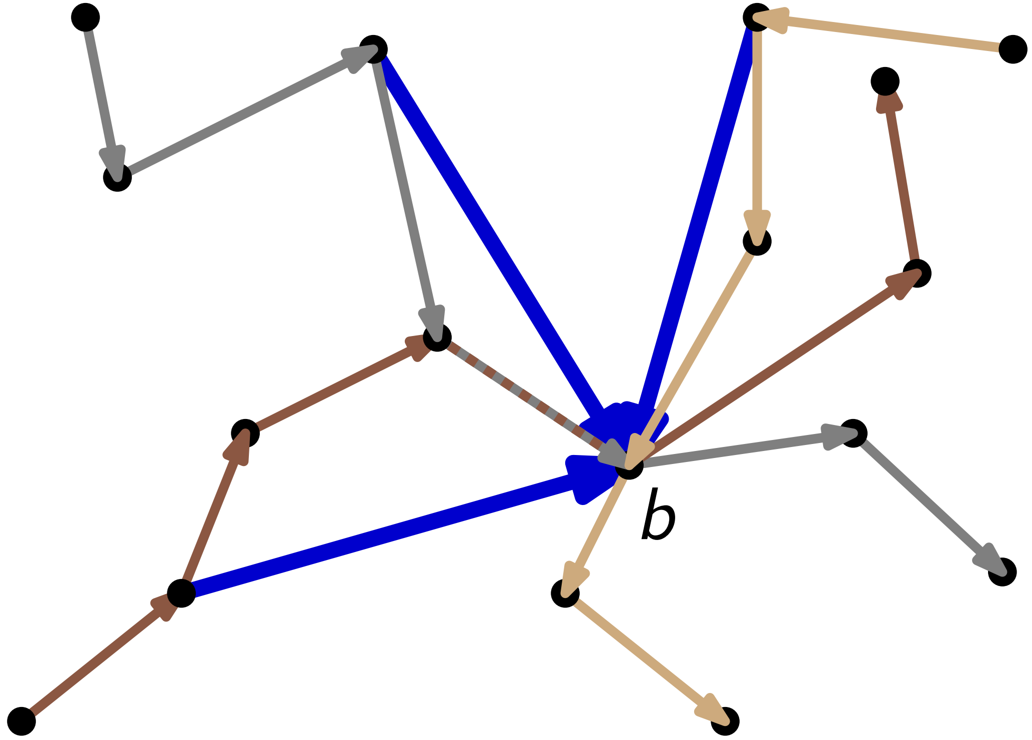
star around b

Polyline Star Cover Problem



star around b

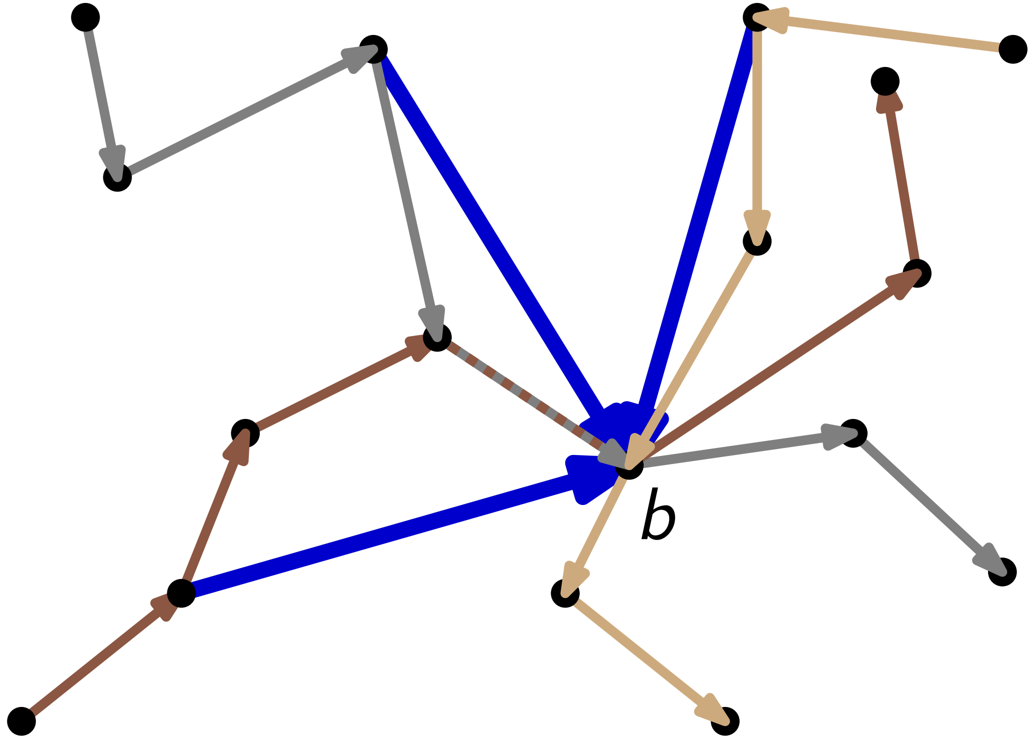
Polyline Star Cover Problem



star around b

Polyline Star Cover Problem
(for an instance of PBS)

