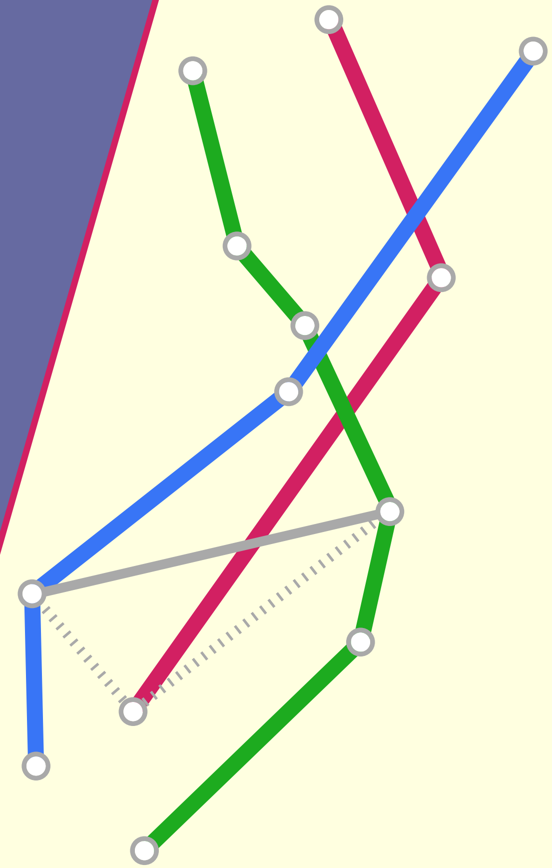
Fine-Grained Analysis of Problems on Curves

Kevin Buchin
Maike Buchin
Maximilian Konzack
Wolfgang Mulzer
André Schulz

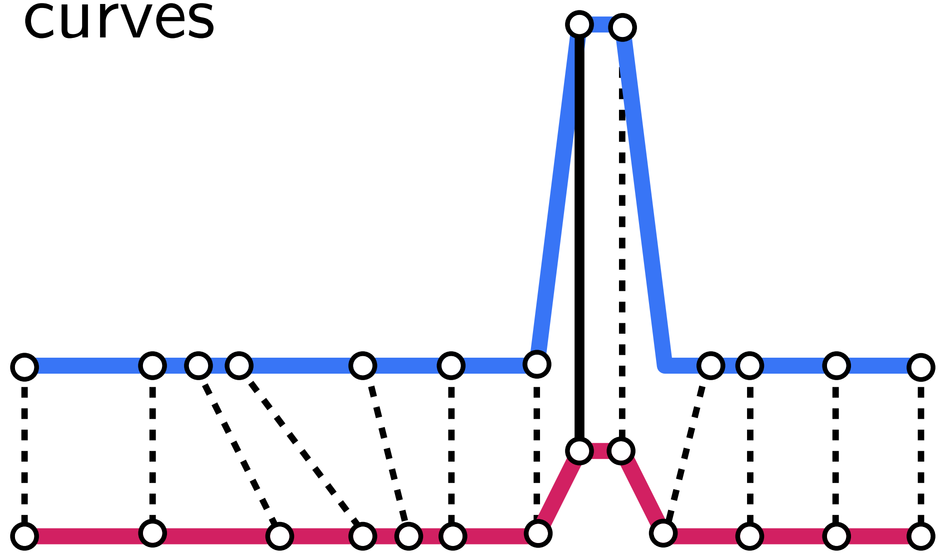


TU/e Technische Universiteit **Eindhoven** University of Technology

Fréchet Distance between two curves

Fréchet distance [Alt and Godau, 1995]

Minimize the maximal distance between curves P and Q



Upper bound [Agarwal et al., 2014]

Running time $O(\frac{mn \log \log n}{\log n})$ for the discrete Fréchet distance

Lower bound [Bringmann, 2014]

Discrete Fréchet distance cannot be computed in $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$ unless the strong exponential time hypothesis fails

Fréchet Distance between k curves

How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

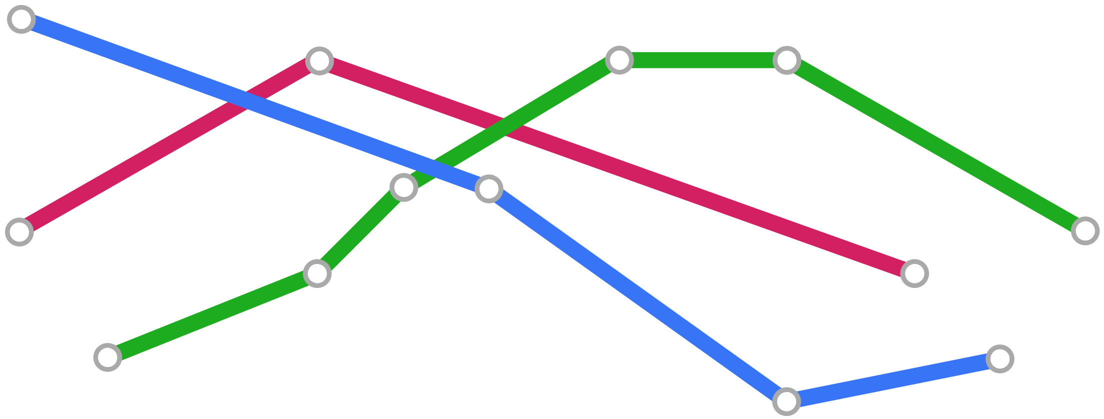
$$\mathcal{C}_1 = (0, 0, \ldots, 0)$$

$$\mathcal{C}_m = (n_1, n_2, \ldots, n_k)$$

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$$

or

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$$



Fréchet Distance between k curves

How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

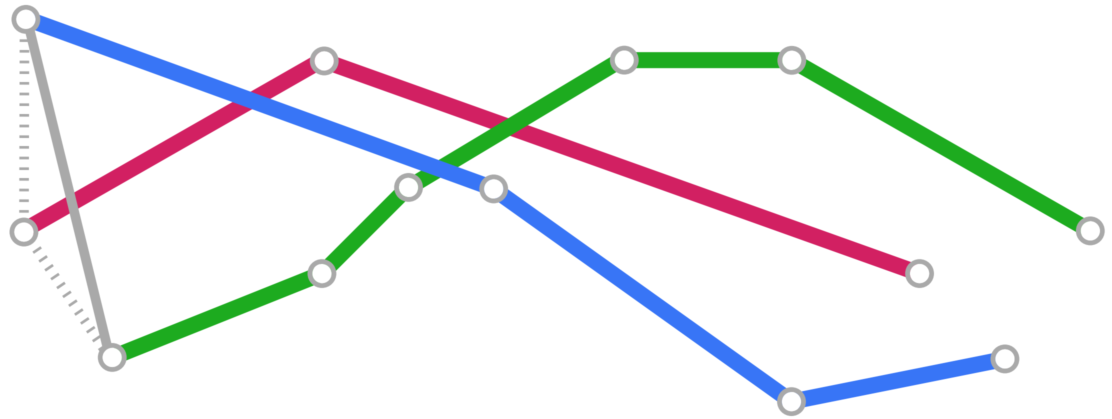
$\mathcal{C}_1 = (0, 0, \ldots, 0)$

$\mathcal{C}_m = (n_1, n_2, \ldots, n_k)$

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$

or

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$



Fréchet Distance between k curves

How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \dots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

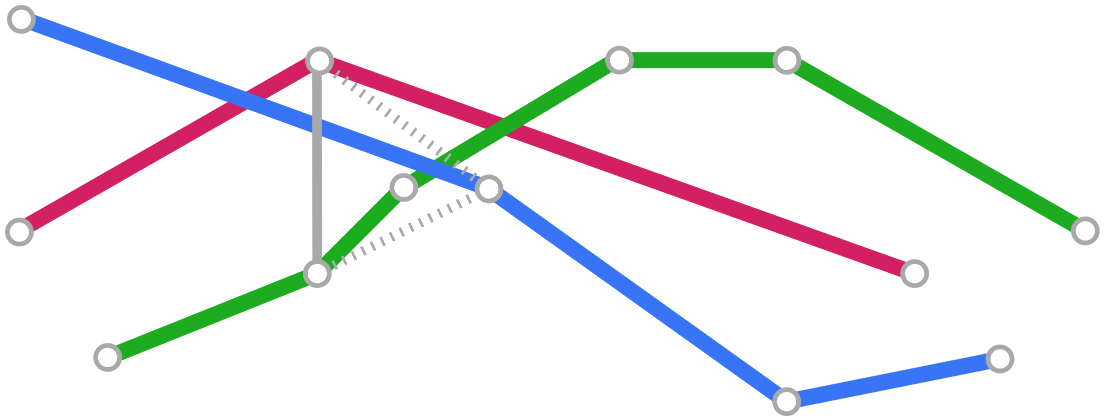
$$\mathcal{C}_1 = (0, 0, \dots, 0)$$

$$\mathcal{C}_m = (n_1, n_2, \dots, n_k)$$

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$$

or

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$$



How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \dots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

