

Walking Your Dog in the Woods in Polynomial Time



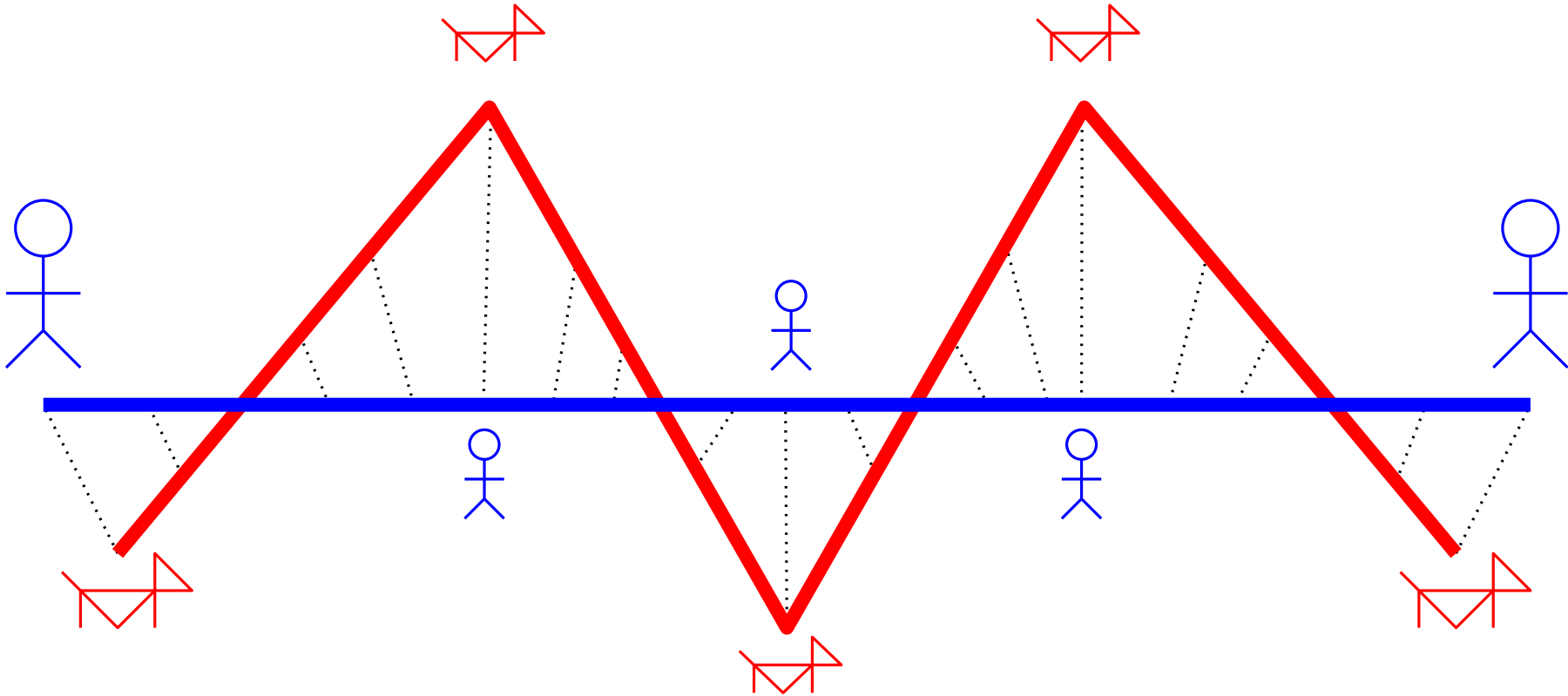
Shripad Thite

California Institute of Technology

Joint work with

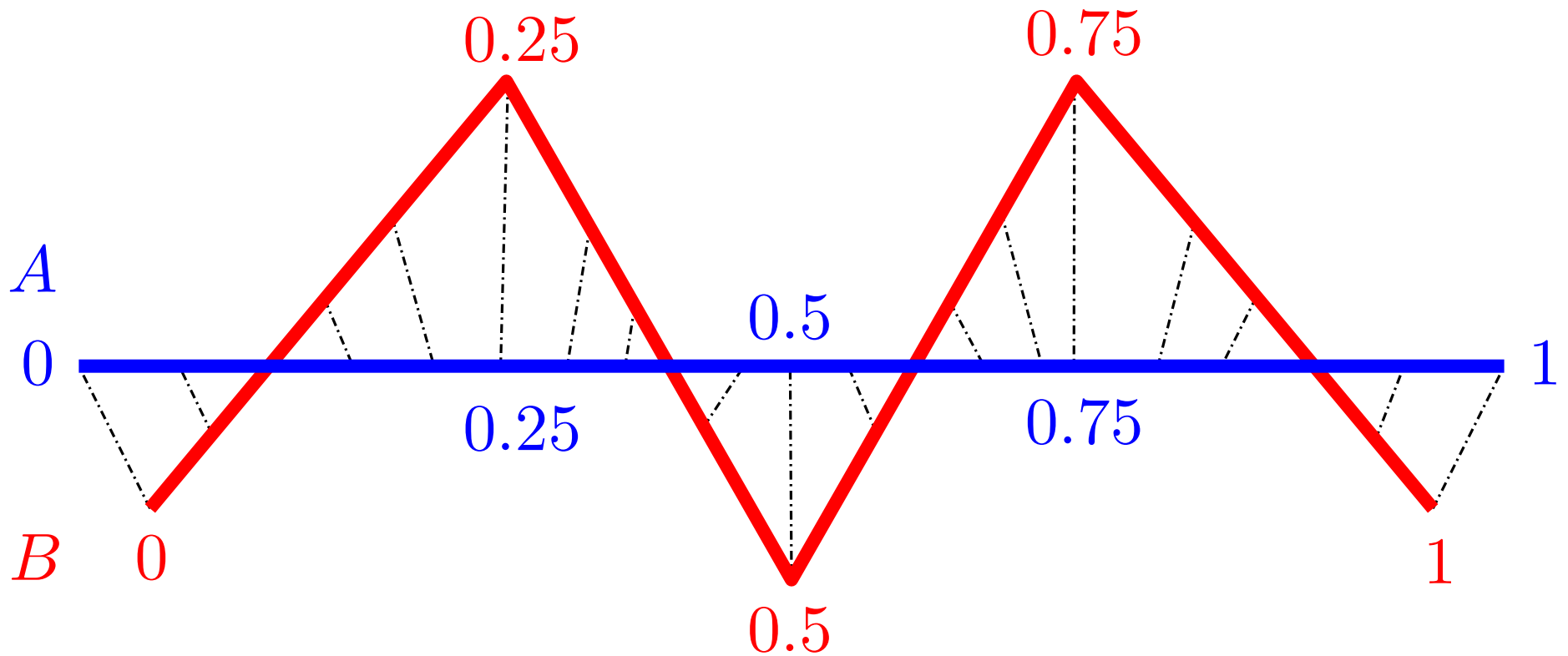
Erin Wolf Chambers, Éric Colin de Verdière, Jeff Erickson,
Sylvain Lazard, Francis Lazarus

Fréchet distance between curves



Dog-leash distance: Minimum length of a straight leash joining a **dog** and its **owner** as they walk along their respective curves, from one endpoint to the other, continuously without backtracking

Fréchet distance between curves



Better measure of similarity than Hausdorff distance since it accounts for the flow of the curves (handwriting recognition)