Polyline Star Cover Problem



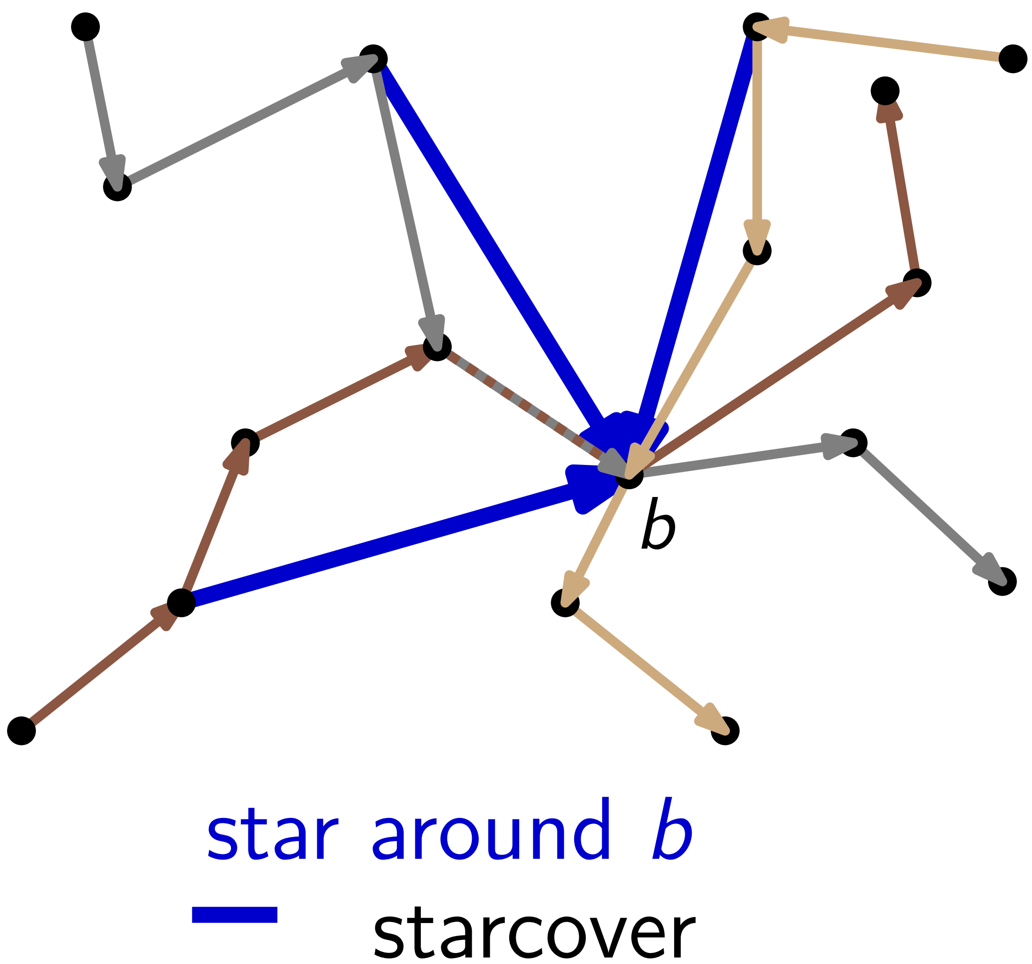
star around b

Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem

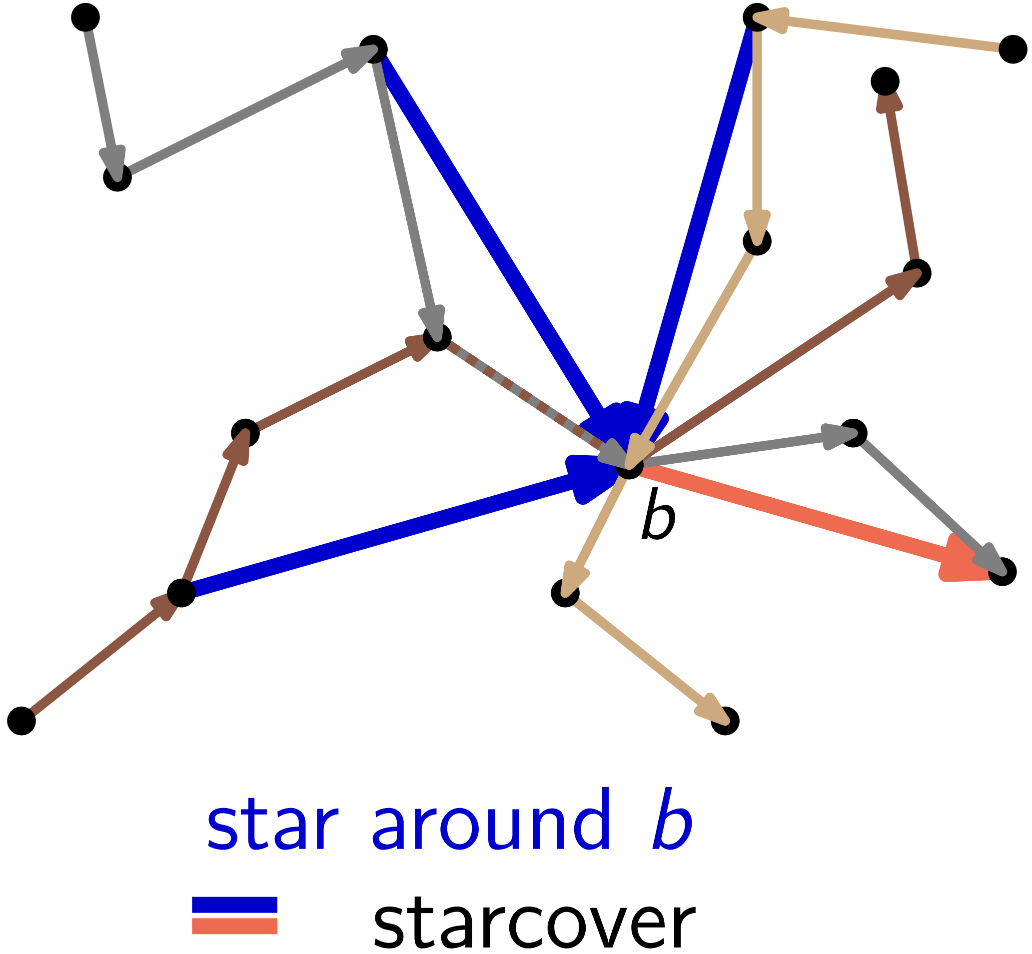


Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem

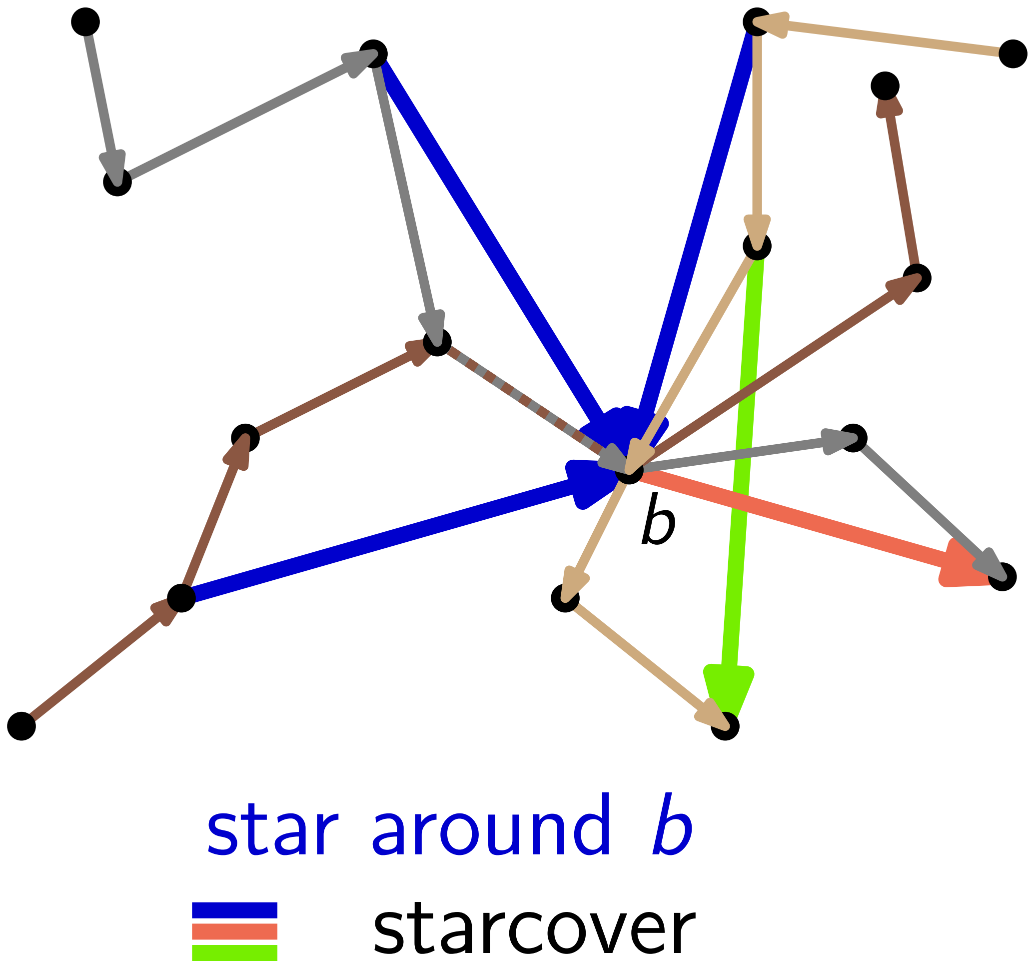


Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem

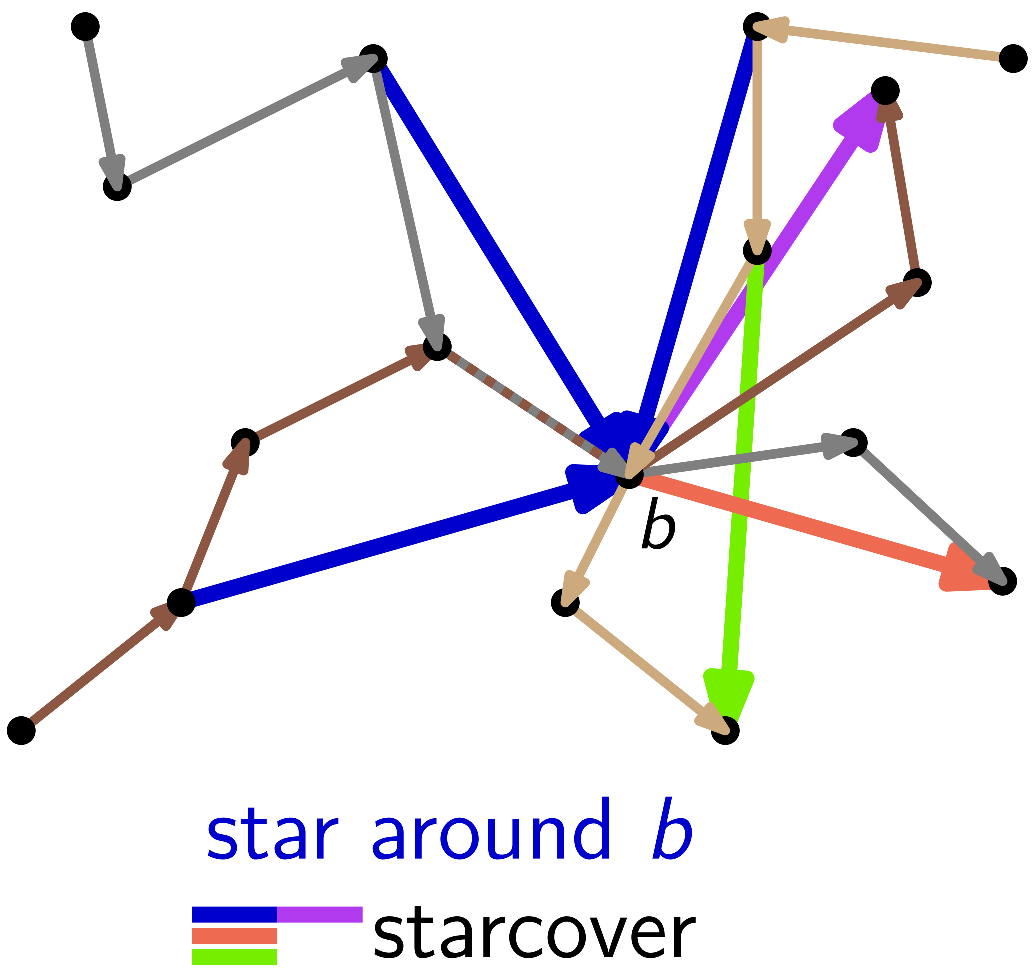


Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem

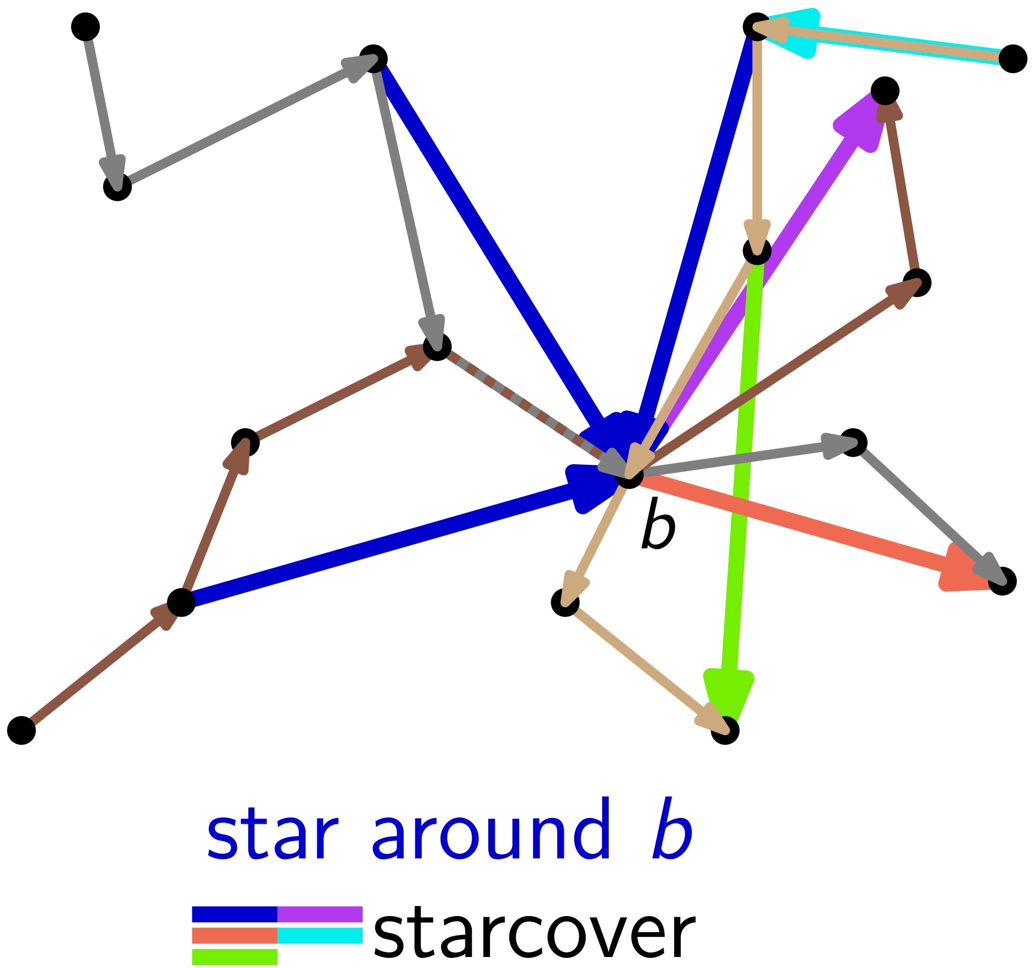


Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem

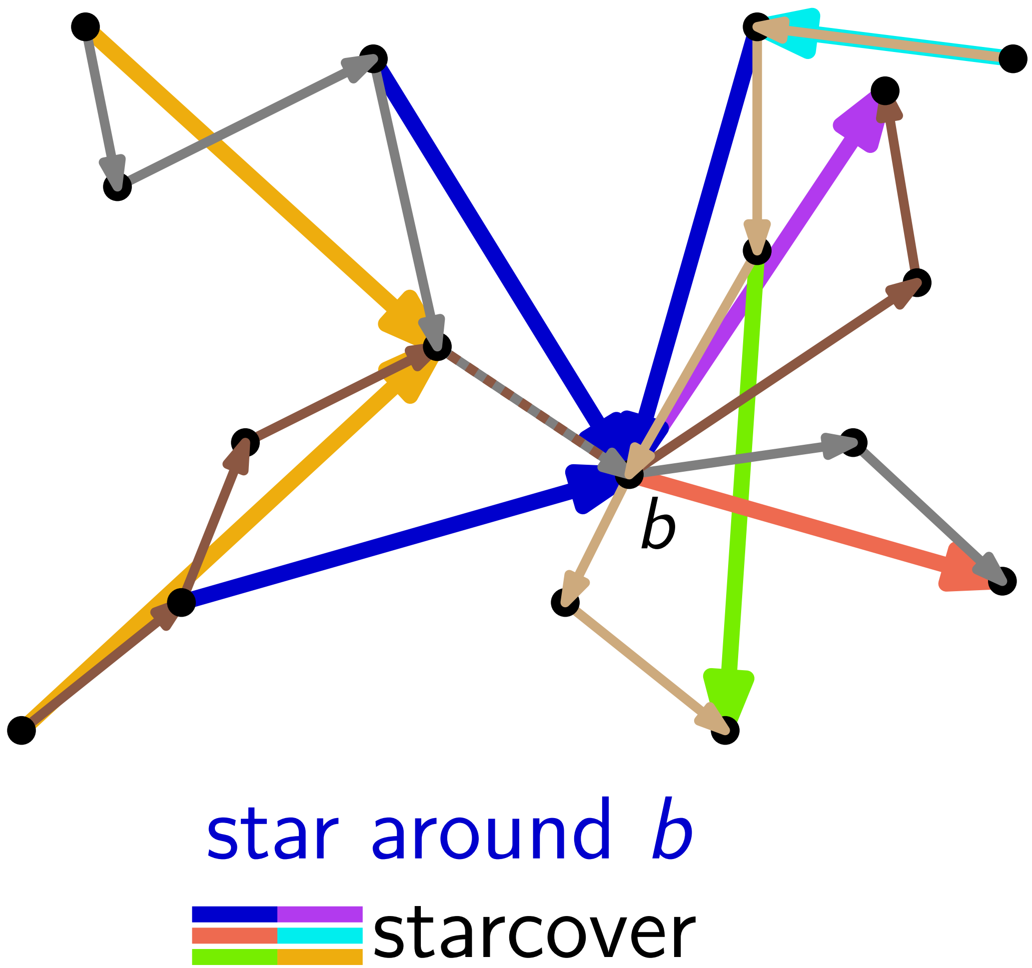


Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem