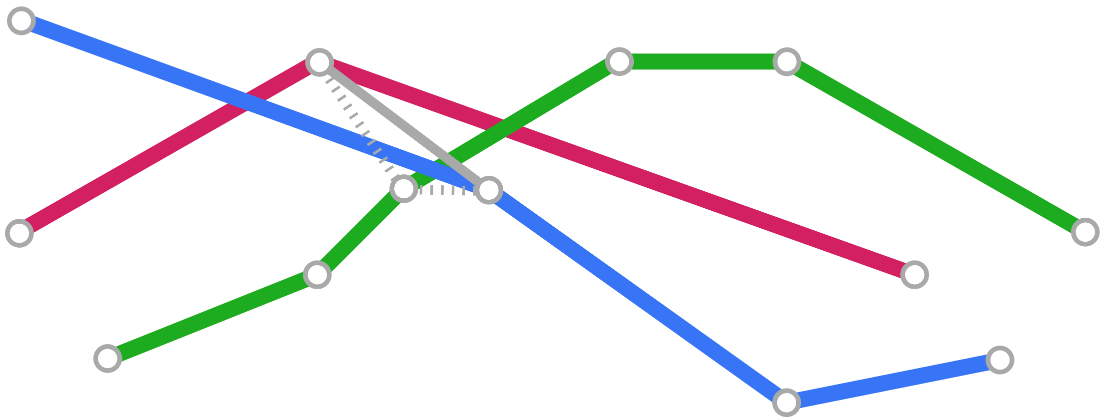
$\mathcal{C}_1 = (0, 0, \dots, 0)$

$\mathcal{C}_m = (n_1, n_2, \dots, n_k)$

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$

or

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$



Fréchet Distance between k curves

How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

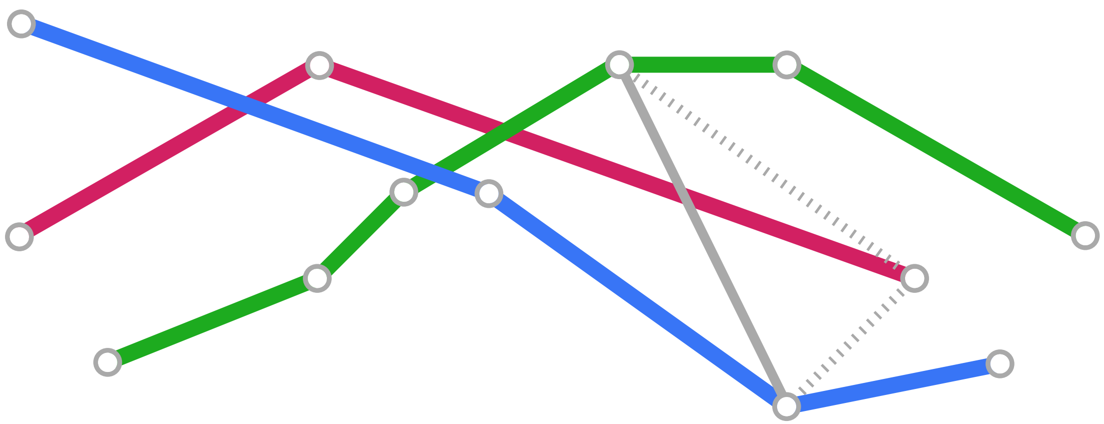
$\mathcal{C}_1 = (0, 0, \ldots, 0)$

$\mathcal{C}_m = (n_1, n_2, \ldots, n_k)$

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$

or

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$



Fréchet Distance between k curves

TU/e Technische Universiteit Eindhoven University of Technology

How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

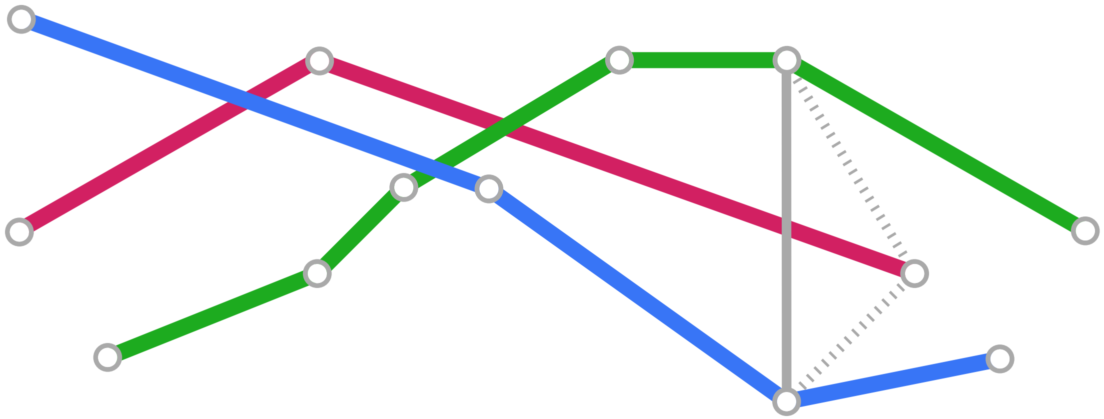
$$\mathcal{C}_1 = (0, 0, \ldots, 0)$$

$$\mathcal{C}_m = (n_1, n_2, \ldots, n_k)$$

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$$

or

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$$



Fréchet Distance between k curves

TU/e Technische Universiteit Eindhoven University of Technology

How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

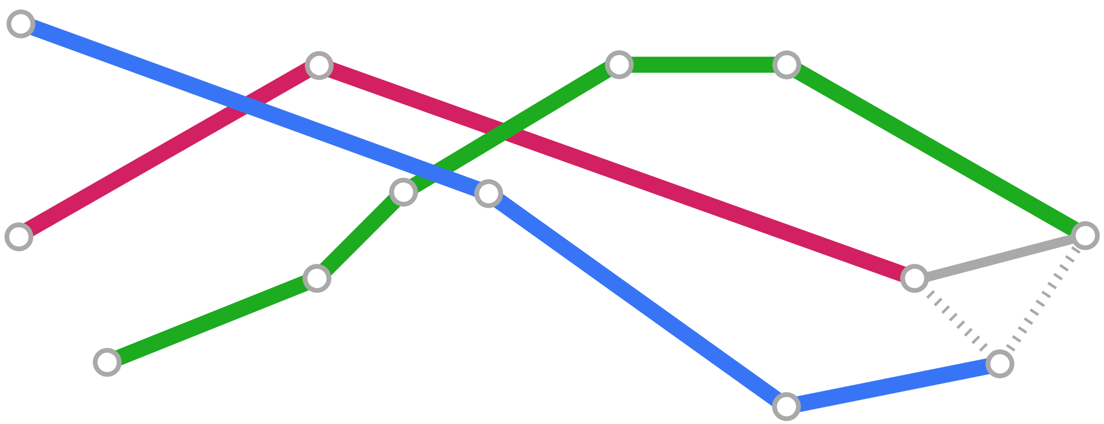
$$\mathcal{C}_1 = (0, 0, \ldots, 0)$$

$$\mathcal{C}_m = (n_1, n_2, \ldots, n_k)$$

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$$

or

$$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$$



How can we capture distances on a tuple of points?

An alignment $\mathcal{C} = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$ of the curves A_1, A_2, A_3

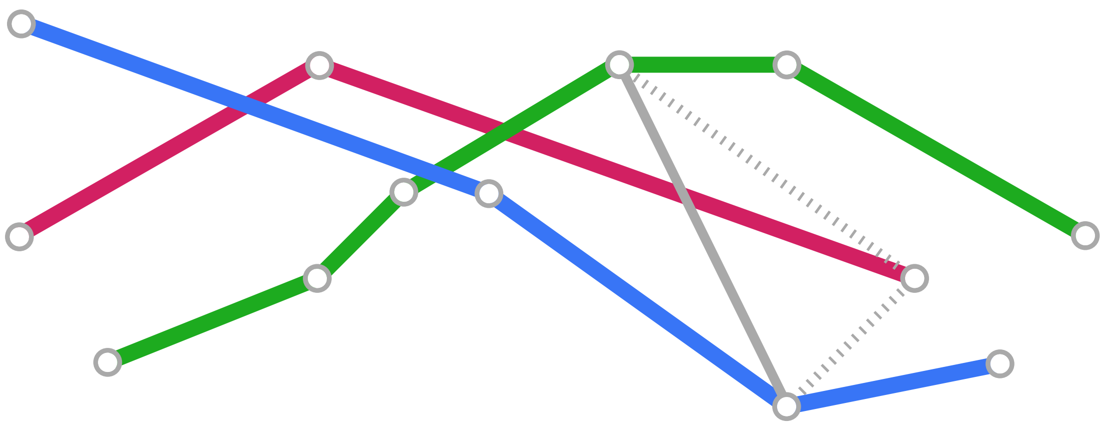
$\mathcal{C}_1 = (0, 0, \ldots, 0)$

$\mathcal{C}_m = (n_1, n_2, \ldots, n_k)$

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h]$

or

$\mathcal{C}_{i+1}[h] = \mathcal{C}_i[h] + 1$

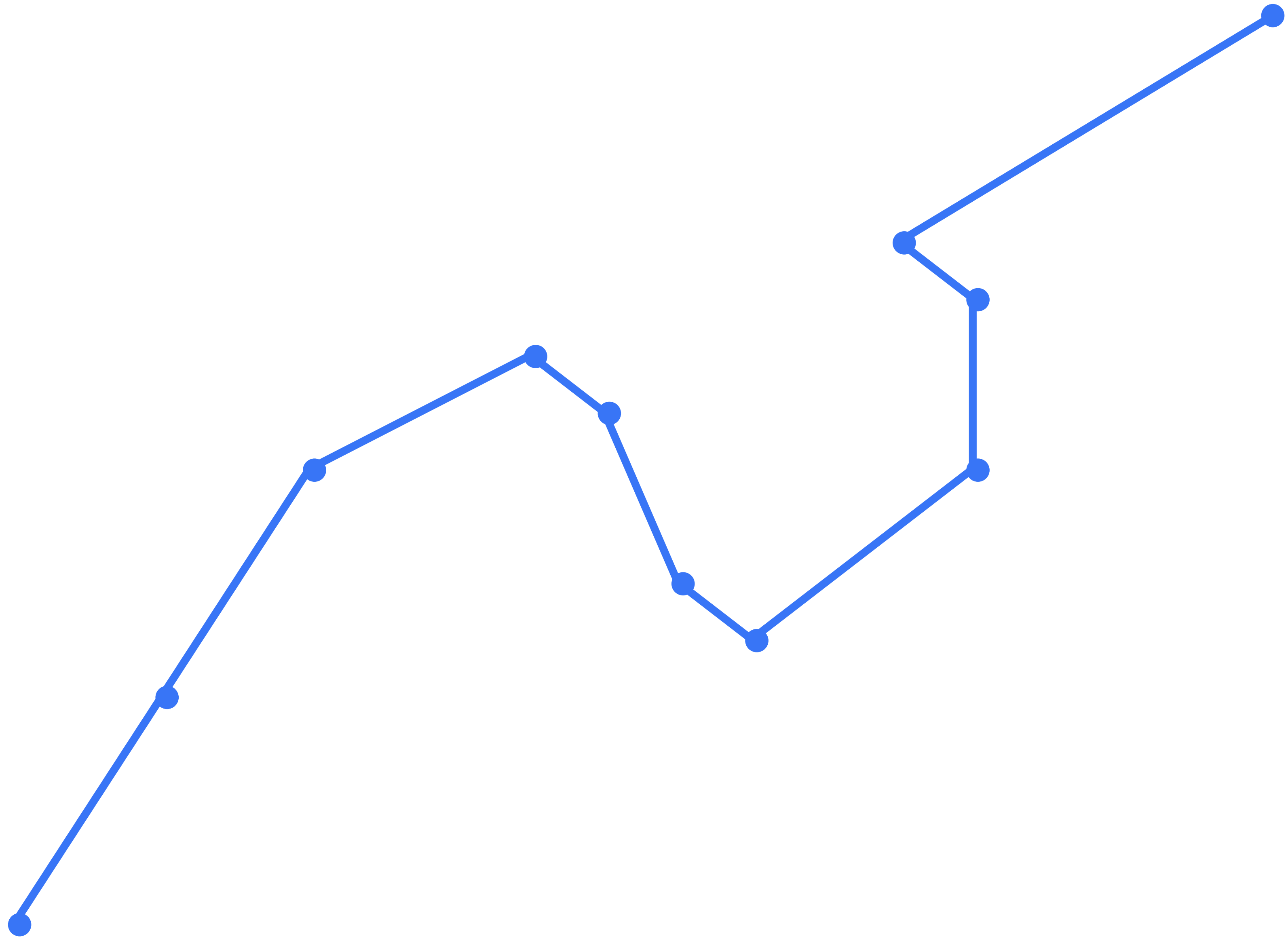


Discrete Fréchet distance: minimize distance over all coupled distances $d_{\mathcal{C}}$

Upper bound [Dumitrescu and Rote, 2004]

Running time $O(n^k)$ for k polygonal curves

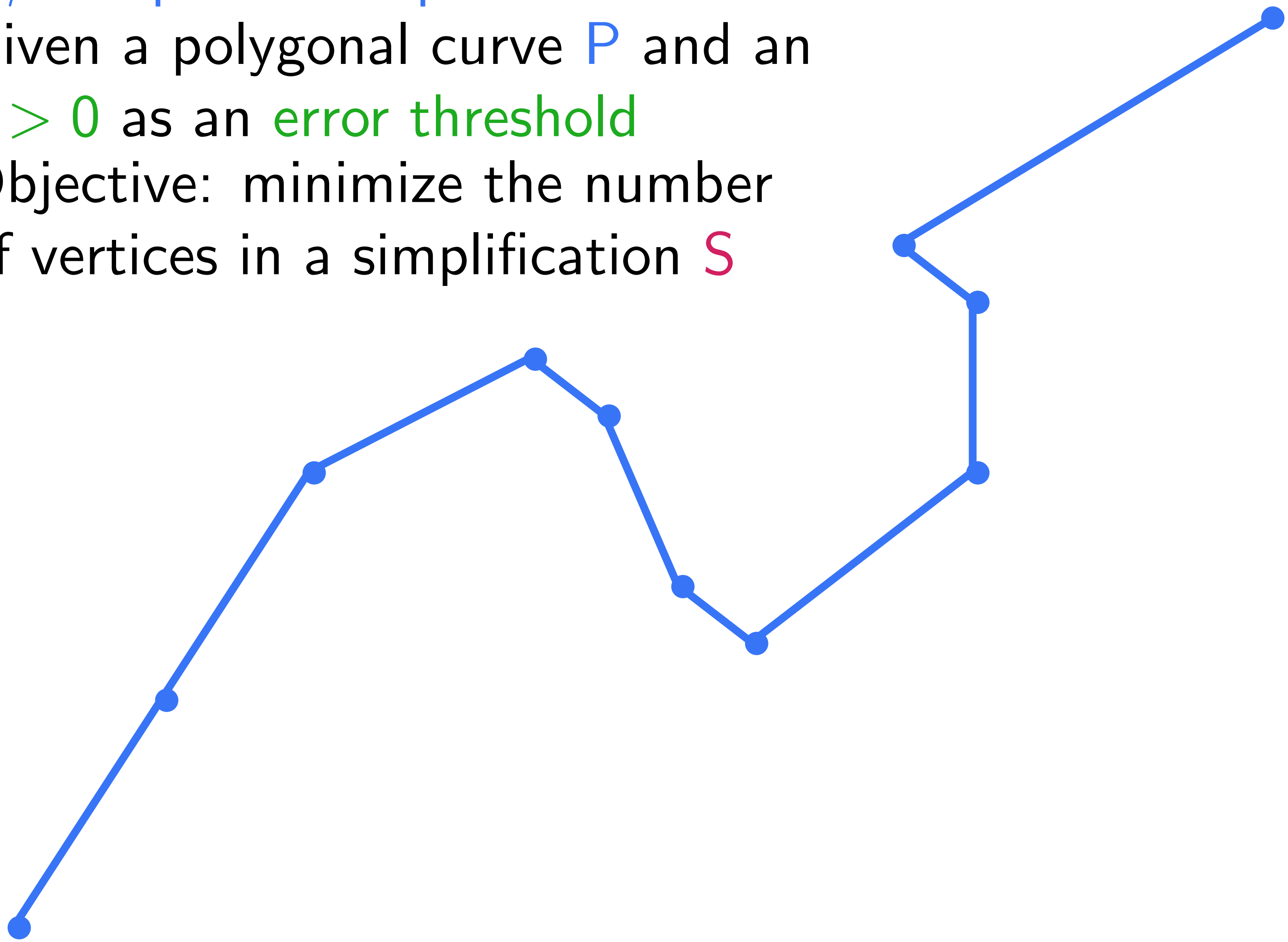
Curve Simplification



Curve Simplification

min-# Simplification problem:

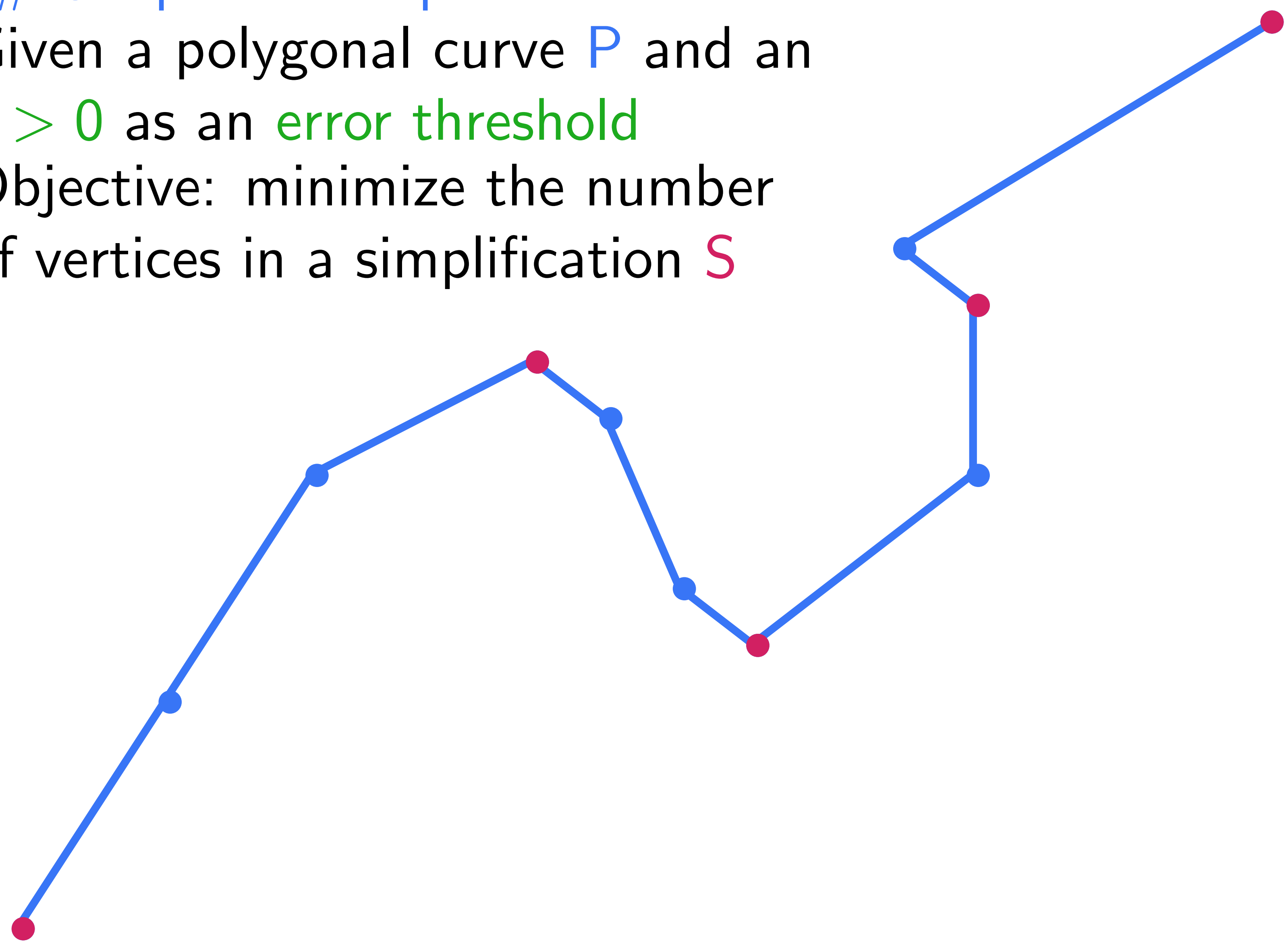
- Given a polygonal curve P and an $\varepsilon > 0$ as an error threshold
- Objective: minimize the number of vertices in a simplification S



Curve Simplification

min-# Simplification problem:

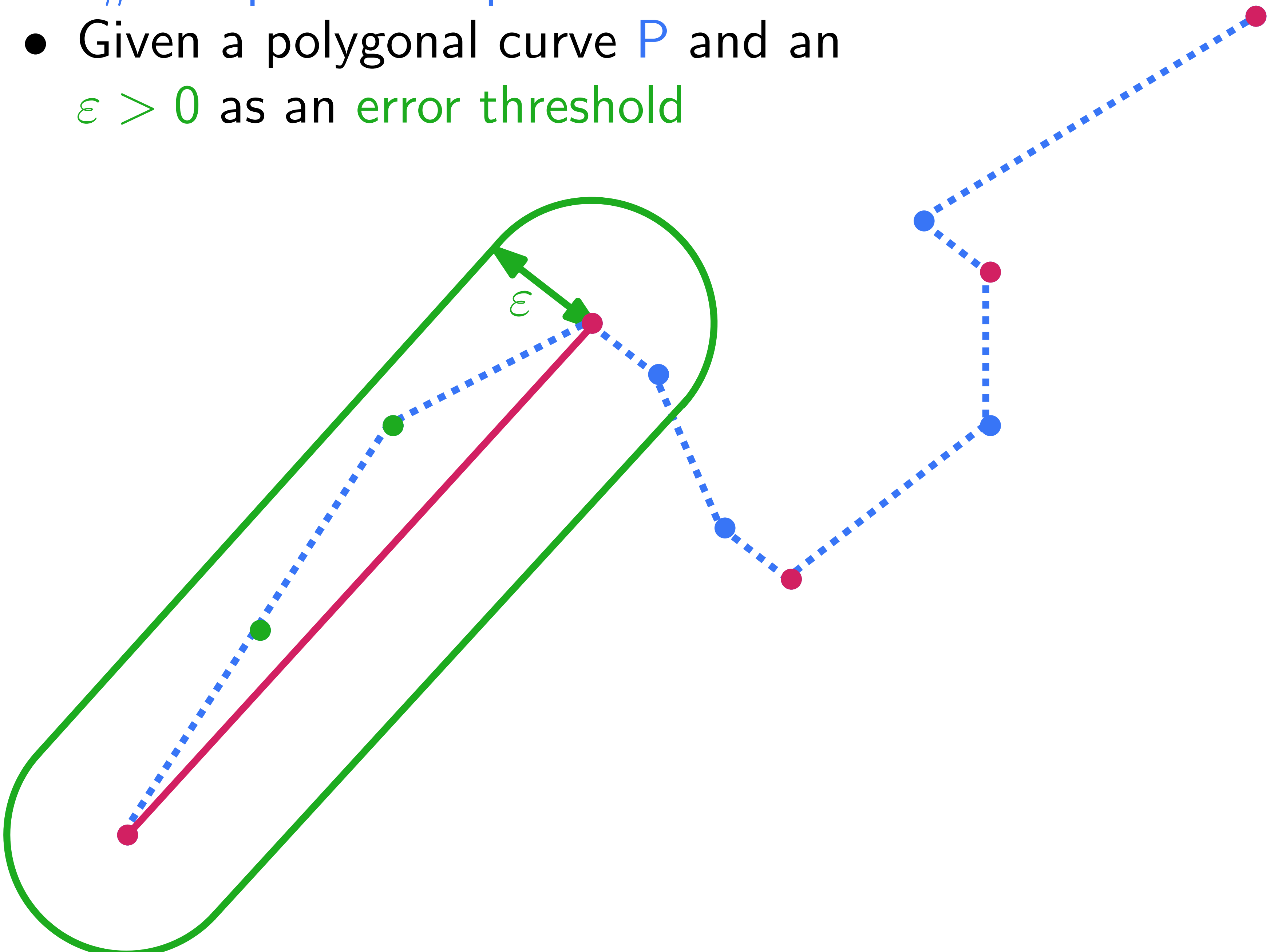
- Given a polygonal curve P and an $\varepsilon > 0$ as an error threshold
- Objective: minimize the number of vertices in a simplification S



Curve Simplification

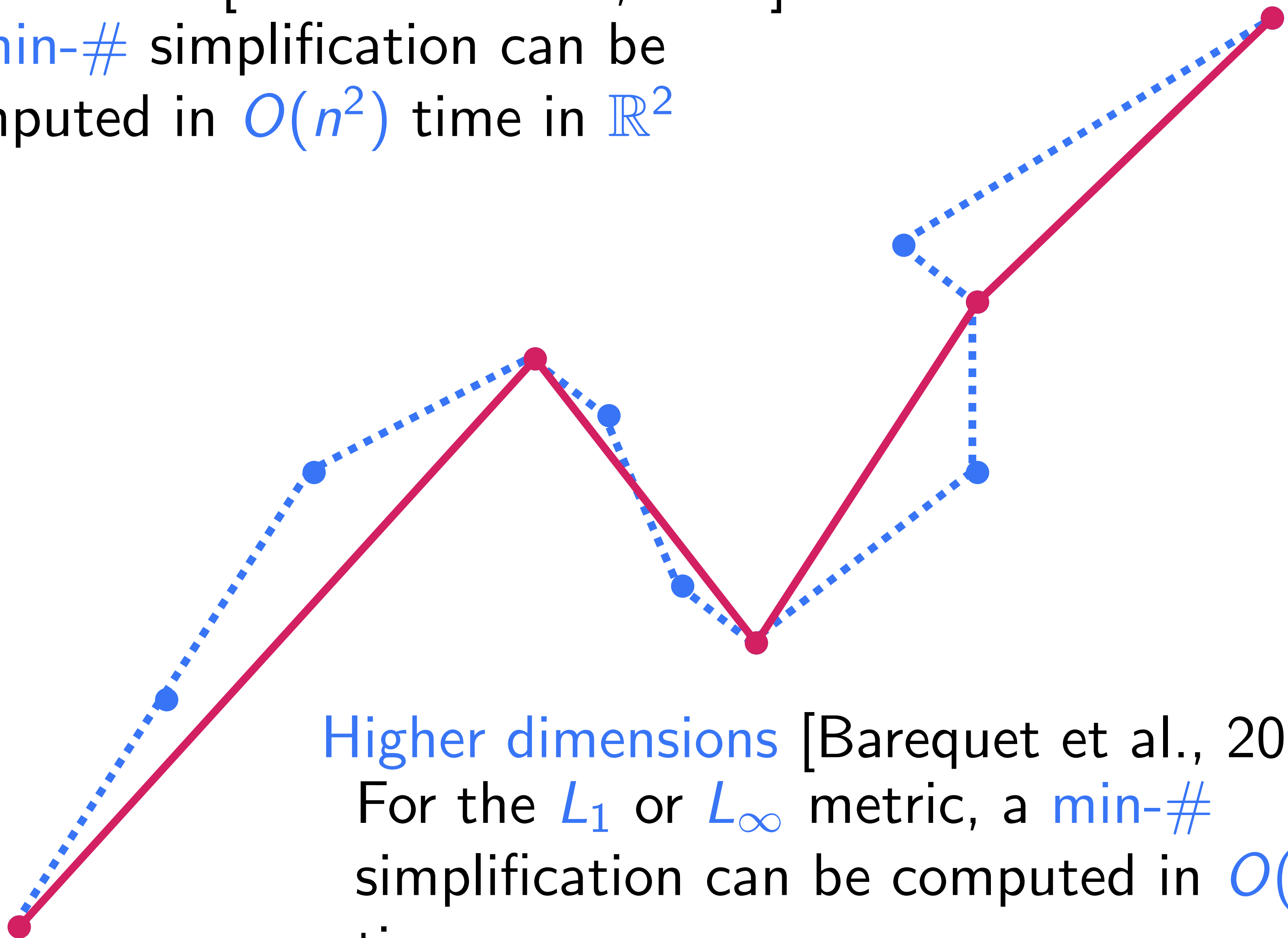
min-# Simplification problem:

- Given a polygonal curve P and an $\varepsilon > 0$ as an error threshold



Curve Simplification

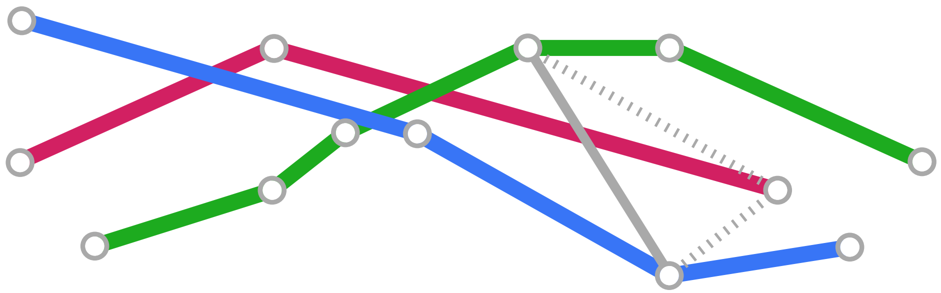
Upper bound [Chan and Chin, 1996]
A min-# simplification can be computed in $O(n^2)$ time in $\mathbb{R}^2$



Higher dimensions [Barequet et al., 2002]
For the L_1 or L_∞ metric, a min-# simplification can be computed in $O(n^2)$ time

1. Result

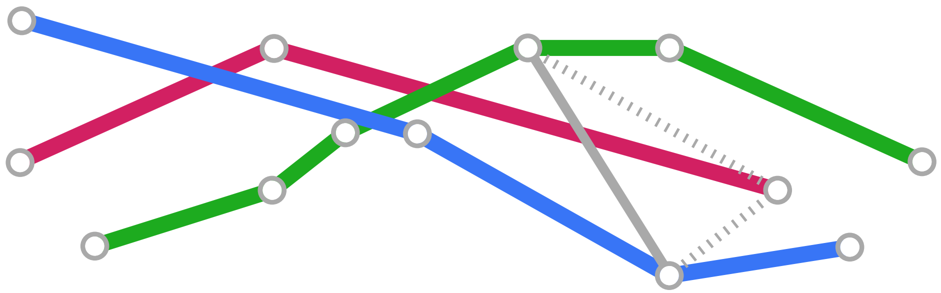
- No $O(n^{k-\varepsilon})$ time algorithm for the discrete Fréchet distance on k curves for any $\varepsilon > 0$ unless SETH fails