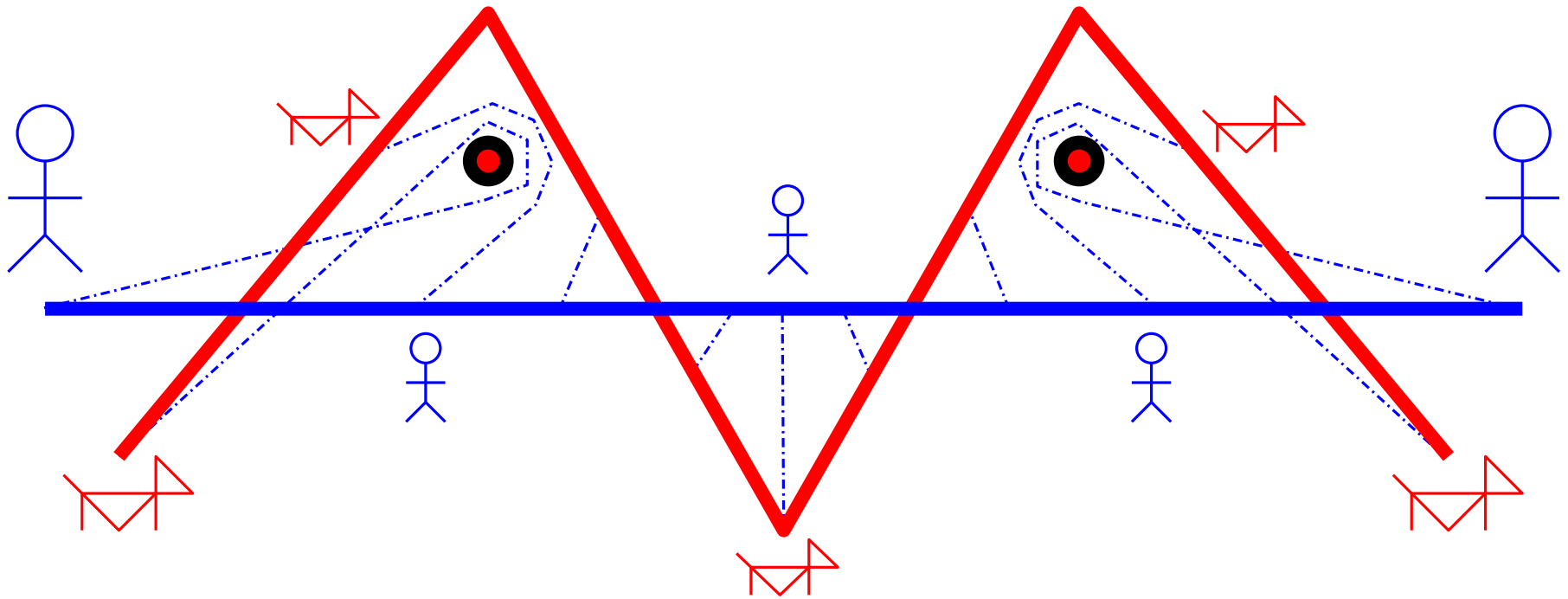
Defined by *Maurice Fréchet (1878-1973)*