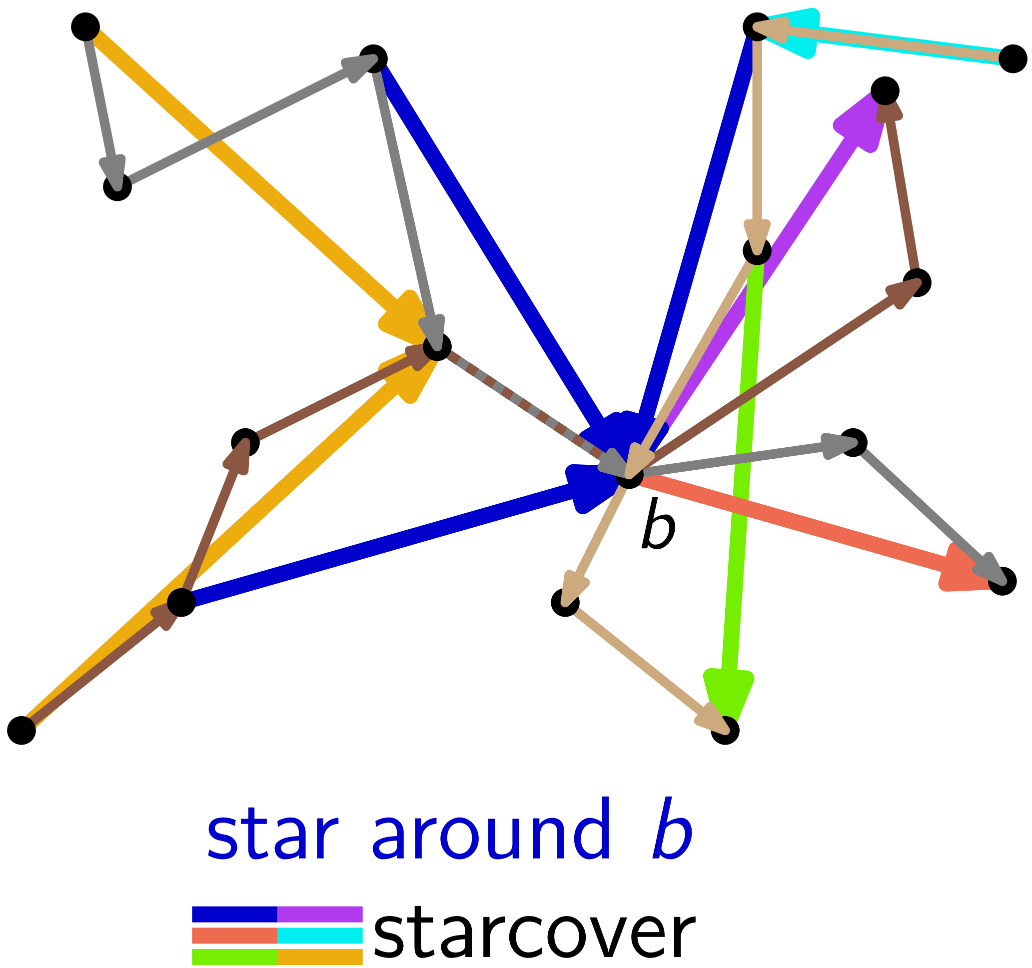


Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Polyline Star Cover Problem



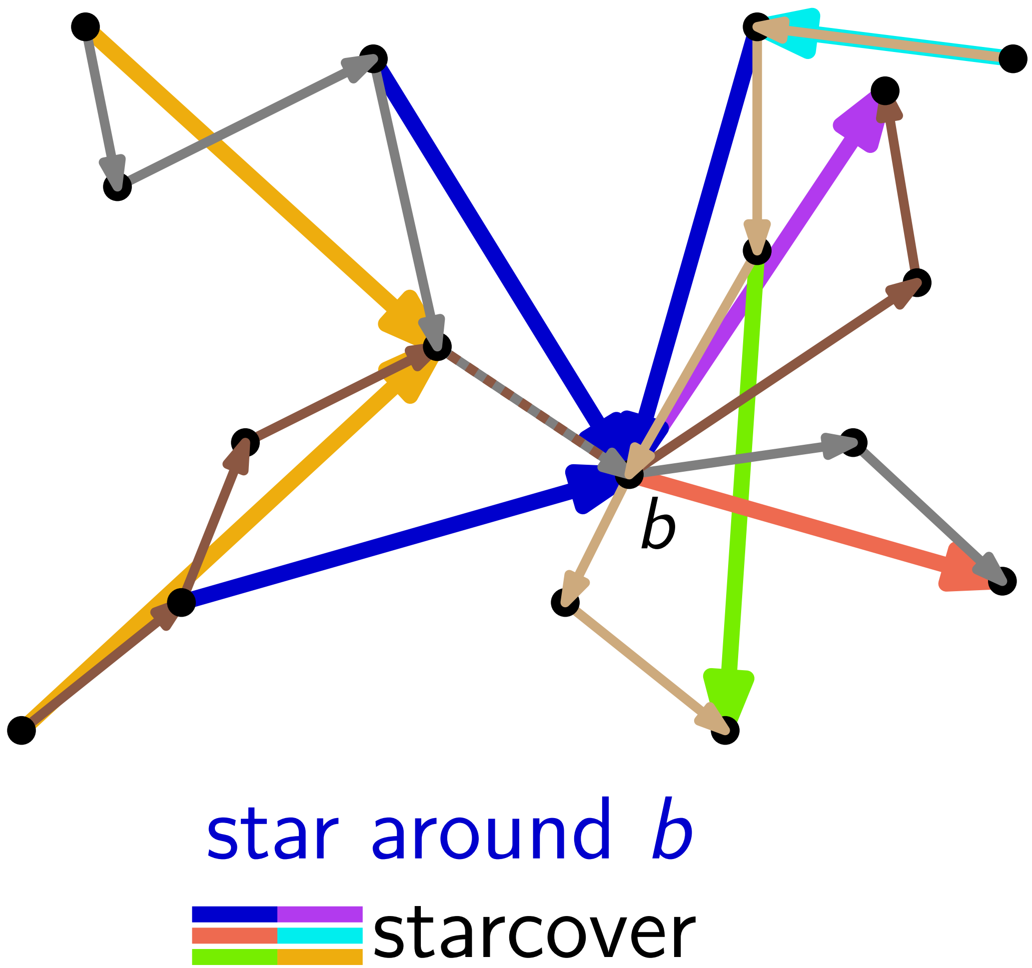
Polyline Star Cover Problem
(for an instance of PBS)

Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Observations:

Polyline Star Cover Problem



Polyline Star Cover Problem
(for an instance of PBS)

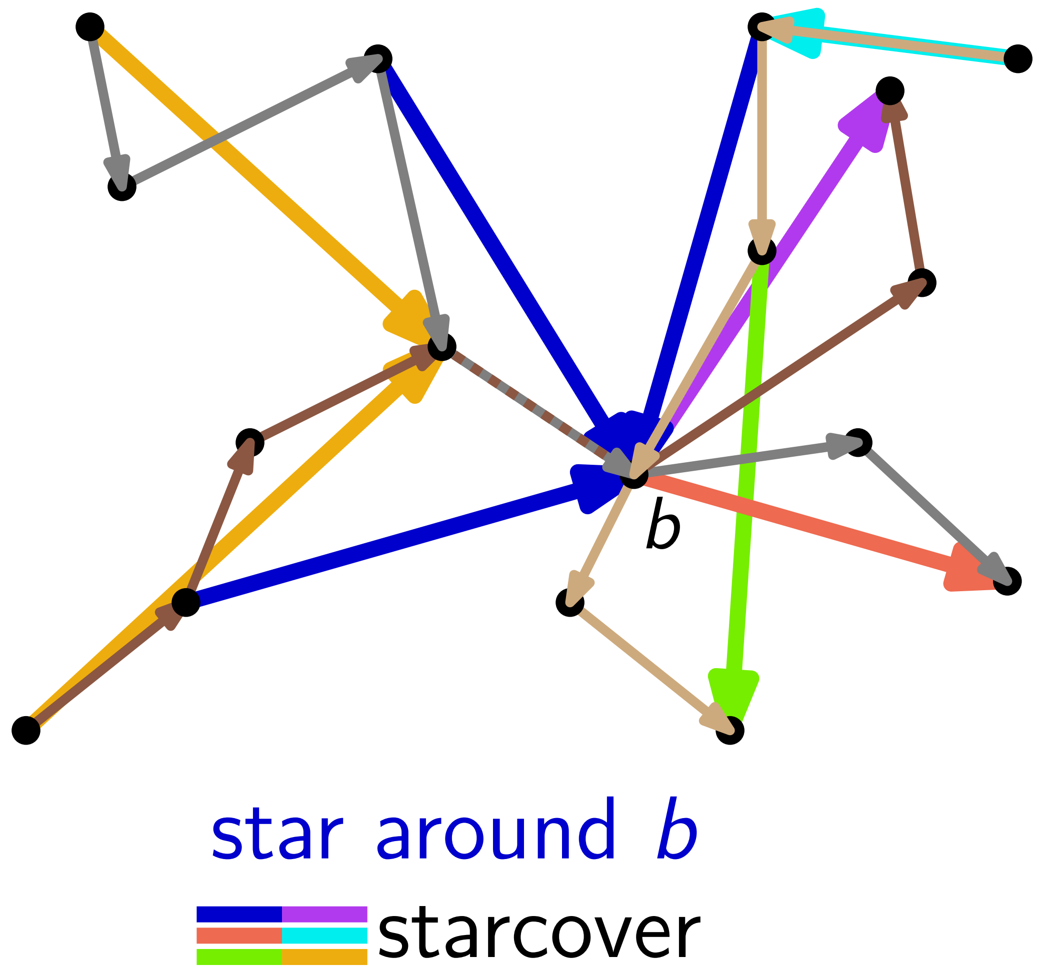
Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Observations:

- it suffices to consider all n maximal stars

Polyline Star Cover Problem



Polyline Star Cover Problem
(for an instance of PBS)