Lower Bounds

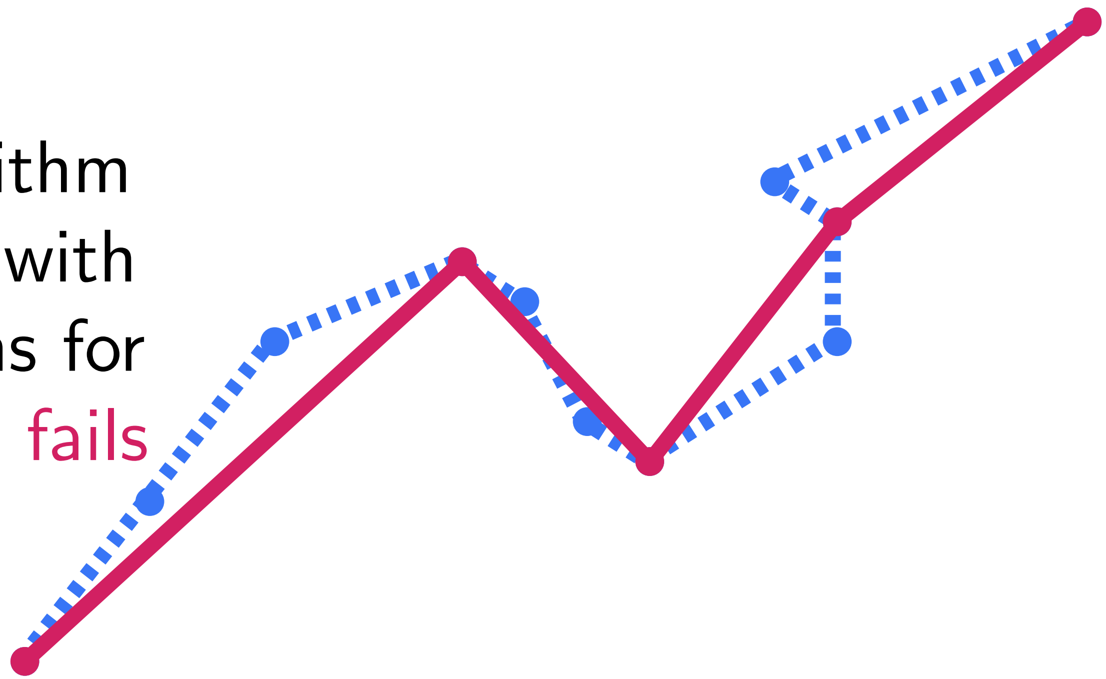
1. Result

- No $O(n^{k-\varepsilon})$ time algorithm for the discrete Fréchet distance on k curves for any $\varepsilon > 0$ unless SETH fails



2. Result

- No $O(n^{2-\varepsilon})$ time algorithm for curve simplification with $d = \Omega(\log n)$ dimensions for any $\varepsilon > 0$ unless SETH fails



Orthogonal Vectors

Proof idea

- Transform k Orthogonal Vectors to curves
  1. Gadgets for coordinates
  2. Synchronized walk of the composite curves
- Fréchet distance $d_F(\cdot) \leq 1$ iff. vectors are orthogonal

Orthogonal Vectors

TU/e Technische Universiteit Eindhoven University of Technology

Proof idea

- Transform k Orthogonal Vectors to curves
  1. Gadgets for coordinates
  2. Synchronized walk of the composite curves
- Fréchet distance $d_F(\cdot) \leq 1$ iff. vectors are orthogonal

$$\begin{pmatrix} \alpha^1 \\ \alpha^2 \\ \vdots \\ \alpha^k \end{pmatrix} = \begin{pmatrix} 0 & 1 & \dots & 0 & 1 \\ 1 & 0 & \dots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \dots & 1 & 1 \end{pmatrix}$$

$\alpha^1, \alpha^2, \dots, \alpha^k$ are non-orthogonal

Orthogonal Vectors

$$\begin{pmatrix} \alpha^1 \\ \alpha^2 \\ \vdots \\ \alpha^k \end{pmatrix} = \begin{pmatrix} 0 & 1 & \dots & 0 & 1 \\ 1 & 0 & \dots & 1 & \color{red}{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \dots & 1 & 1 \end{pmatrix}$$

$\alpha^1, \alpha^2, \dots, \alpha^k$ are orthogonal

Orthogonal Vectors

k Orthogonal Vectors

- Given k $\{0,1\}^d$ vectors $\alpha^1, \alpha^2, \dots, \alpha^k$
- Do they satisfy

$$\sum_{h=1}^{d} \prod_{t \in [k]} \alpha^t[h] = 0?$$

$$\begin{pmatrix} \alpha^1 \\ \alpha^2 \\ \vdots \\ \alpha^k \end{pmatrix} = \begin{pmatrix} 0 & 1 & \dots & 0 & 1 \\ 1 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \dots & 1 & 1 \end{pmatrix}$$

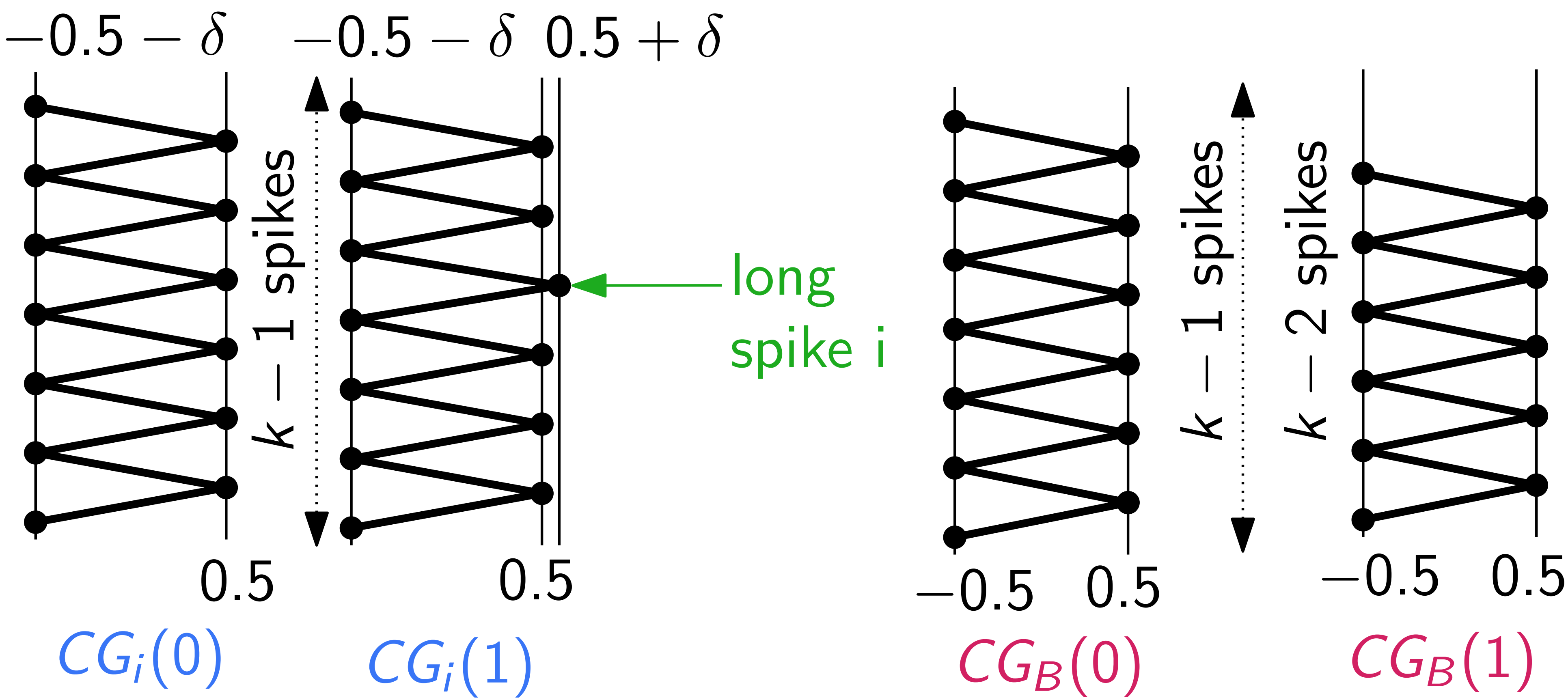
$\alpha^1, \alpha^2, \dots, \alpha^k$ are orthogonal

Strong Exponential Time Hypothesis (SETH)

- For every ε there is a k such that
- SAT on k-CNF cannot be solved in subexponential time

Coordinate Gadget

Encoding coordinates from k-Orthogonal Vectors
$A_1, A_2, \ldots A_{k-1}$ and B by curves