$O(N^3 \log N)$ algorithm by *Alt and Godau, 1995*

Parks have trees

...and other obstacles

New condition: Leash cannot jump over obstacles

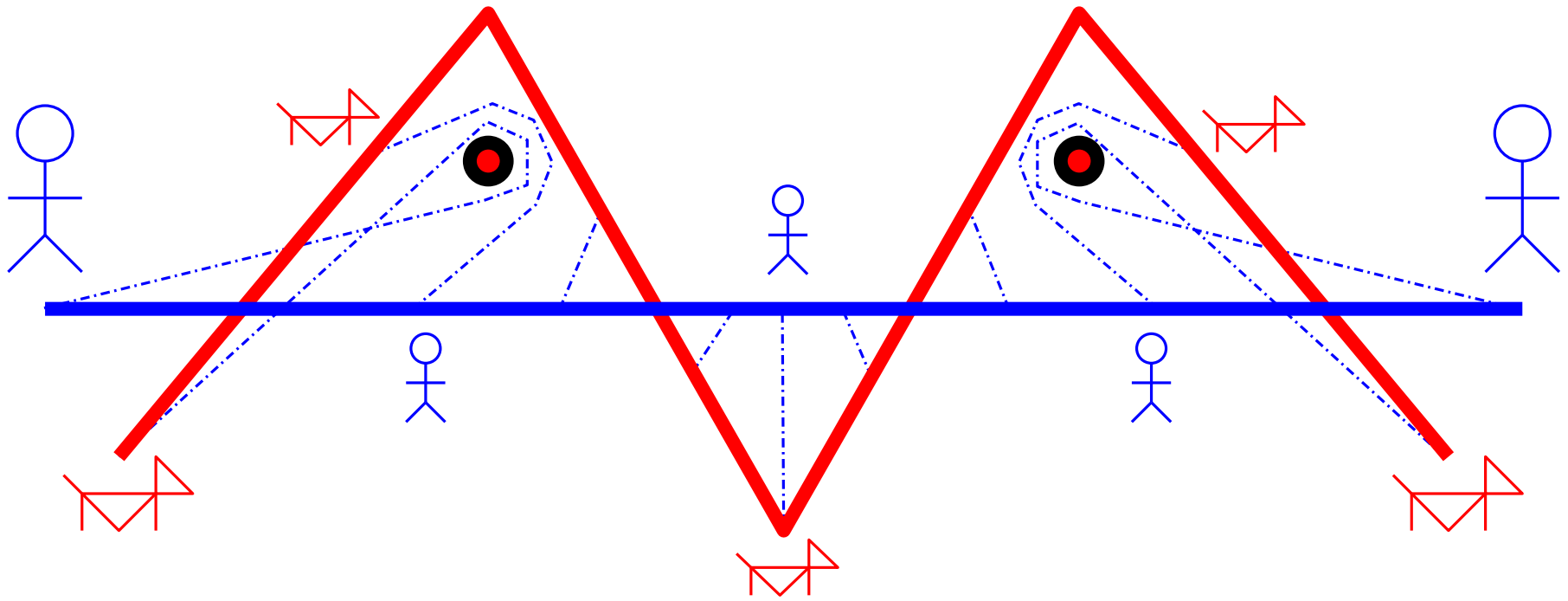


Goal: Walk the dog with the shortest leash possible

A longer leash may be required than without obstacles

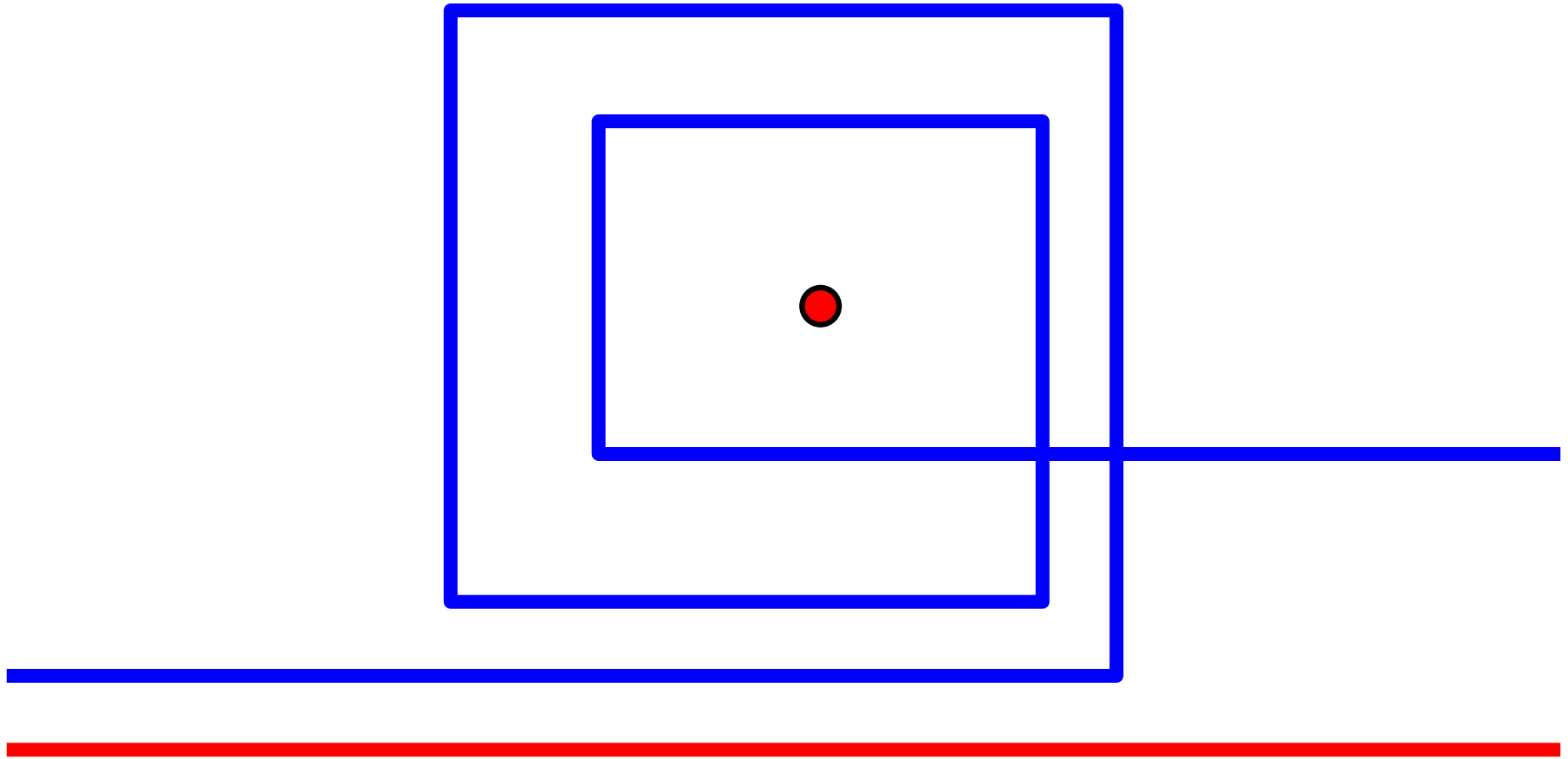
Homotopic Fréchet distance