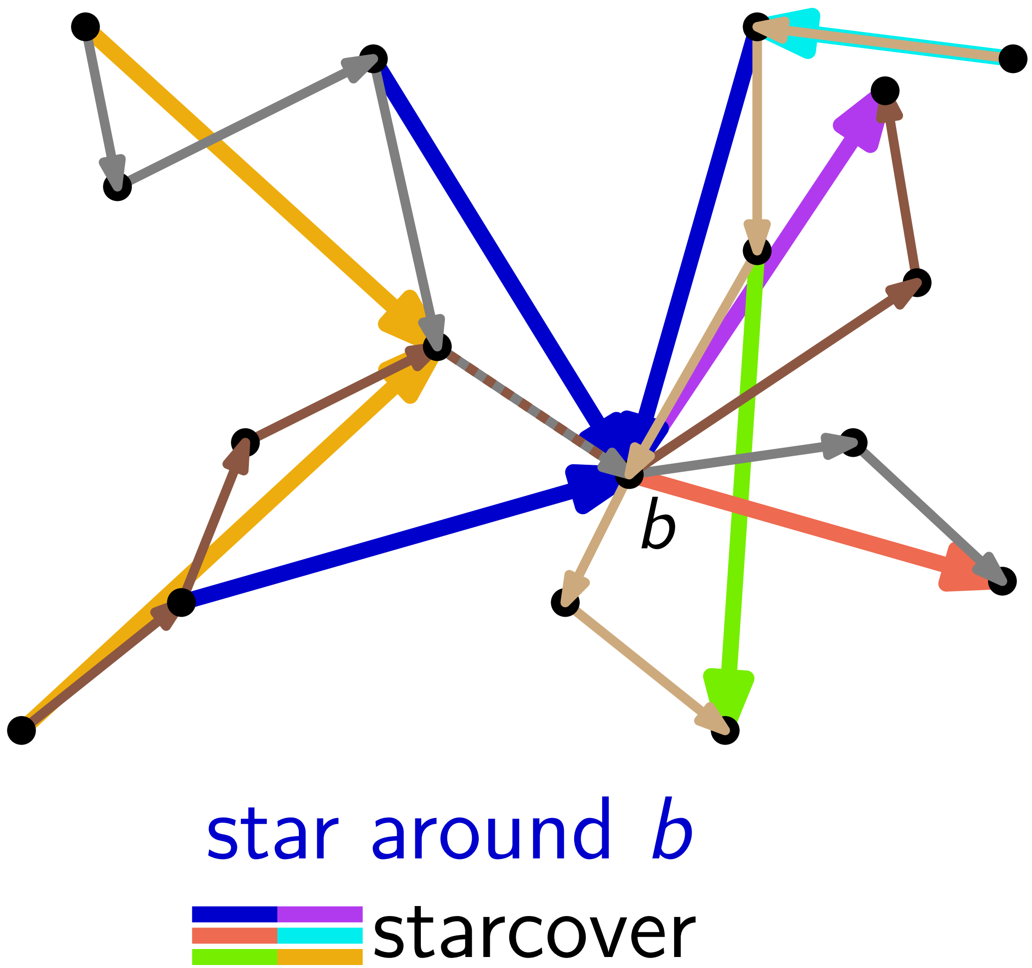
Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

Observations:

- it suffices to consider all n maximal stars
- any star covers $\leq n \cdot \ell$ polyline–segment pairs

Polyline Star Cover Problem



Polyline Star Cover Problem
(for an instance of PBS)

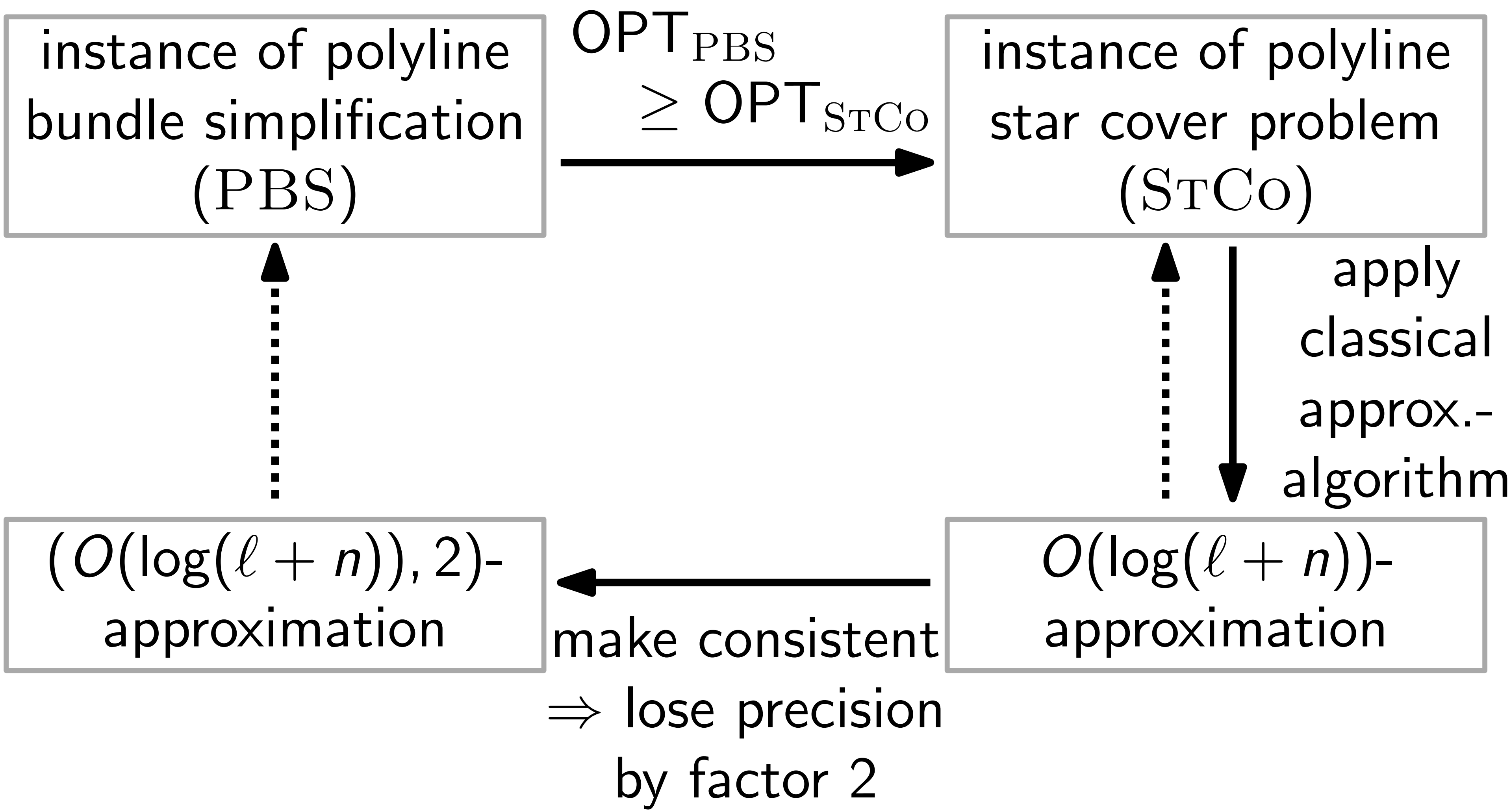
Given: all stars

Find: minimum size set of stars covering all polyline–segment pairs

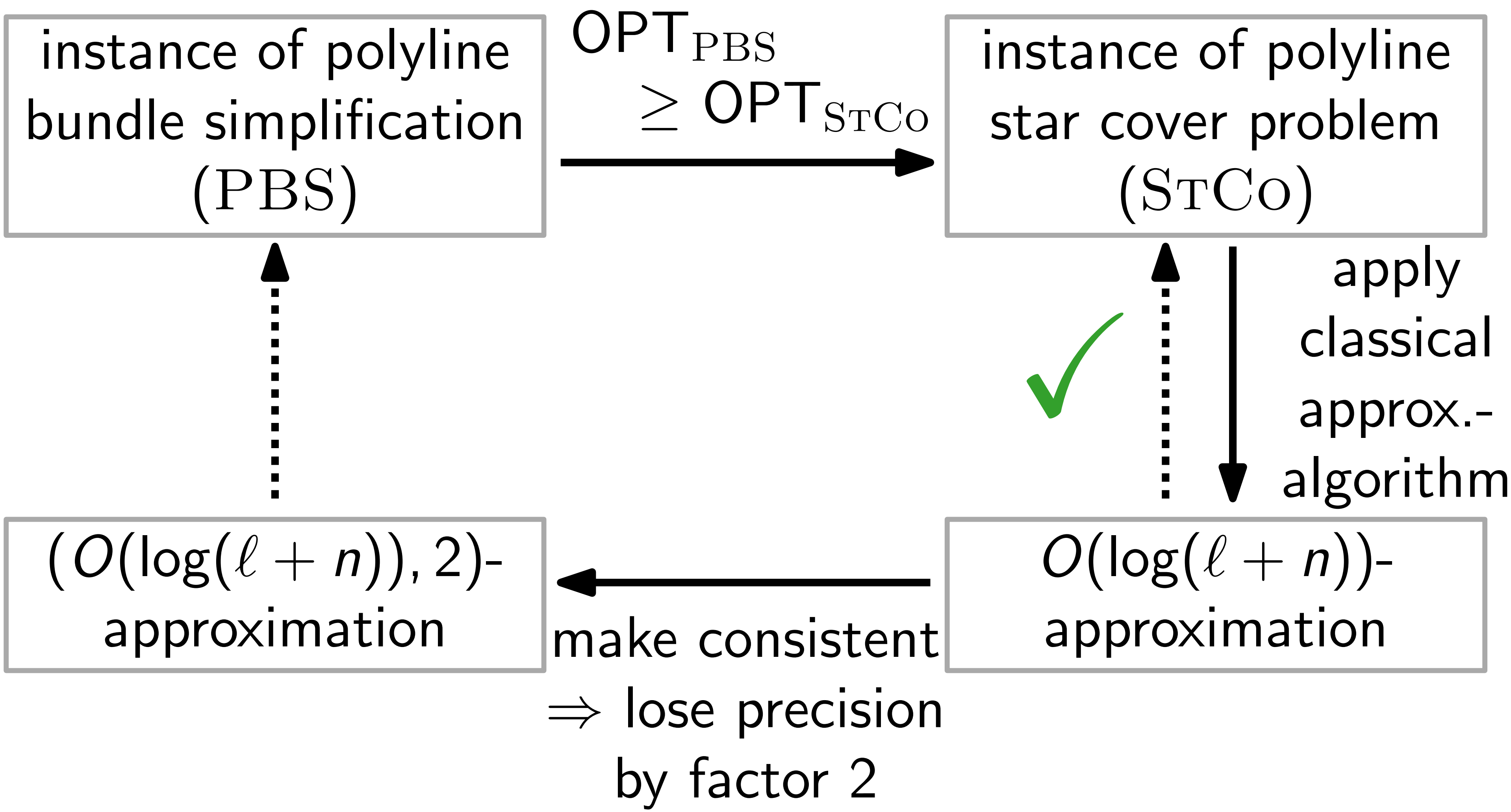
Observations:
- it suffices to consider all n maximal stars
- any star covers $\leq n \cdot \ell$ polyline–segment pairs

$\Rightarrow$ standard greedy set cover yields $O(\log(n \cdot \ell)) = O(\log(n + \ell))$ approximation