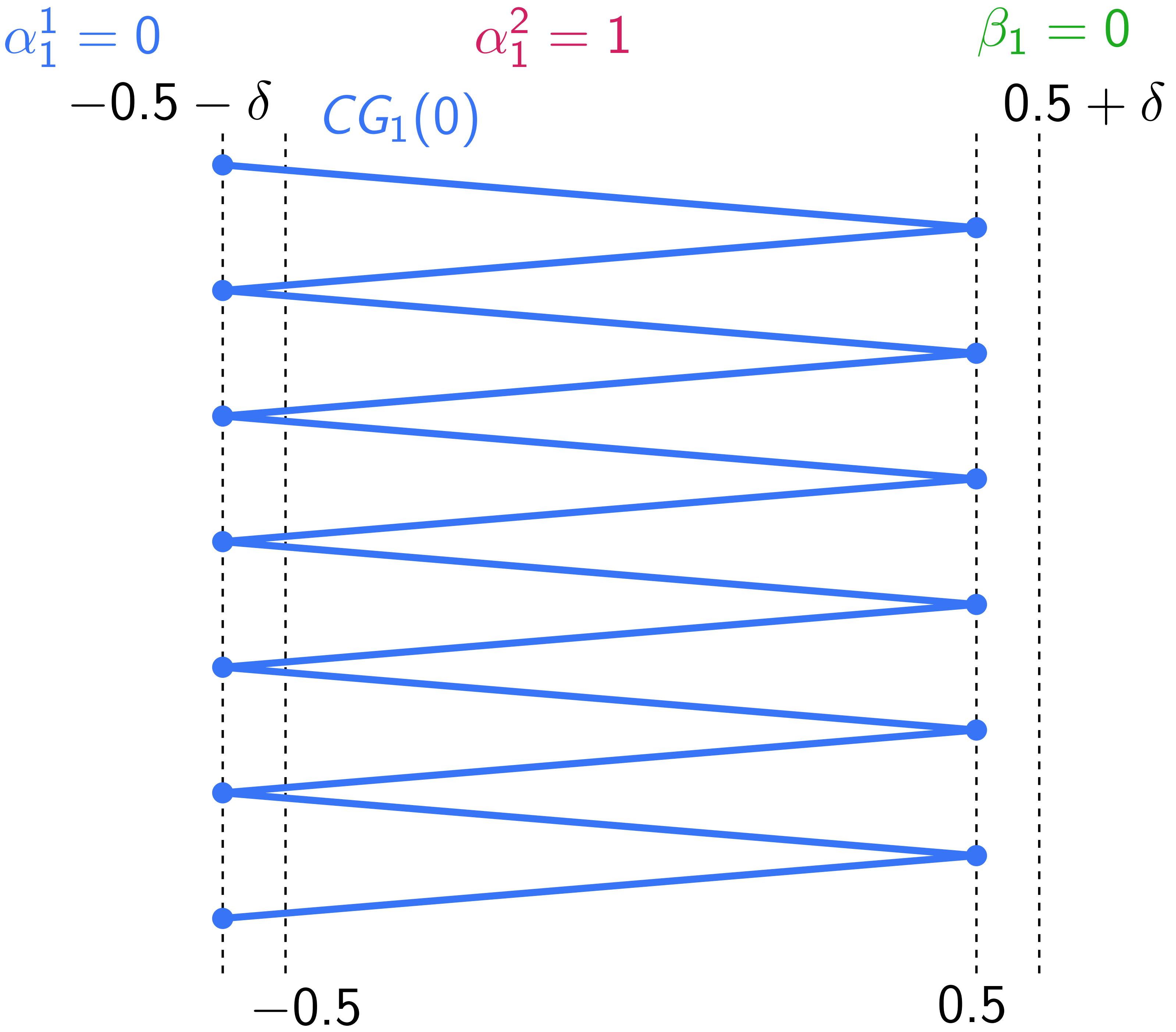
Example for Coordinate Gadgets

$\alpha_1^1 = 0$ $\alpha_1^2 = 1$ $\beta_1 = 0$

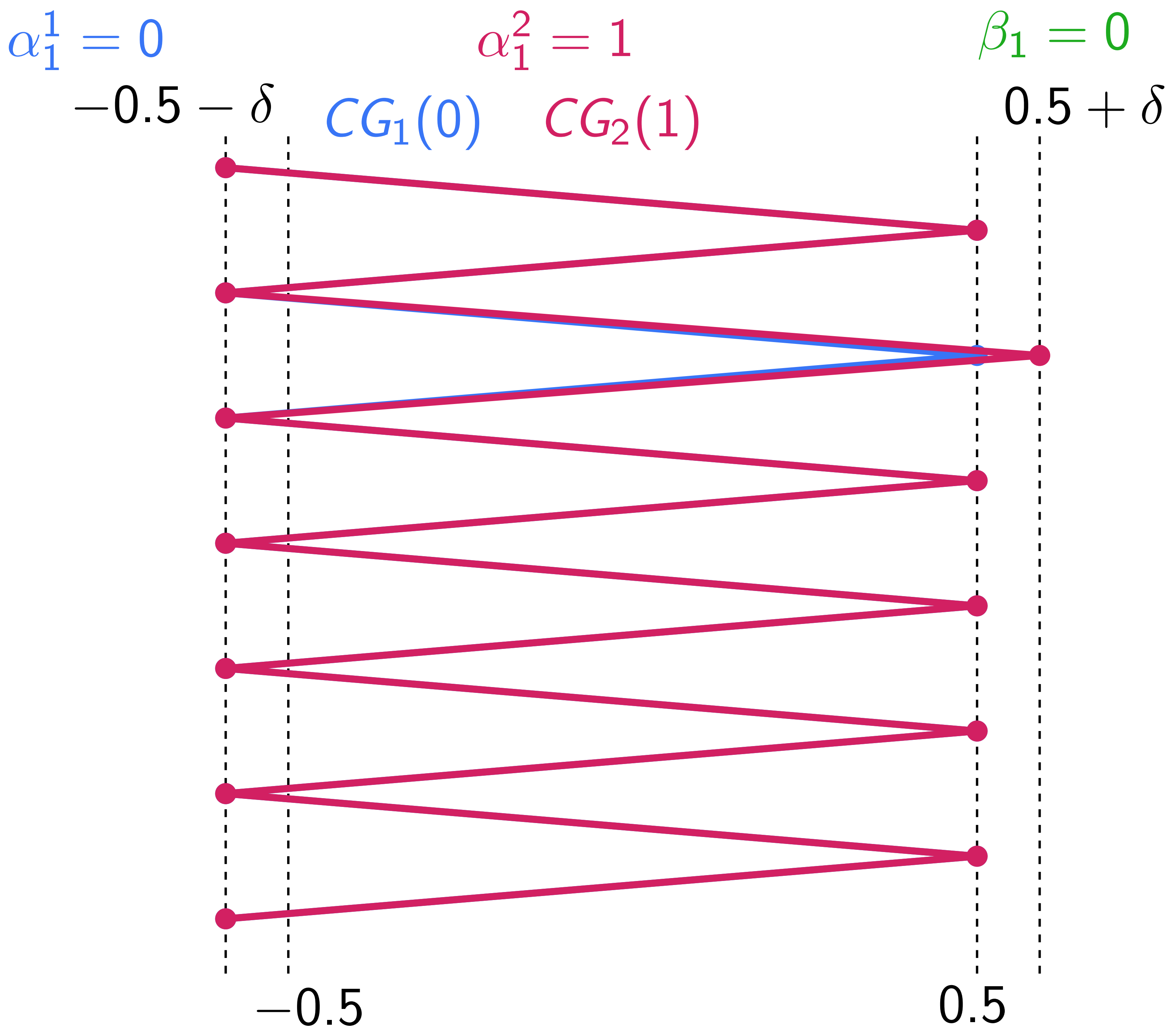
$-0.5 - \delta$ $0.5 + \delta$

-0.5 0.5

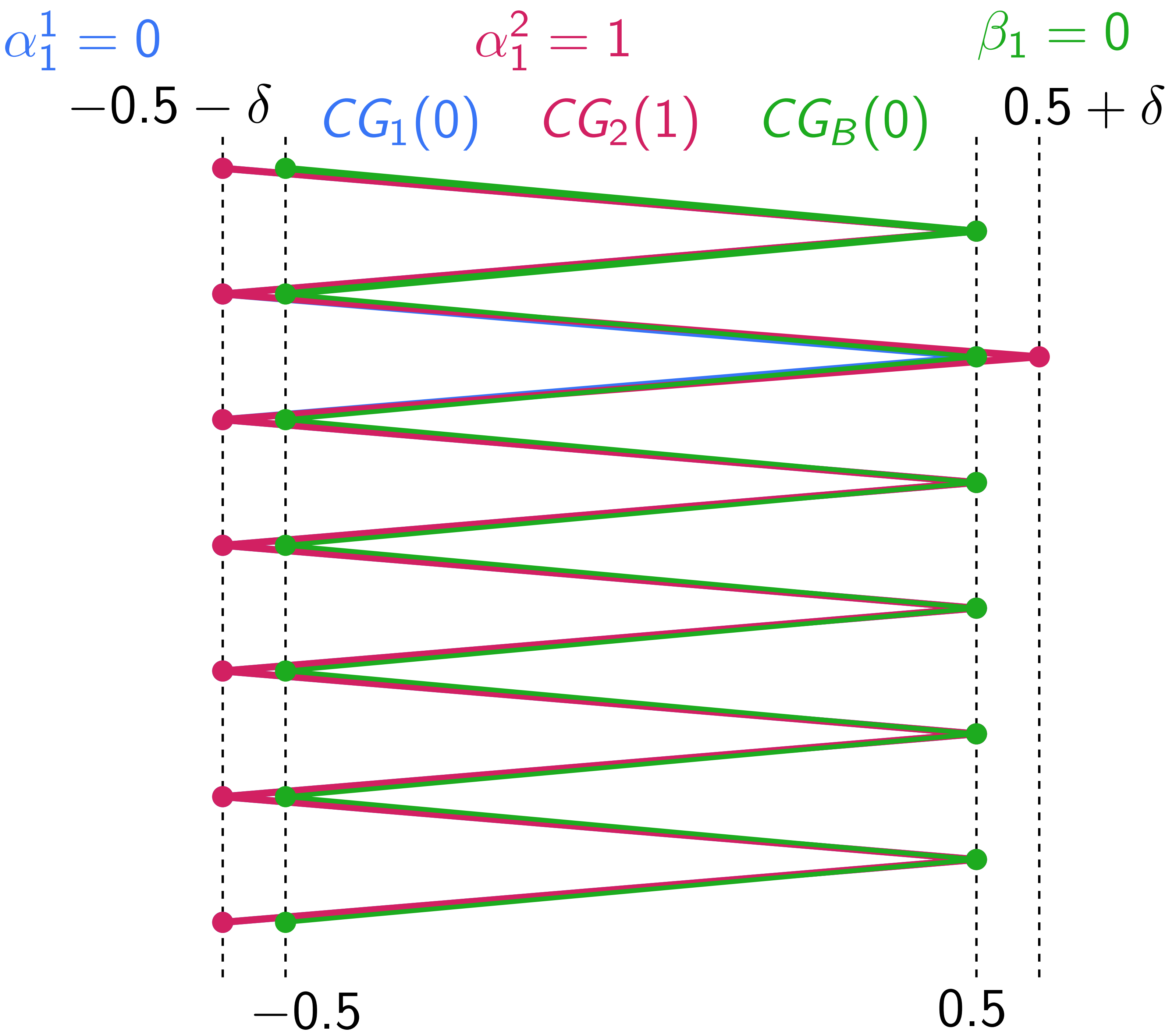
Example for Coordinate Gadgets



Example for Coordinate Gadgets

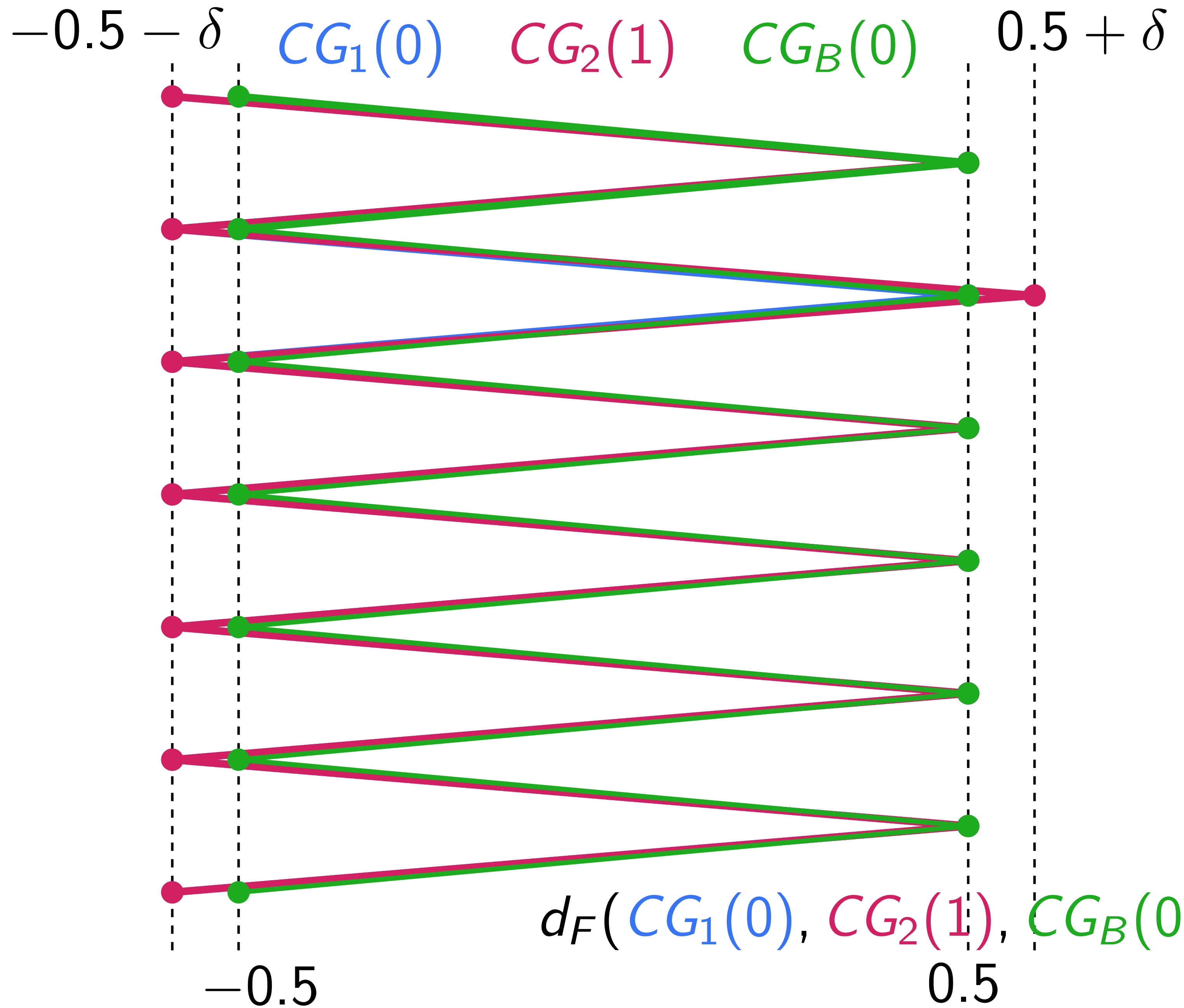


Example for Coordinate Gadgets



Example for Coordinate Gadgets

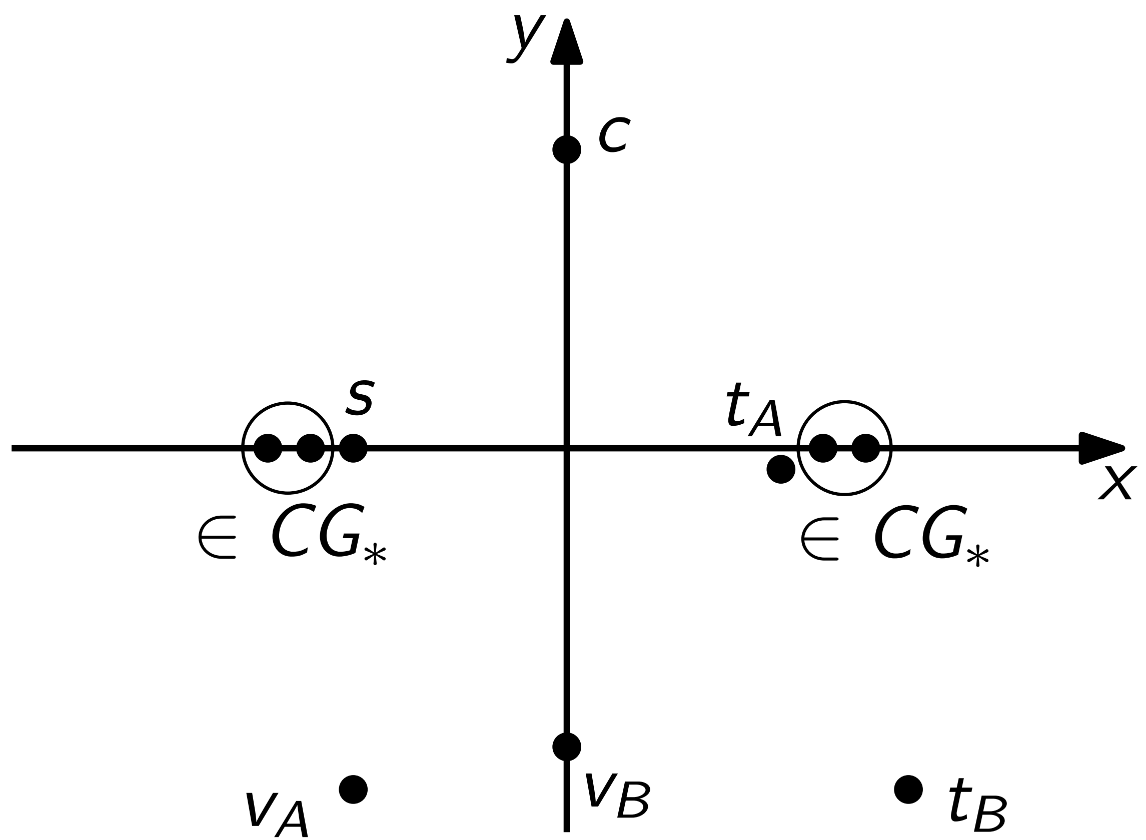
$\alpha_1^1 = 0$ $\alpha_1^2 = 1$ $\beta_1 = 0$