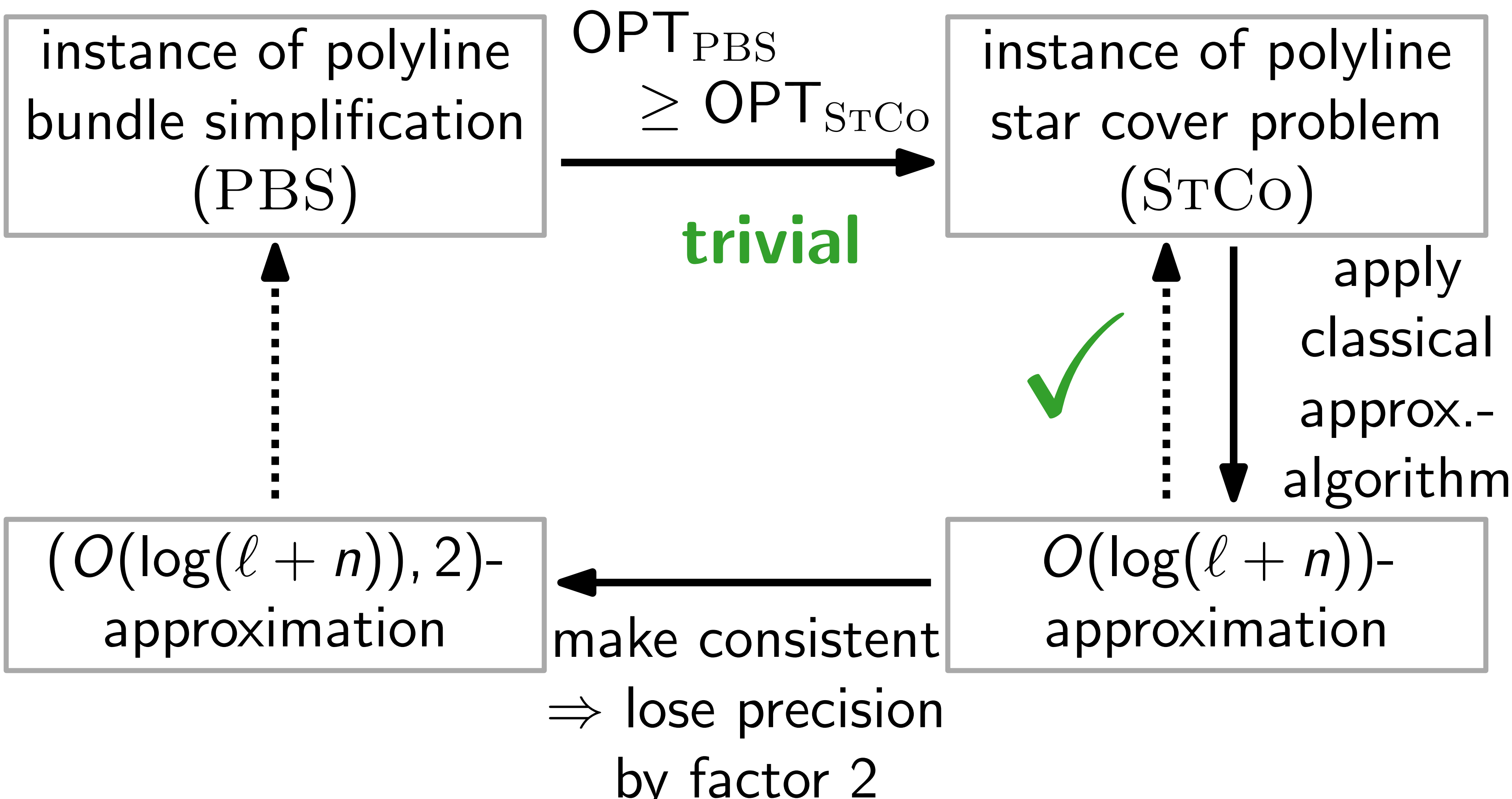
Bi-criteria Approximation



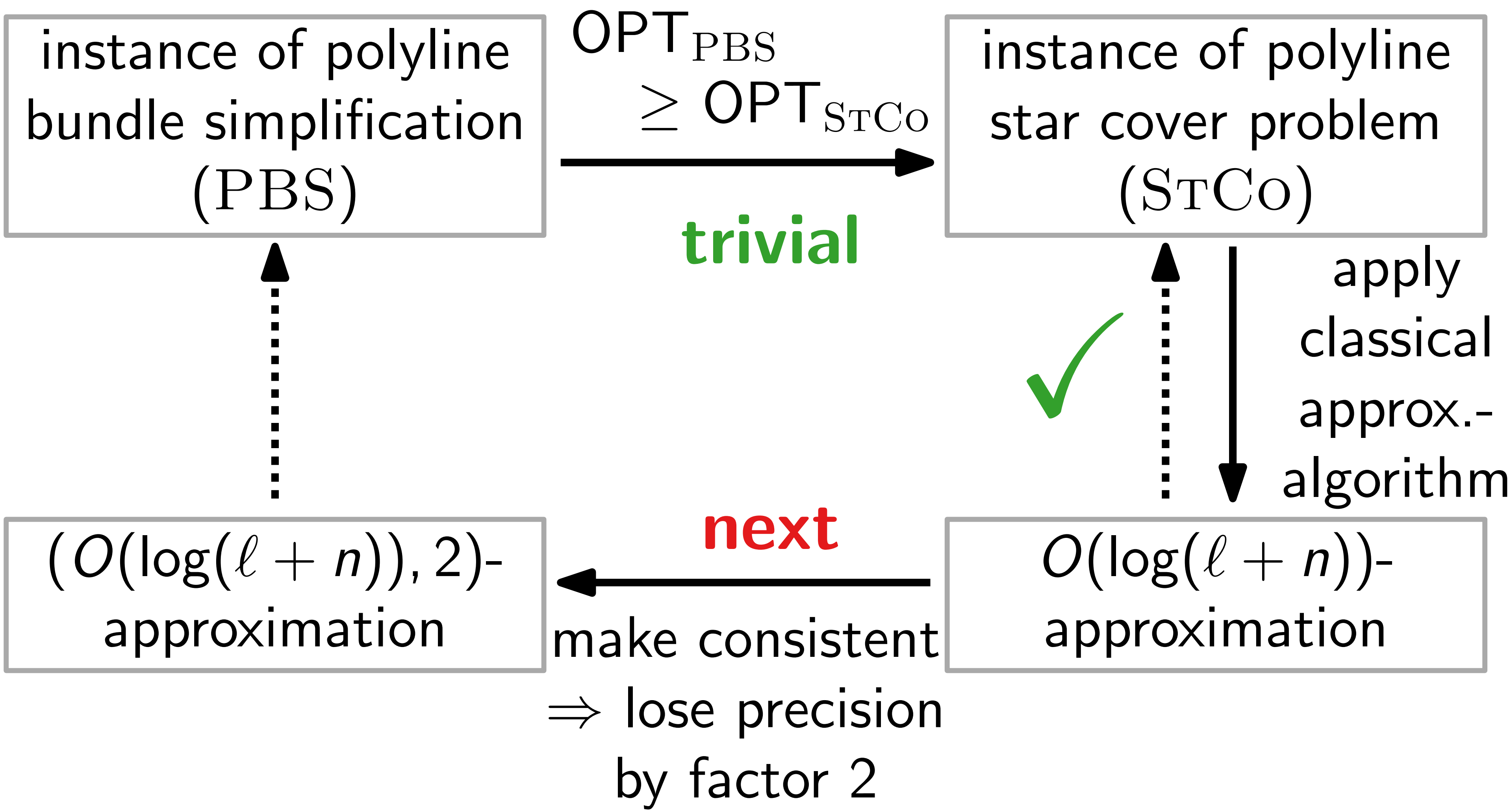
Bi-criteria Approximation



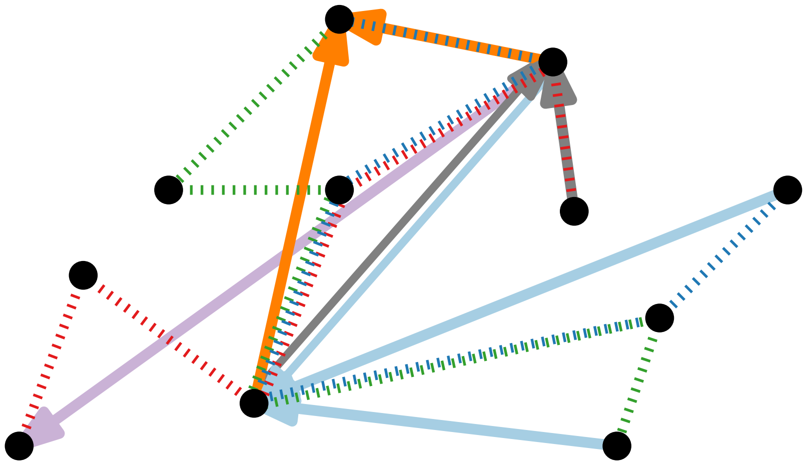
Bi-criteria Approximation



Bi-criteria Approximation

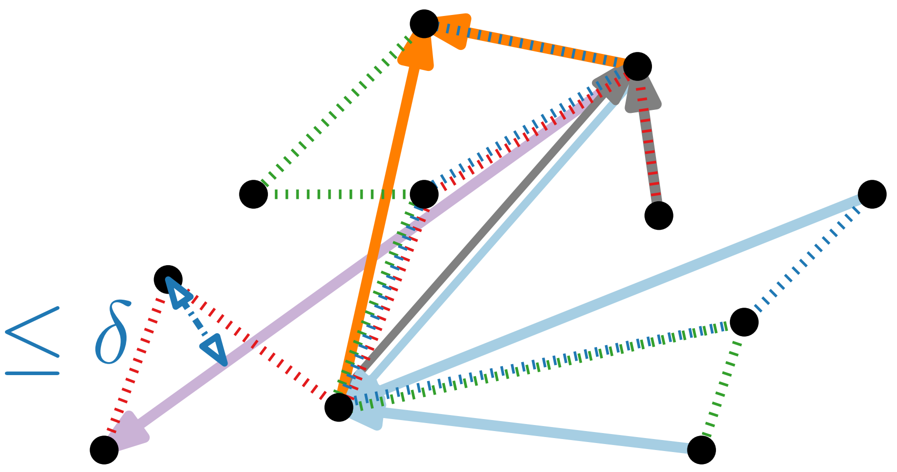


StCo solution → PBS solution



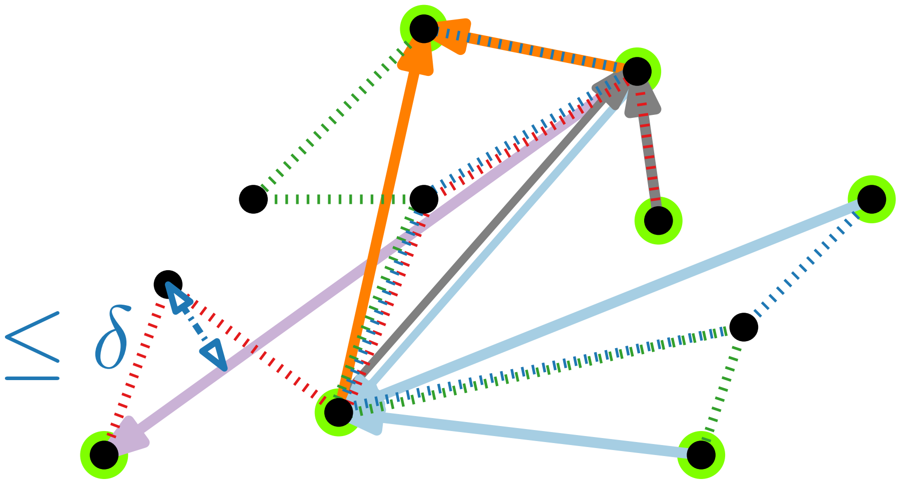
StCo solution

StCo solution → PBS solution



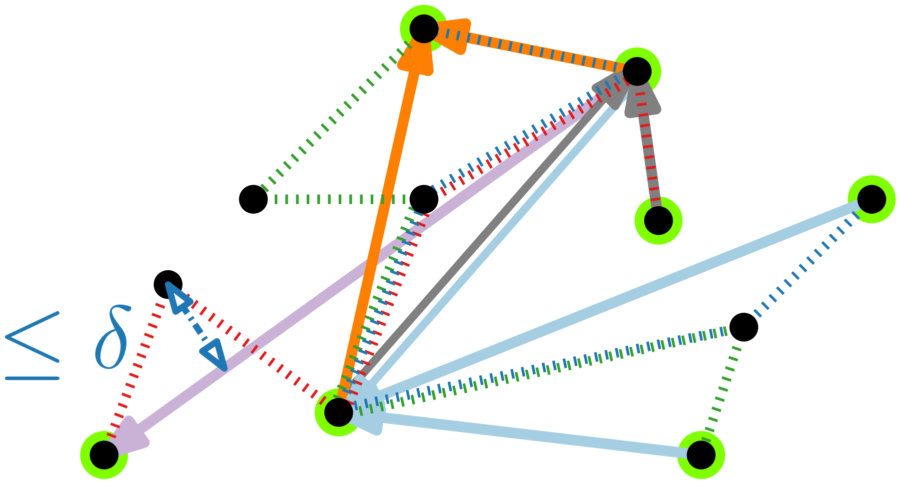
StCo solution

StCo solution → PBS solution

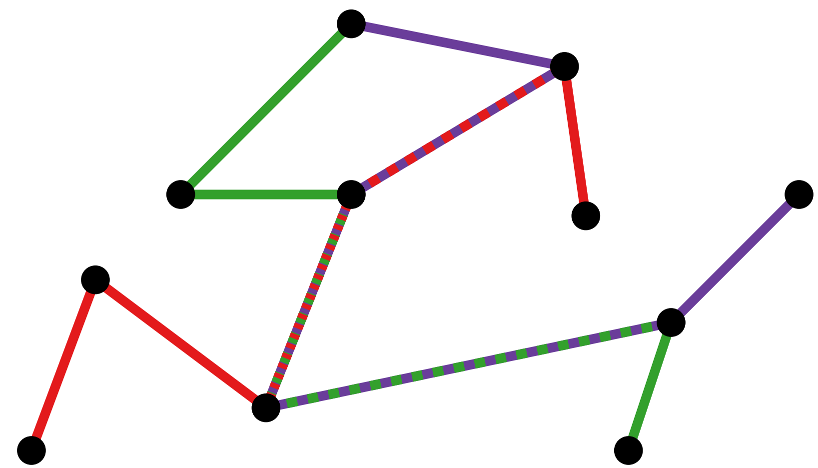


StCo solution

StCo solution → PBS solution

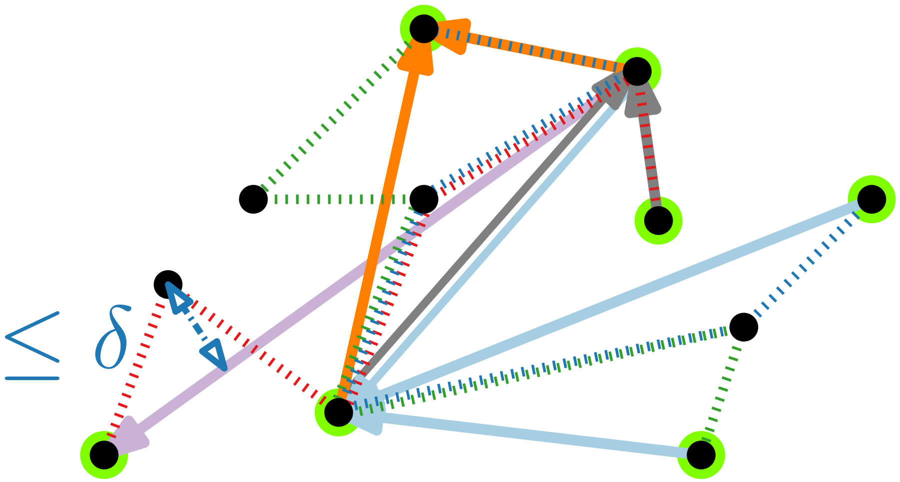


StCo solution

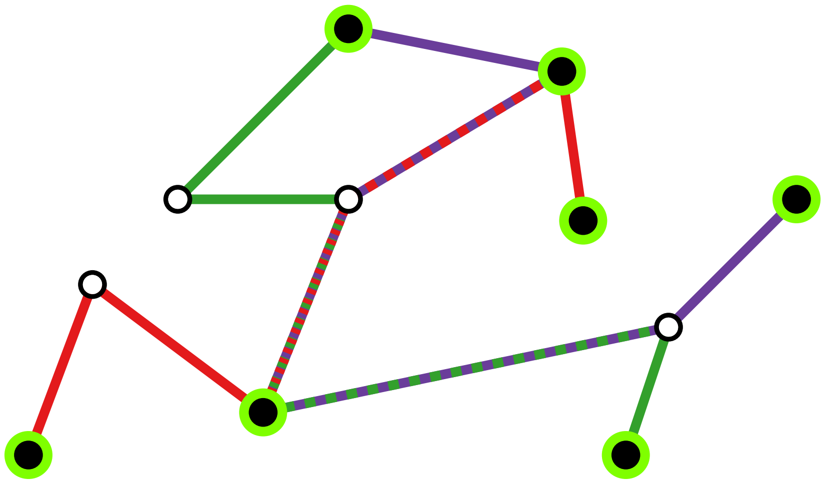


PBS solution

StCo solution → PBS solution

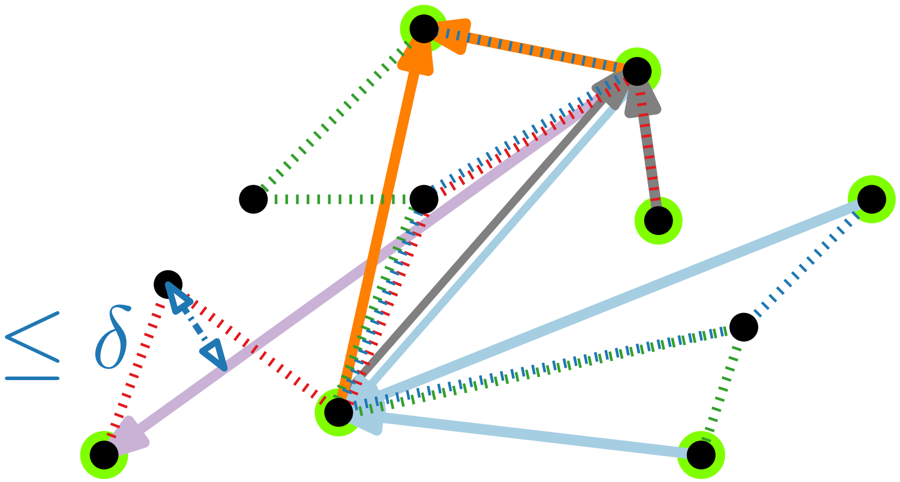


StCo solution

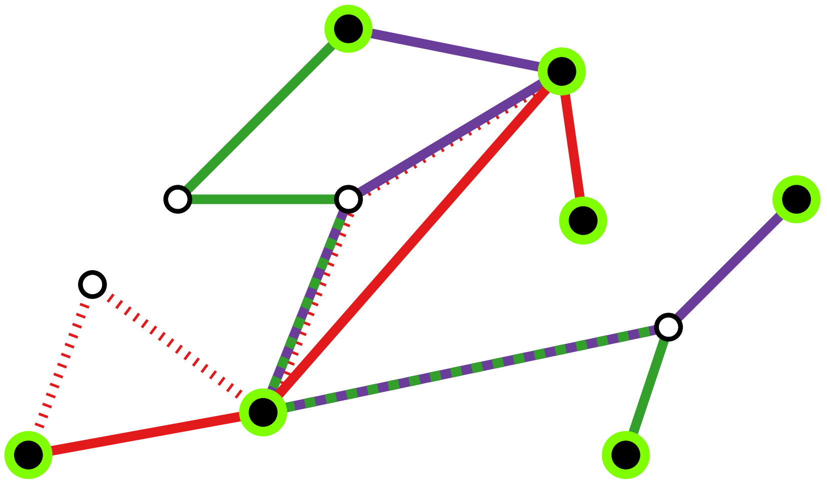


PBS solution

StCo solution → PBS solution

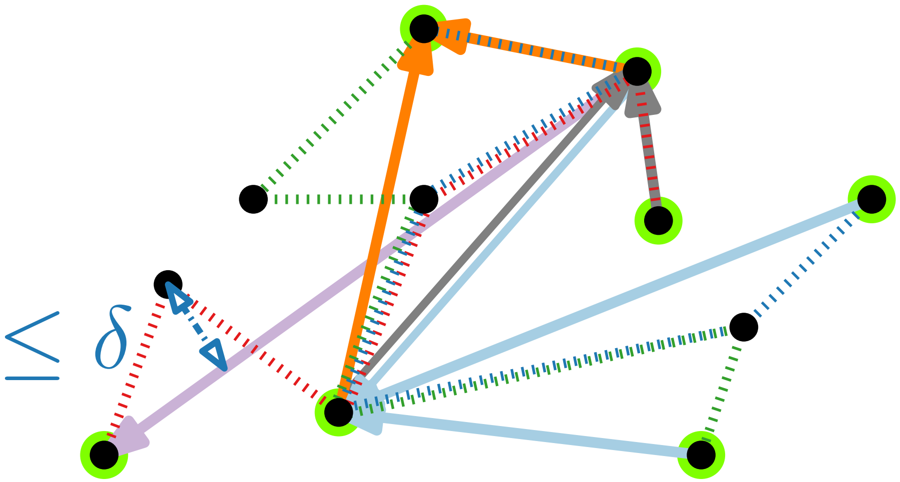


StCo solution

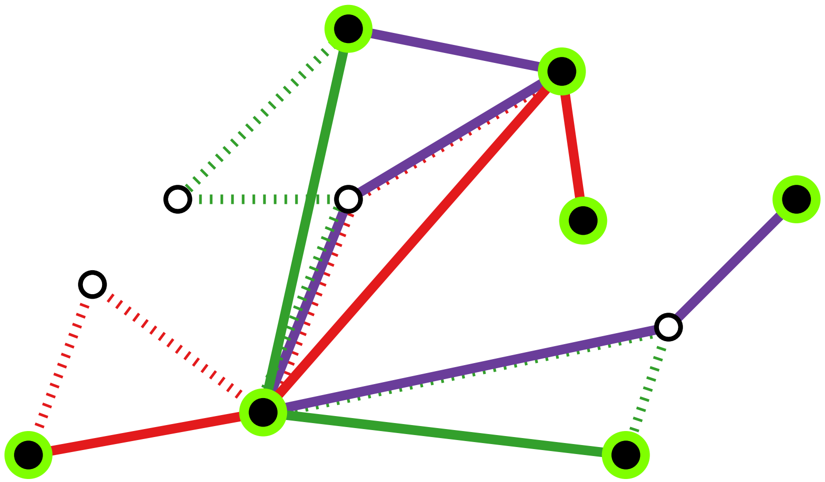


PBS solution

StCo solution → PBS solution

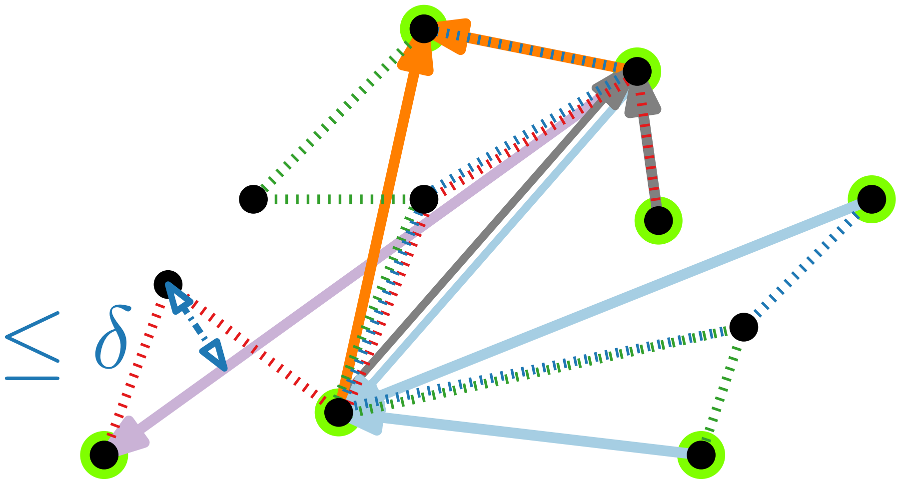


StCo solution

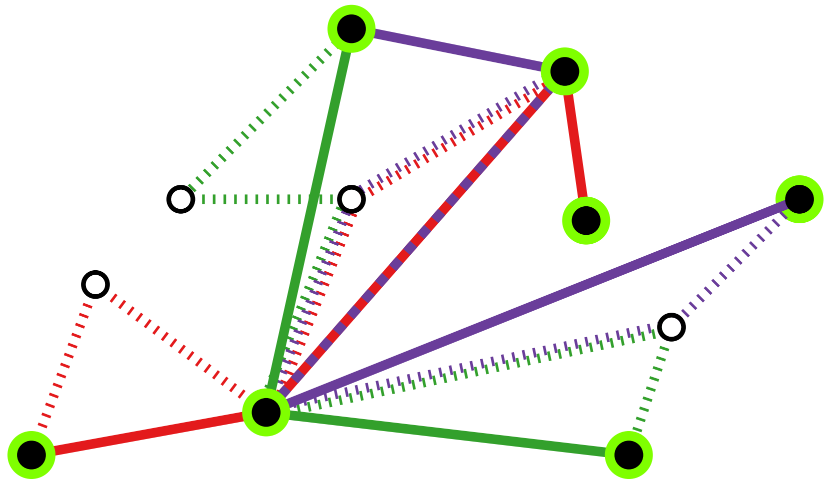


PBS solution

StCo solution → PBS solution

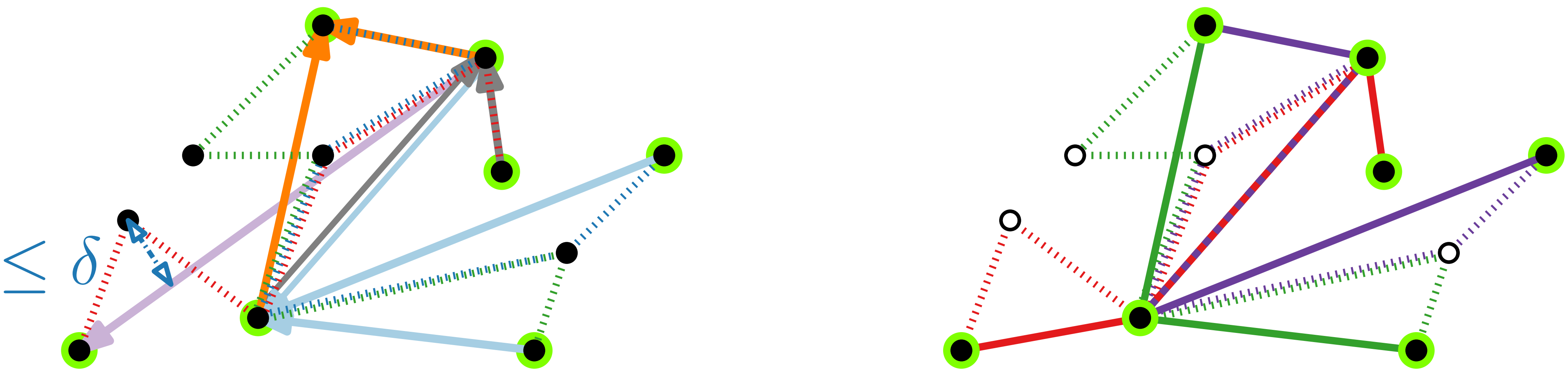


StCo solution



PBS solution

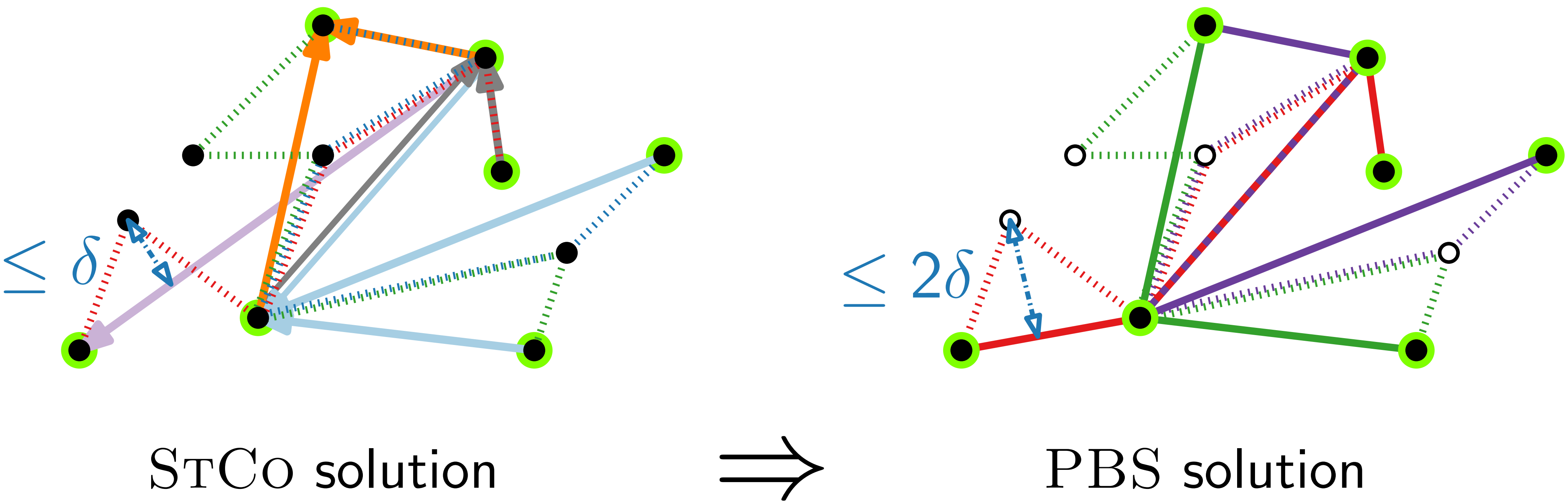