$d_F(CG_1(0), CG_2(1), CG_B(0)) \leq 1$

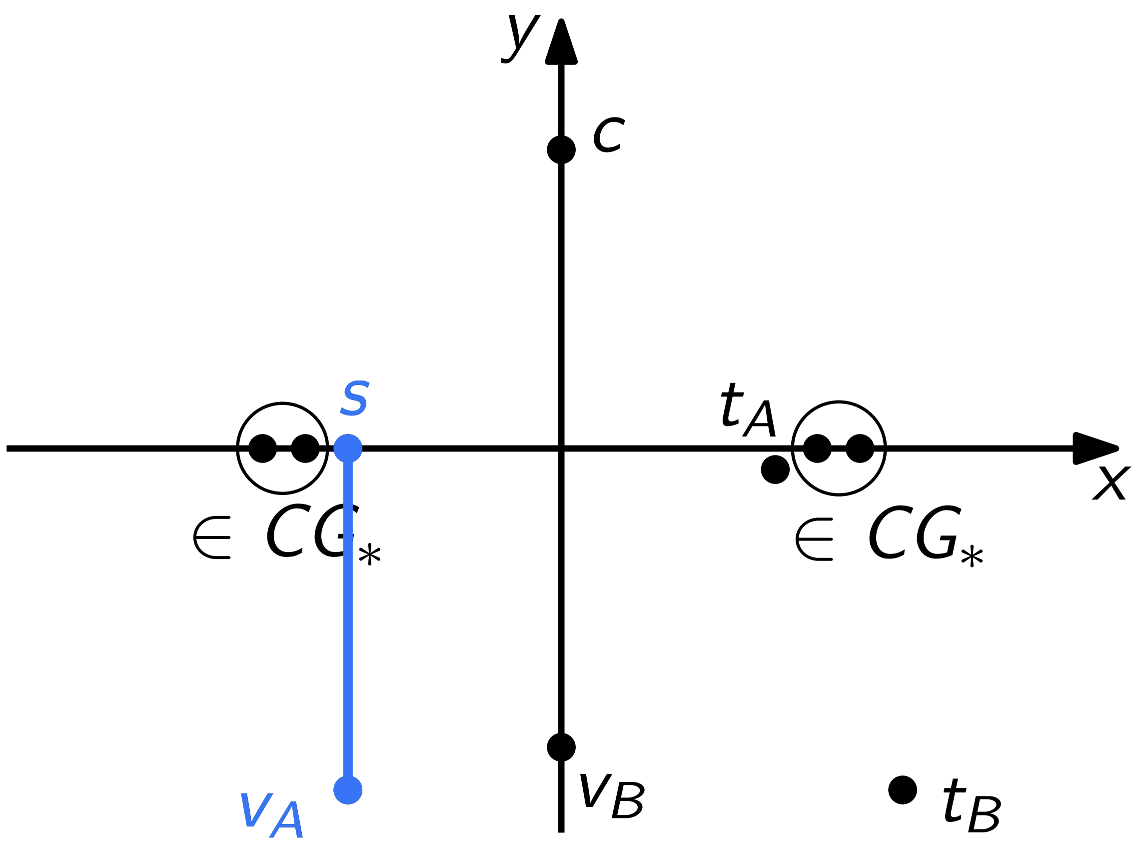
Reduction Construction

Proof idea



Proof idea

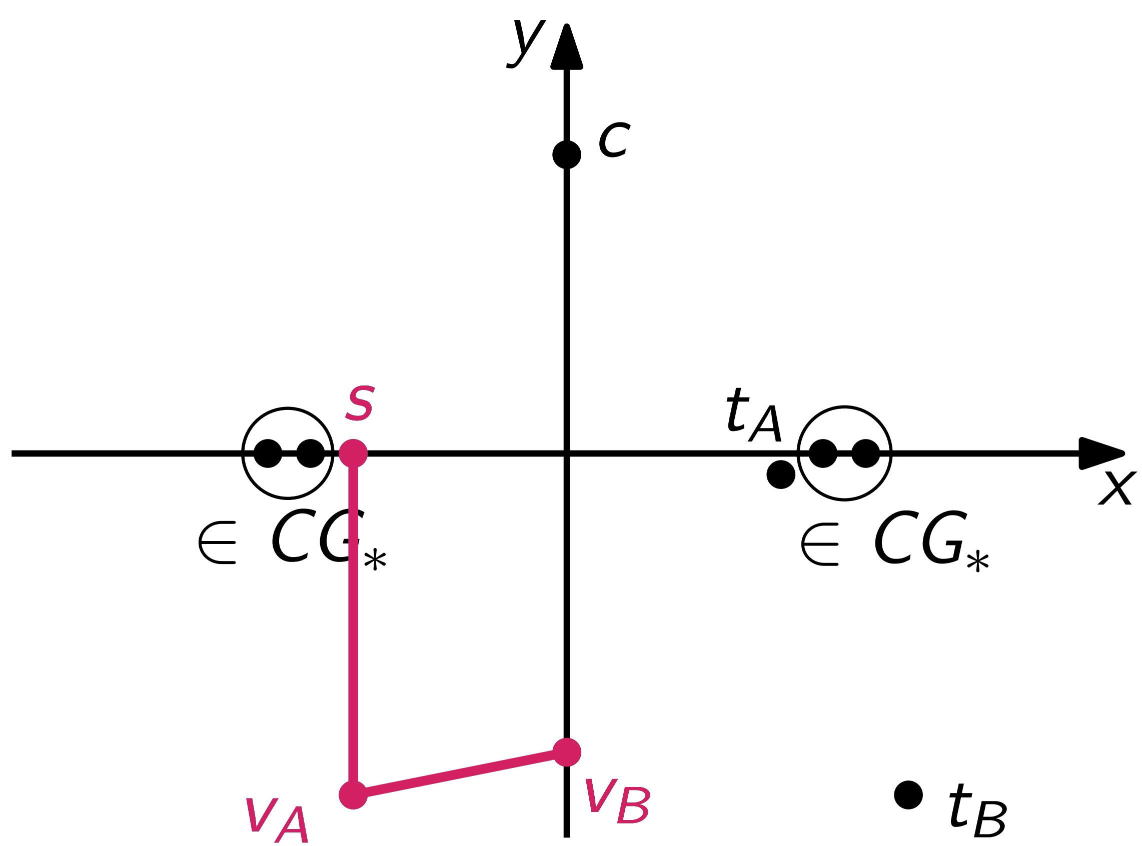
1. Simultaneous walk of curves $A_1, \ldots A_{k-1}$ from s to v_A



Reduction Construction

Proof idea

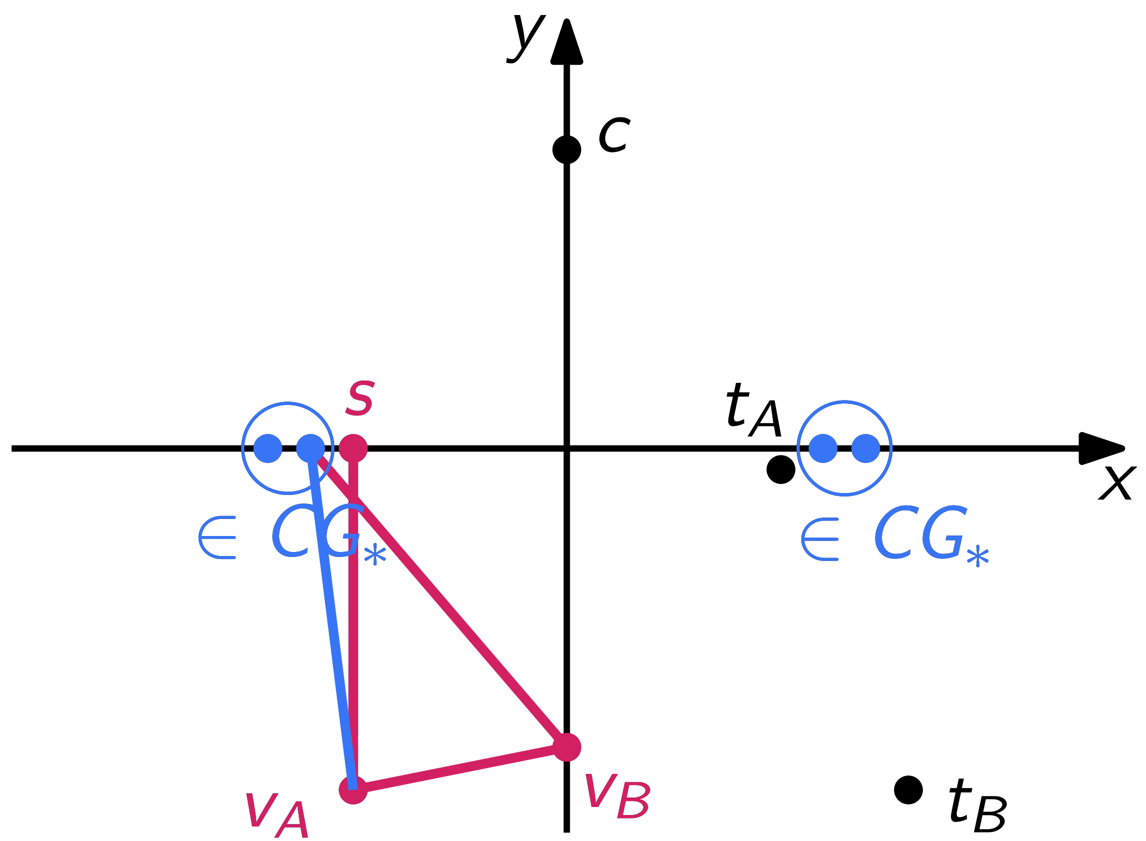
1. Simultaneous walk of curves $A_1, \dots A_{k-1}$ from s to v_A
2. Curve B walks to v_B over s, v_A



Reduction Construction

Proof idea

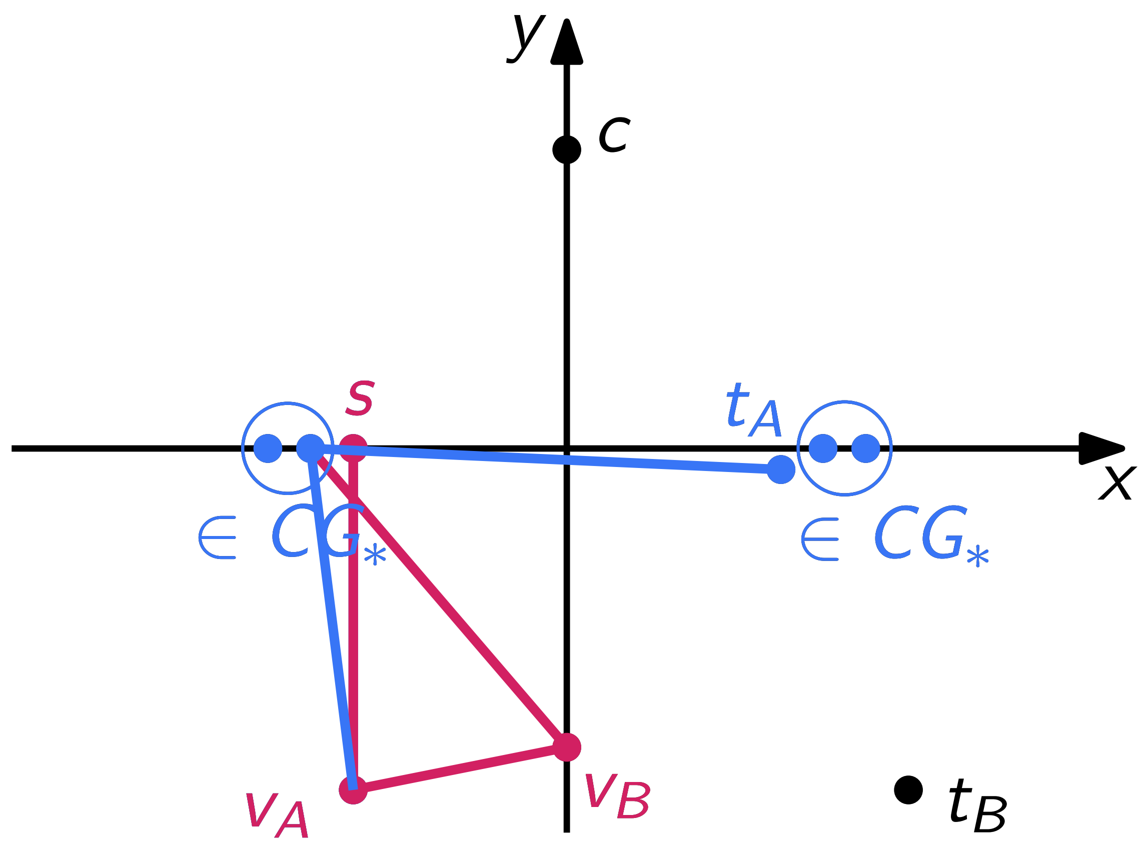
1. Simultaneous walk of curves $A_1, \ldots A_{k-1}$ from s to v_A
2. Curve B walks to v_B over s, v_A
3. Synchronized traversal of all coordinate gadgets CG_*