StCo solution → PBS solution



StCo solution $\Rightarrow$ PBS solution

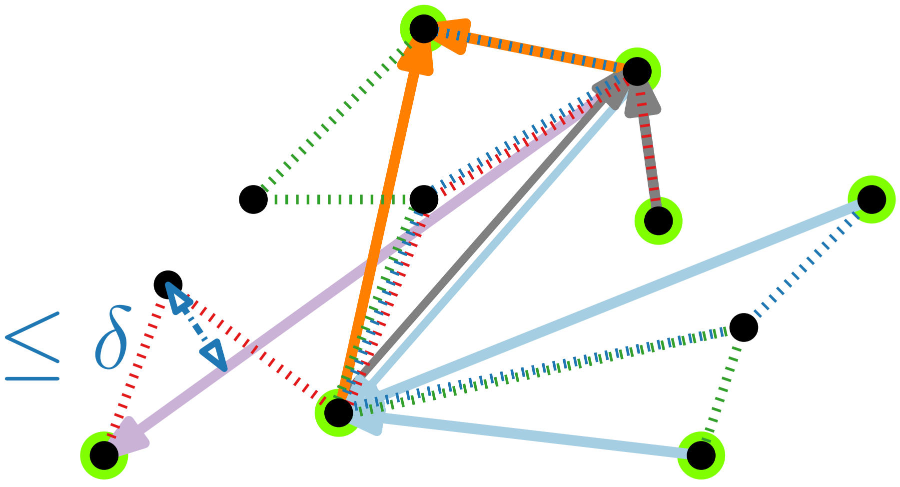
Lemma: [Agarwal et al.]
If $d_{\text{Fréchet}}((b_y, b_z), (b_y, \ldots, b_i, \ldots, b_j, \ldots, b_z)) \leq \delta$,
then $d_{\text{Fréchet}}((b_i, b_j), (b_i, \ldots, b_j)) \leq 2\delta$
for all $y \leq i < j \leq z$.

StCo solution → PBS solution

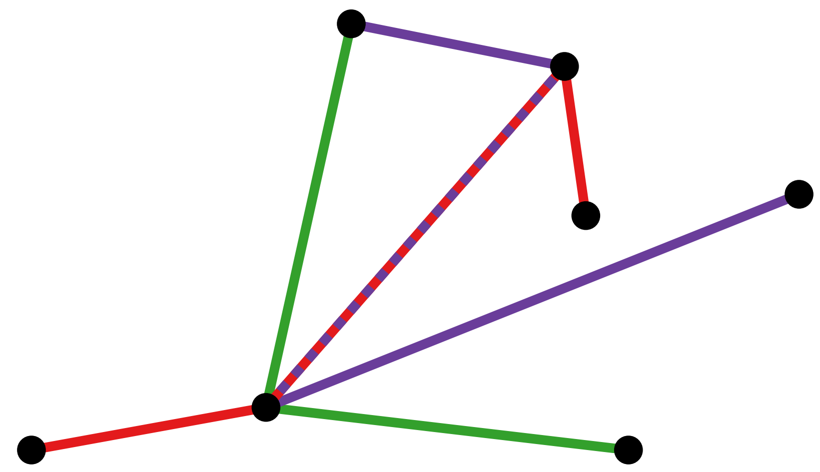


Lemma: [Agarwal et al.]
If $d_{\text{Fréchet}}((b_y, b_z), (b_y, \ldots, b_i, \ldots, b_j, \ldots, b_z)) \leq \delta$,
then $d_{\text{Fréchet}}((b_i, b_j), (b_i, \ldots, b_j)) \leq 2\delta$
for all $y \leq i < j \leq z$.

StCo solution → PBS solution

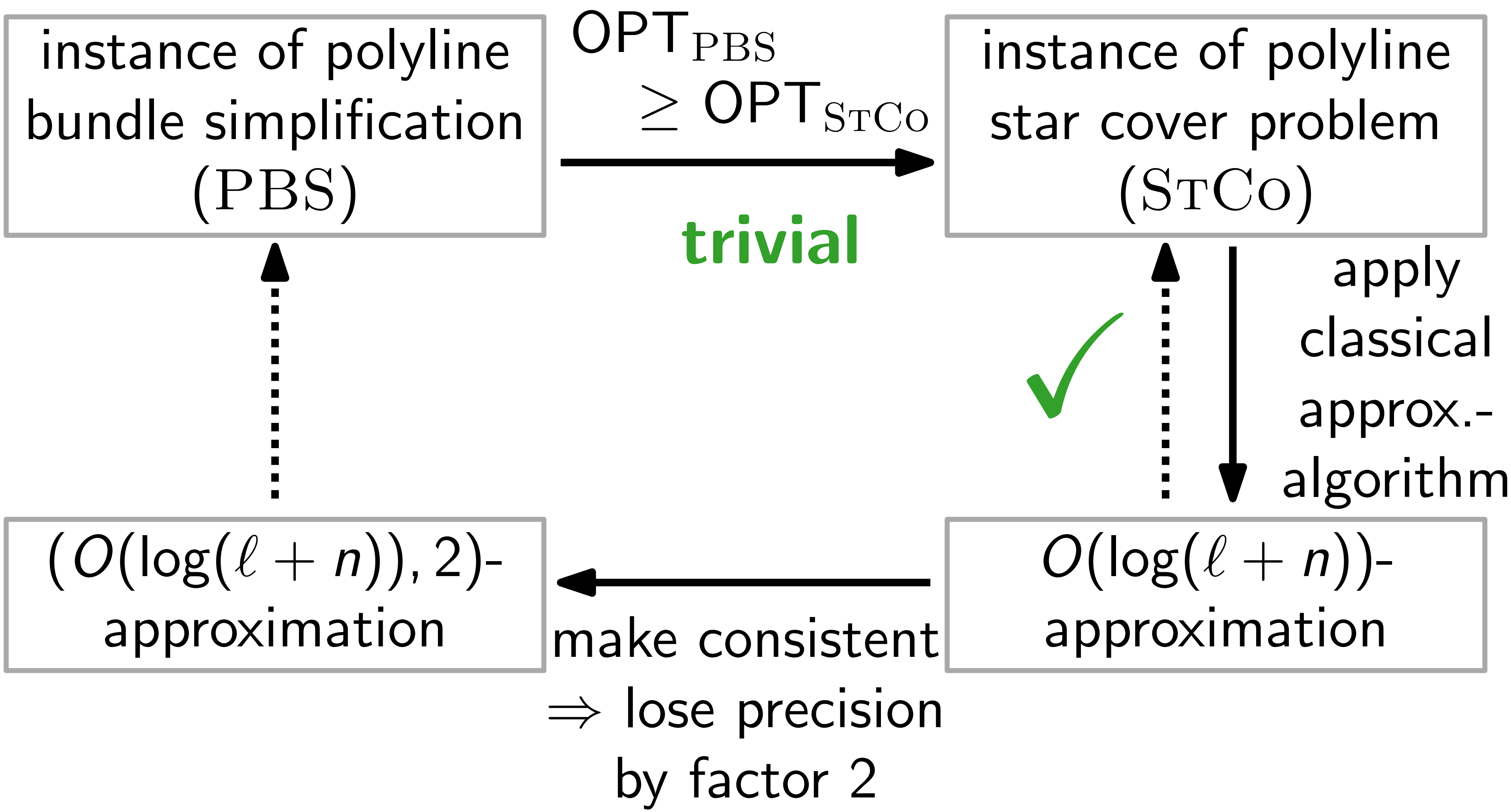


StCo solution $\Longrightarrow$ PBS solution

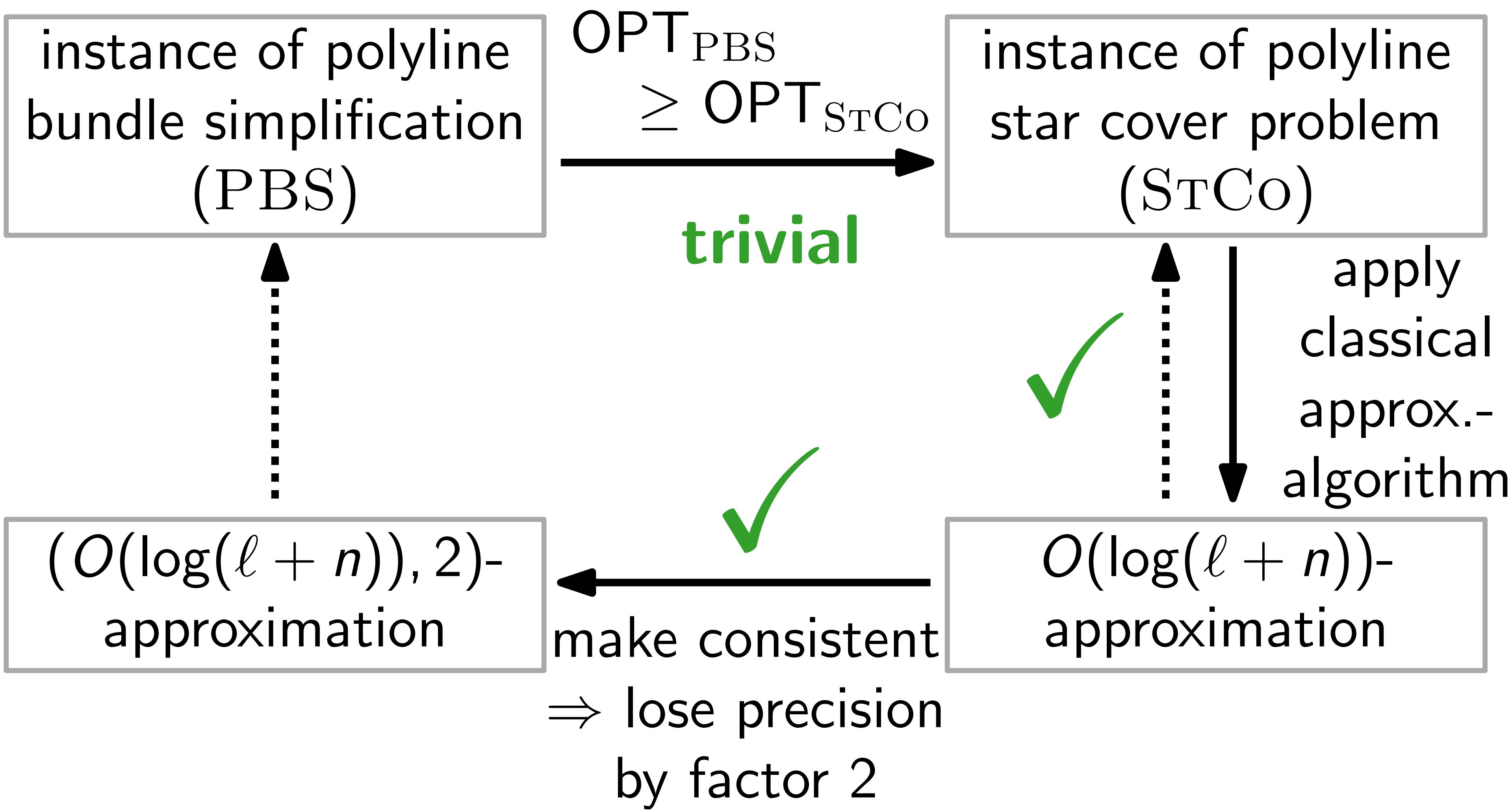


Lemma: [Agarwal et al.]
If $d_{\text{Fréchet}}((b_y, b_z), (b_y, \ldots, b_i, \ldots, b_j, \ldots, b_z)) \leq \delta$,
then $d_{\text{Fréchet}}((b_i, b_j), (b_i, \ldots, b_j)) \leq 2\delta$
for all $y \leq i < j \leq z$.

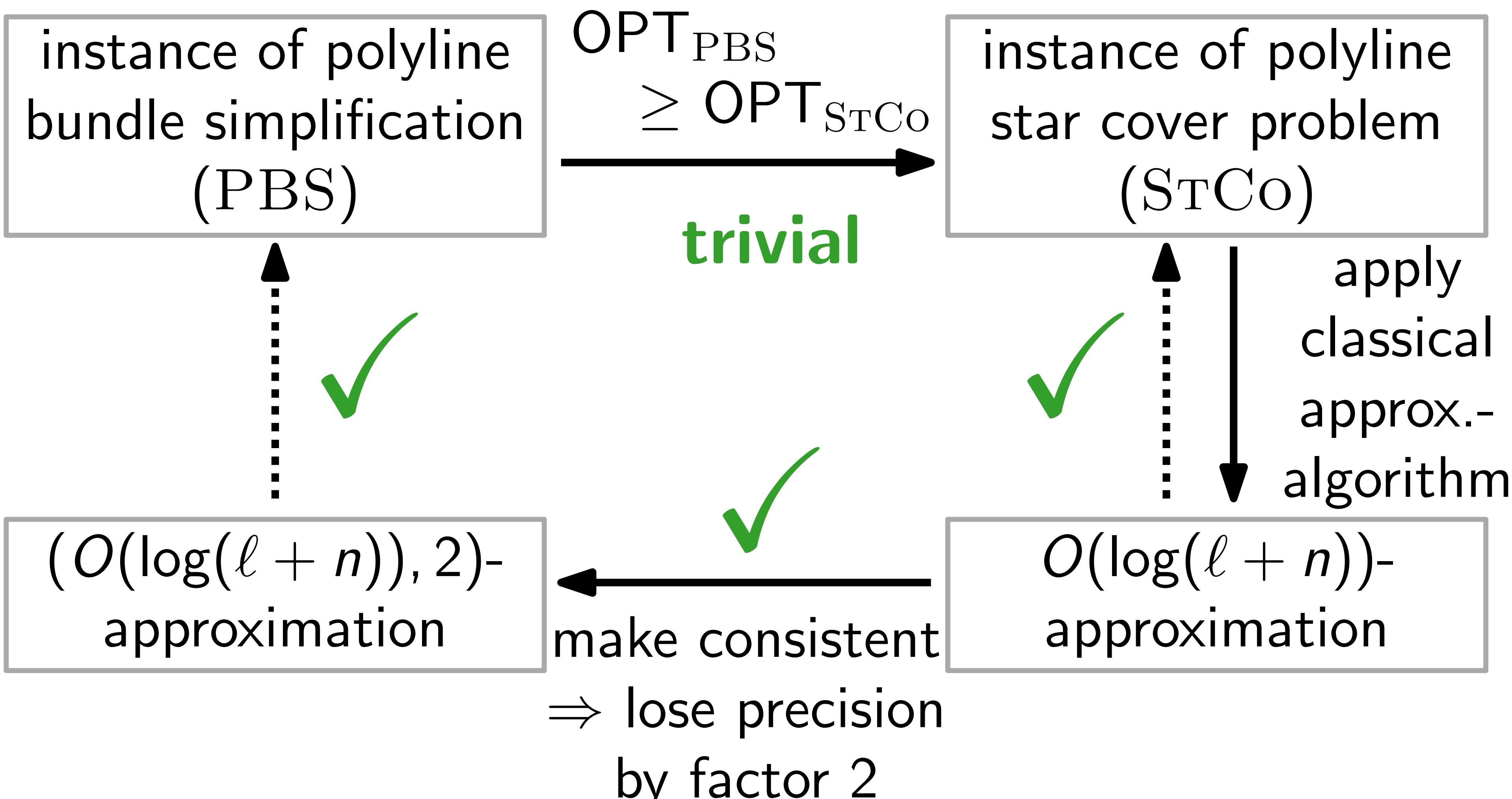
Bi-criteria Approximation



Bi-criteria Approximation



Bi-criteria Approximation



Agenda

1. Motivation and Introduction
2. Problem Definition
3. Hardness of Approximation (+ Proof Sketch)
4. Bi-Criteria Approximation (+ Proof Sketch)
5. **Summary**

Summary

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ
- Can be approximated within a factor of $O(\log(\ell + n))$ when violating the distance threshold by at most a factor of 2

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ
- Can be approximated within a factor of $O(\log(\ell + n))$ when violating the distance threshold by at most a factor of 2
- FPT in the number of shared bends

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ
- Can be approximated within a factor of $O(\log(\ell + n))$ when violating the distance threshold by at most a factor of 2
- FPT in the number of shared bends

Future research?

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ
- Can be approximated within a factor of $O(\log(\ell + n))$ when violating the distance threshold by at most a factor of 2
- FPT in the number of shared bends
- Sharpen our bounds

Future research?

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ
- Can be approximated within a factor of $O(\log(\ell + n))$ when violating the distance threshold by at most a factor of 2
- FPT in the number of shared bends

Future research?

- Sharpen our bounds
- Consider alternative goals (e.g. min. $\#$ segments) and distance measurements (e.g. Hausdorff distance)

Summary

Problem:
Simplify a set of polylines sharing bends & segments consistently

- Generalizes the well-known problem of simplifying a polyline
- Becomes NP-hard to approximate within a factor of $n^{1/3-\varepsilon}$
- Not FPT in the number of polylines ℓ
- Can be approximated within a factor of $O(\log(\ell + n))$ when violating the distance threshold by at most a factor of 2
- FPT in the number of shared bends

Future research?

- Sharpen our bounds
- Consider alternative goals (e.g. min. # segments) and distance measurements (e.g. Hausdorff distance)
- Simplify bundles of polylines in practical applications