Reduction Construction

Proof idea

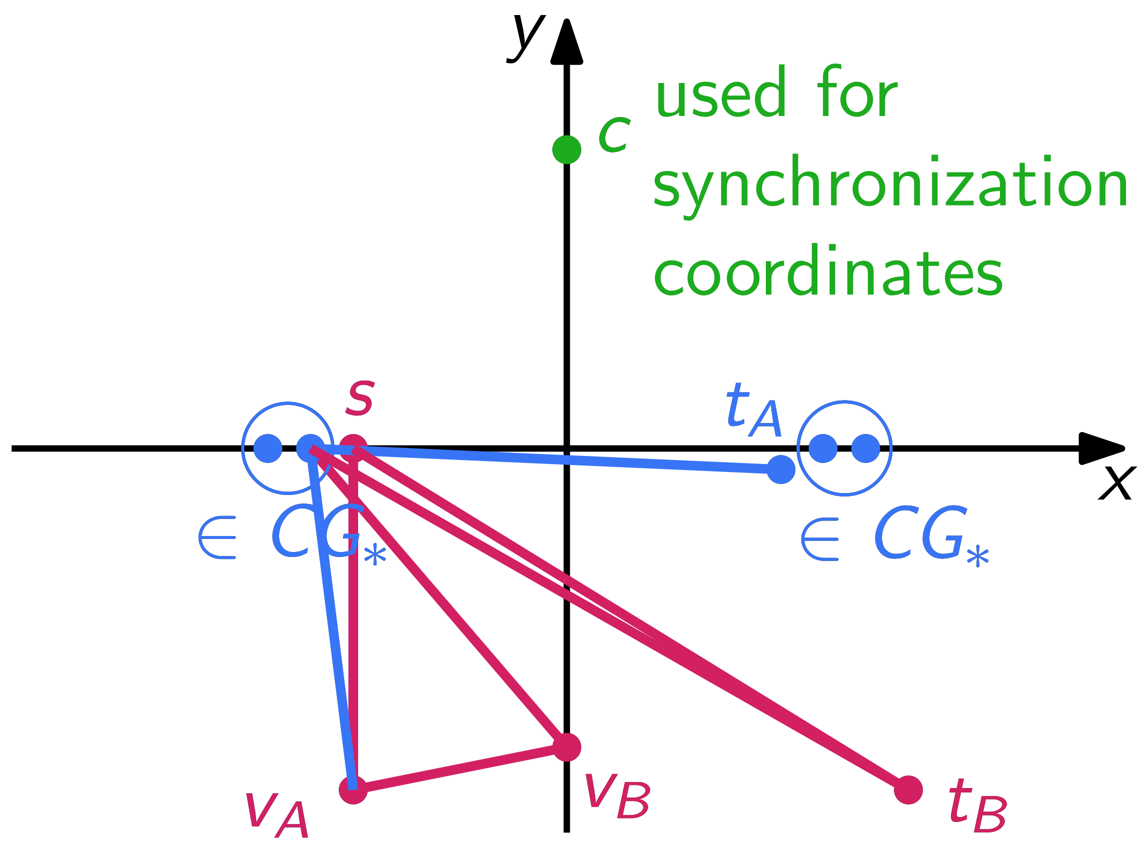
1. Simultaneous walk of curves $A_1, \ldots A_{k-1}$ from s to v_A
2. Curve B walks to v_B over s, v_A
3. Synchronized traversal of all coordinate gadgets CG_*
4. Curves $A_1, \ldots, A_{k-1}$ walk to t_A simultaneously



Reduction Construction

Proof idea

1. Simultaneous walk of curves $A_1, \dots A_{k-1}$ from s to v_A
2. Curve B walks to v_B over s, v_A
3. Synchronized traversal of all coordinate gadgets CG_*
4. Curves $A_1, \dots, A_{k-1}$ walk to t_A simultaneously
5. First B terminates at s, then $A_1, \dots, A_{k-1}$



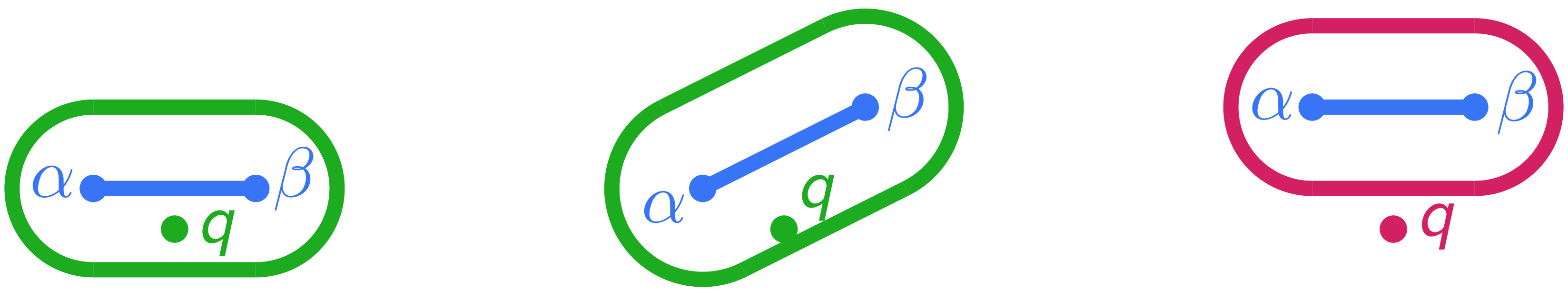
Lower Bound on Curve Simplification

Proof construction

- Reduction from 2 Orthogonal Vectors α, β in d dimensions
- to Curve Simplification in $d + 1$ dimensions

Proof construction

- Reduction from 2 Orthogonal Vectors α, β in d dimensions
- to Curve Simplification in $d+1$ dimensions

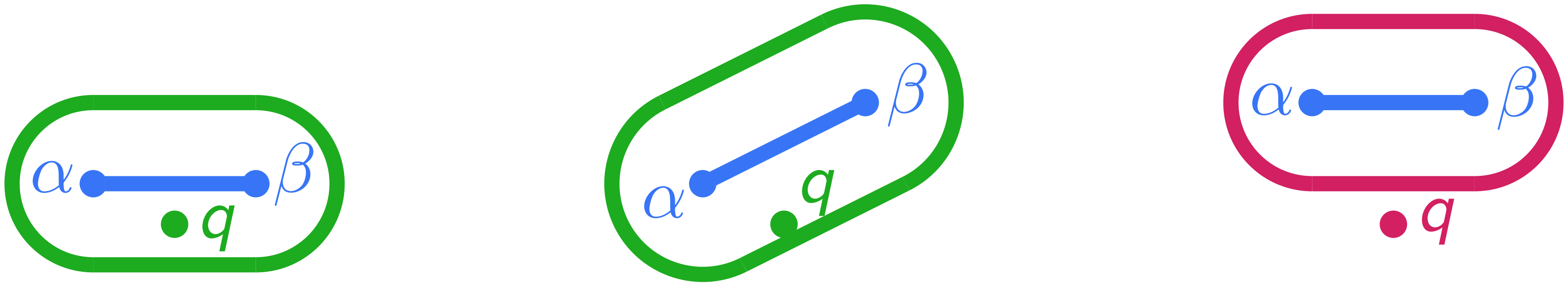


$\alpha = 0, \beta = 0$ $\alpha = 0, \beta = 1$ $\alpha = 1, \beta = 1$

Lower Bound on Curve Simplification

Proof construction

- Reduction from 2 Orthogonal Vectors α, β in d dimensions
- to Curve Simplification in $d+1$ dimensions



$\alpha = 0, \beta = 0$ $\alpha = 0, \beta = 1$ $\alpha = 1, \beta = 1$

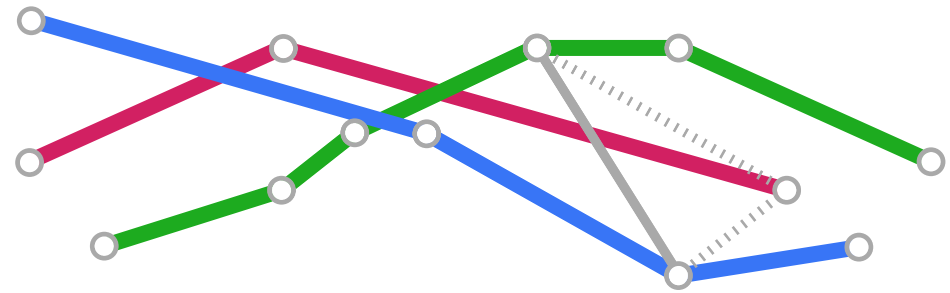
Proof idea

- Checkpoints q are dropped within the simplification iff. α, β are orthogonal
- Simplification consists of 4 vertices iff. α, β are orthogonal
- Simplification consists of at least 5 vertices iff. α, β are non-orthogonal

Conclusion

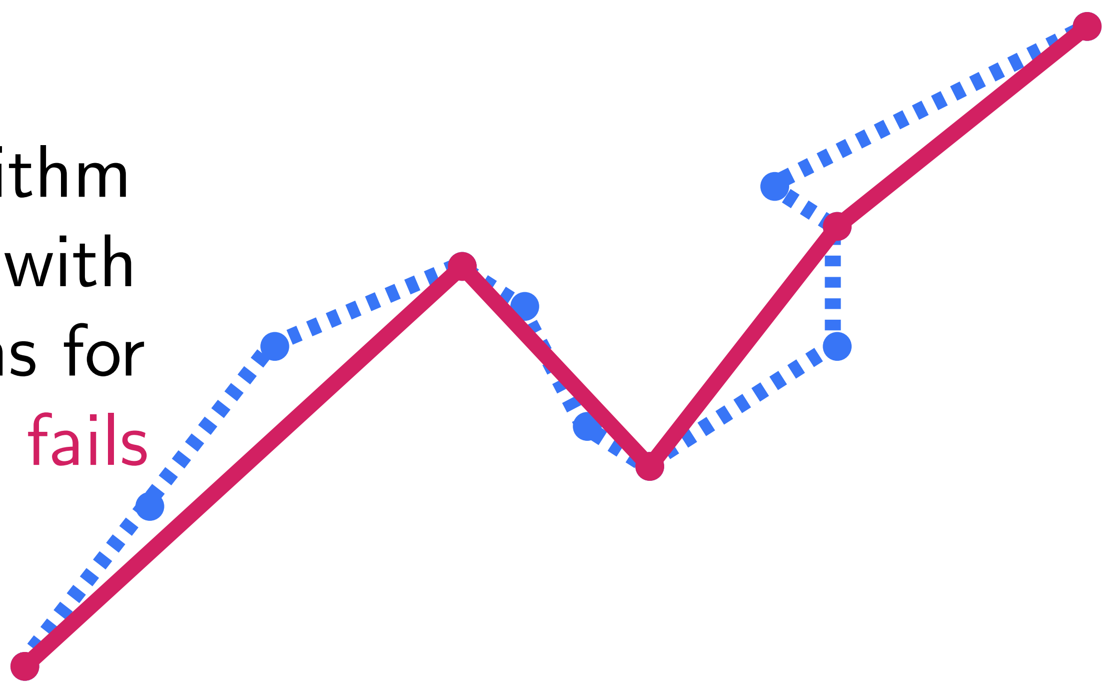
1. Result

- No $O(n^{k-\varepsilon})$ time algorithm for the discrete Fréchet distance on k curves for any $\varepsilon > 0$ unless SETH fails



2. Result

- No $O(n^{2-\varepsilon})$ time algorithm for curve simplification with $d = \Omega(\log n)$ dimensions for any $\varepsilon > 0$ unless SETH fails



Conclusion

Open problems

- Lower bound for simplification in $\mathbb{R}^2$ or $\mathbb{R}^3$
- Upper bound for simplification in $\mathbb{R}^2$ or $\mathbb{R}^3$
- Lower bound for Fréchet distance on k curves with dimension $d = 1$

Thank you for your attention.

Simplification: Orthogonal Case

$L_\infty : \varepsilon = 1, \delta = 2, \delta' = 0.5$

$\alpha = \langle 0, 1, 0 \rangle$

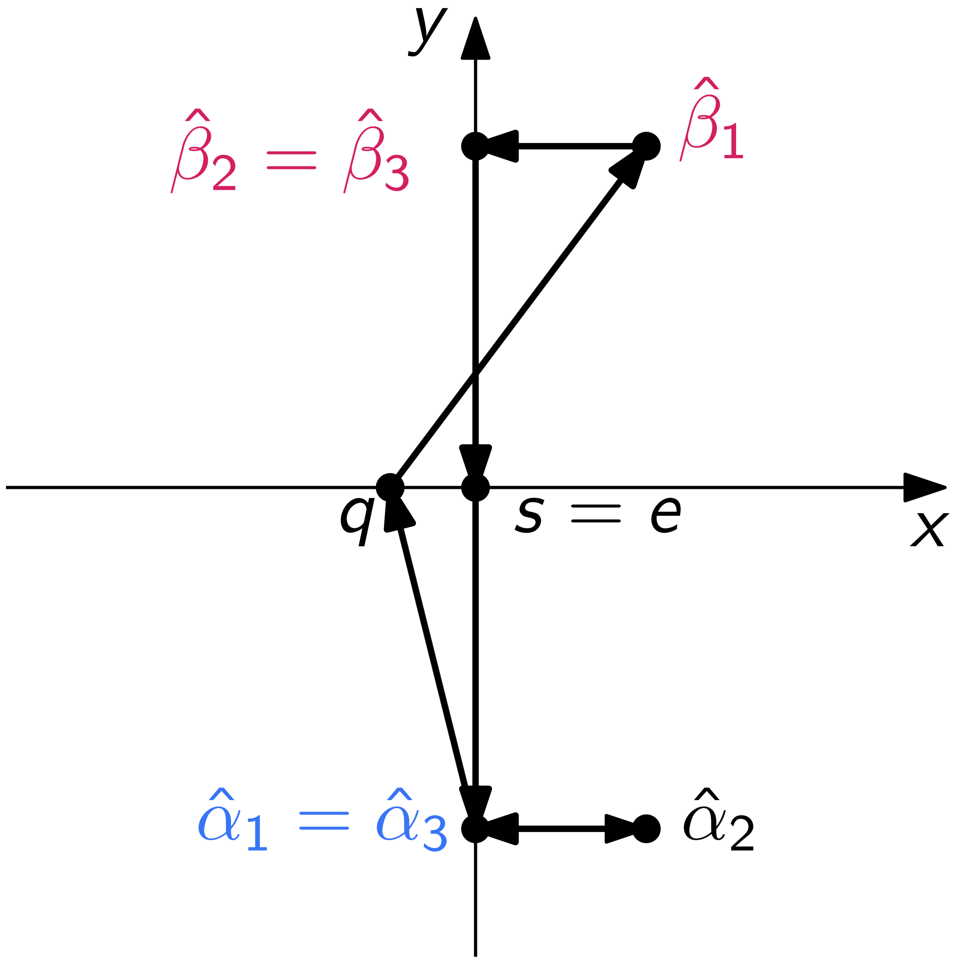
$\beta = \langle 1, 0, 0 \rangle$

$s = e = (0, 0)$

$\hat{\alpha} = \langle (0, -2), (1, -2), (0, -2) \rangle$

$\hat{\beta} = \langle (1, 2), (0, 2), (0, 2) \rangle$

$q = (-0.5, 0)$



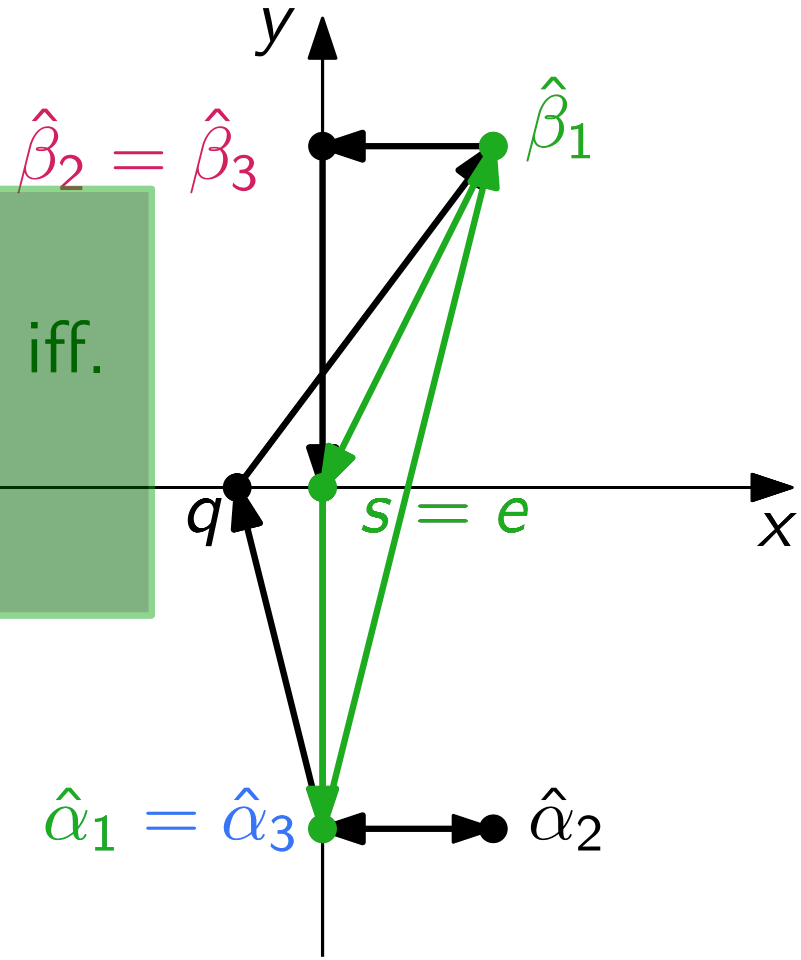
Simplification: Orthogonal Case

$L_\infty : \varepsilon = 1, \delta = 2, \delta' = 0.5$

$\alpha = \langle 0, 1, 0 \rangle$ $\hat{\alpha} = \langle (0, -2), (1, -2), (0, -2) \rangle$

$\beta = \langle 1, 0, 0 \rangle$ $\hat{\beta} = \langle (1, 2), (0, 2), (0, 2) \rangle$

$s = e = (0, 0)$ $q = (-0.5, 0)$



Simplification contains 4 vertices iff. α and β are orthogonal

Simplification: Non-orthogonal Case

$L_\infty : \varepsilon = 1, \delta = 2, \delta' = 0.5$

$\alpha = \langle 1, 0, 1 \rangle$

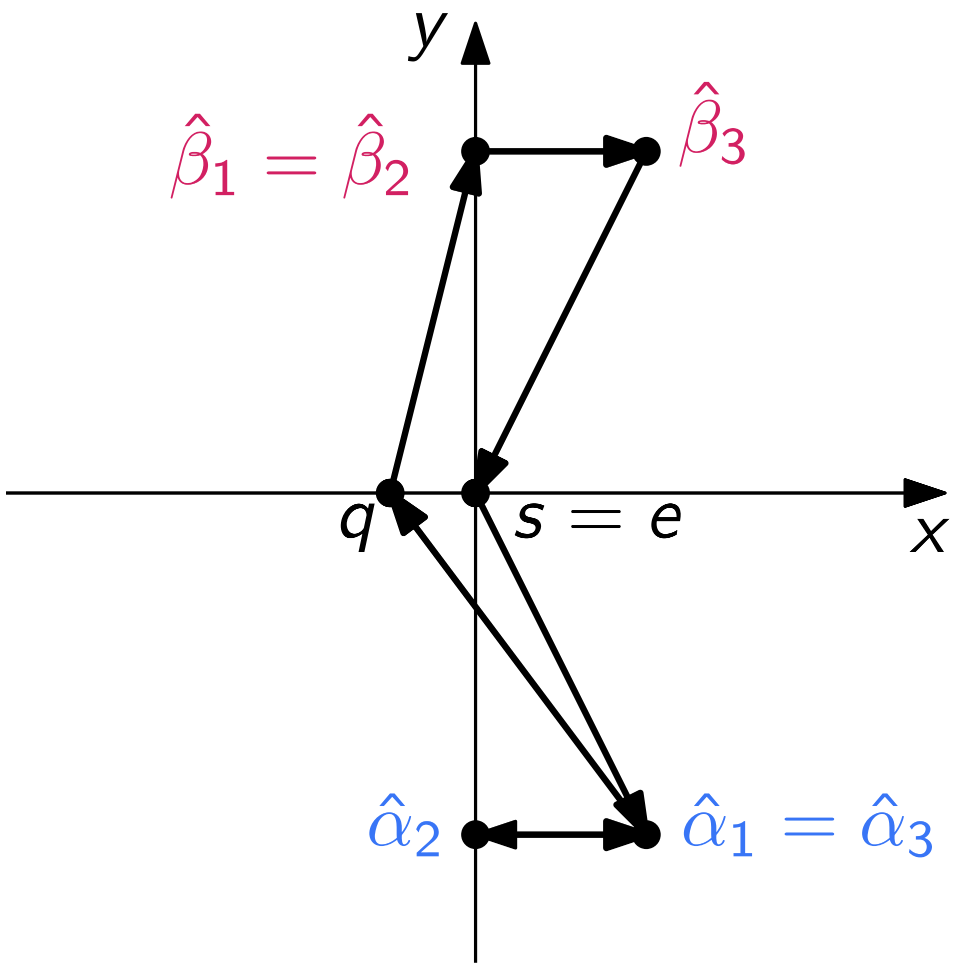
$\beta = \langle 0, 0, 1 \rangle$

$s = e = (0, 0)$

$\hat{\alpha} = \langle (1, -2), (0, -2), (1, -2) \rangle$

$\hat{\beta} = \langle (0, 2), (0, 2), (1, 2) \rangle$

$q = (-0.5, 0)$



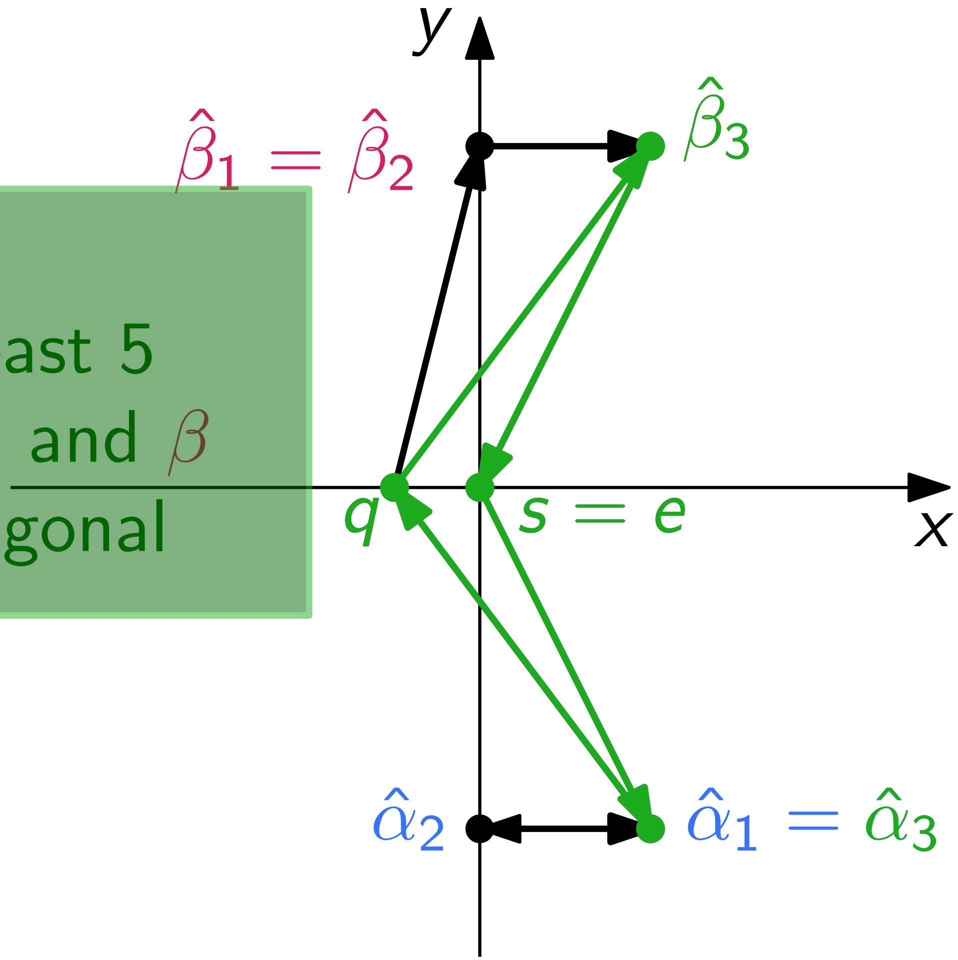
Simplification: Non-orthogonal Case

$L_\infty : \varepsilon = 1, \delta = 2, \delta' = 0.5$

$\alpha = \langle 1, 0, 1 \rangle$ $\hat{\alpha} = \langle (1, -2), (0, -2), (1, -2) \rangle$

$\beta = \langle 0, 0, 1 \rangle$ $\hat{\beta} = \langle (0, 2), (0, 2), (1, 2) \rangle$

$s = e = (0, 0)$ $q = (-0.5, 0)$



Simplification contains at least 5 vertices iff. α and β are non-orthogonal