Dog-leash distance where the leash must move continuously



We give polynomial-time algorithms to compute the homotopic Fréchet distance between two polygonal curves in the plane with **point obstacles**

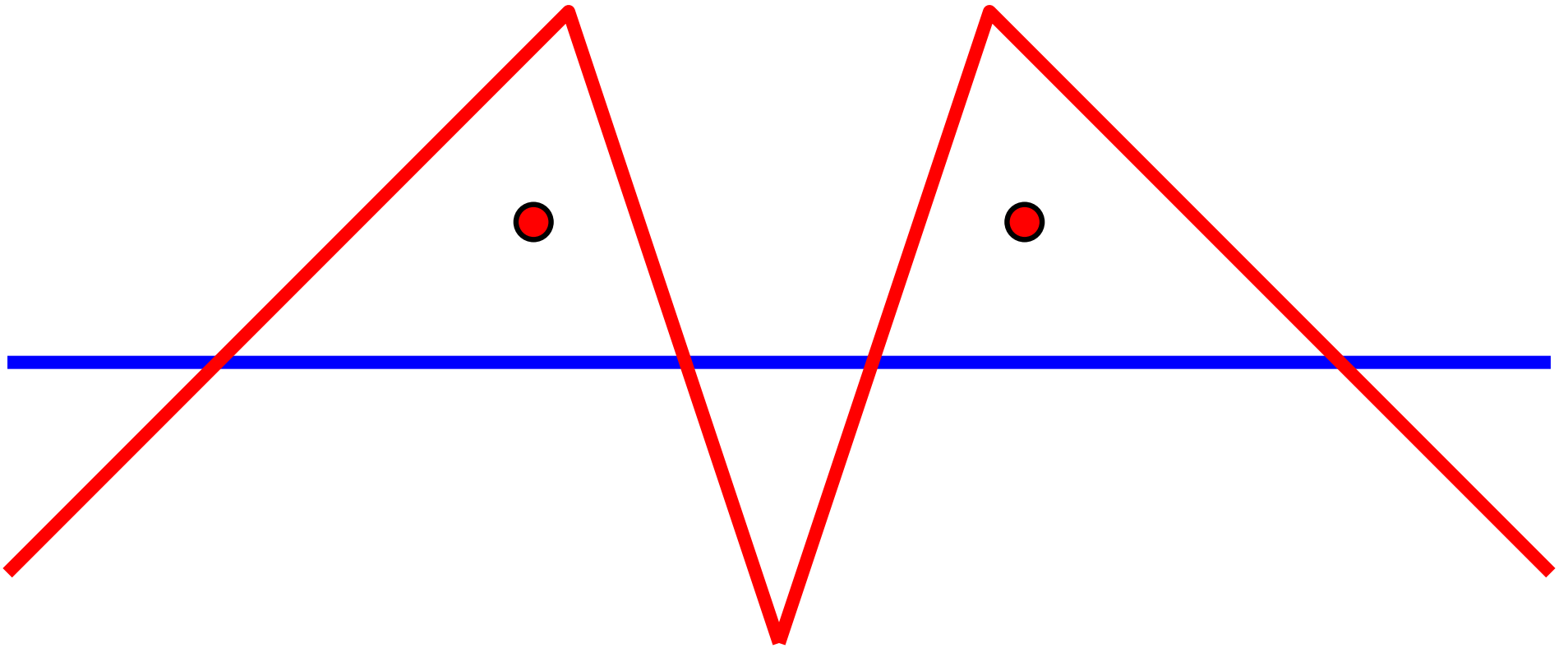
Application 1: Measuring similarity



Robotics: Given two paths in configuration space, how 'similar' are they?

C-space contains obstacles, i.e., disallowed configurations

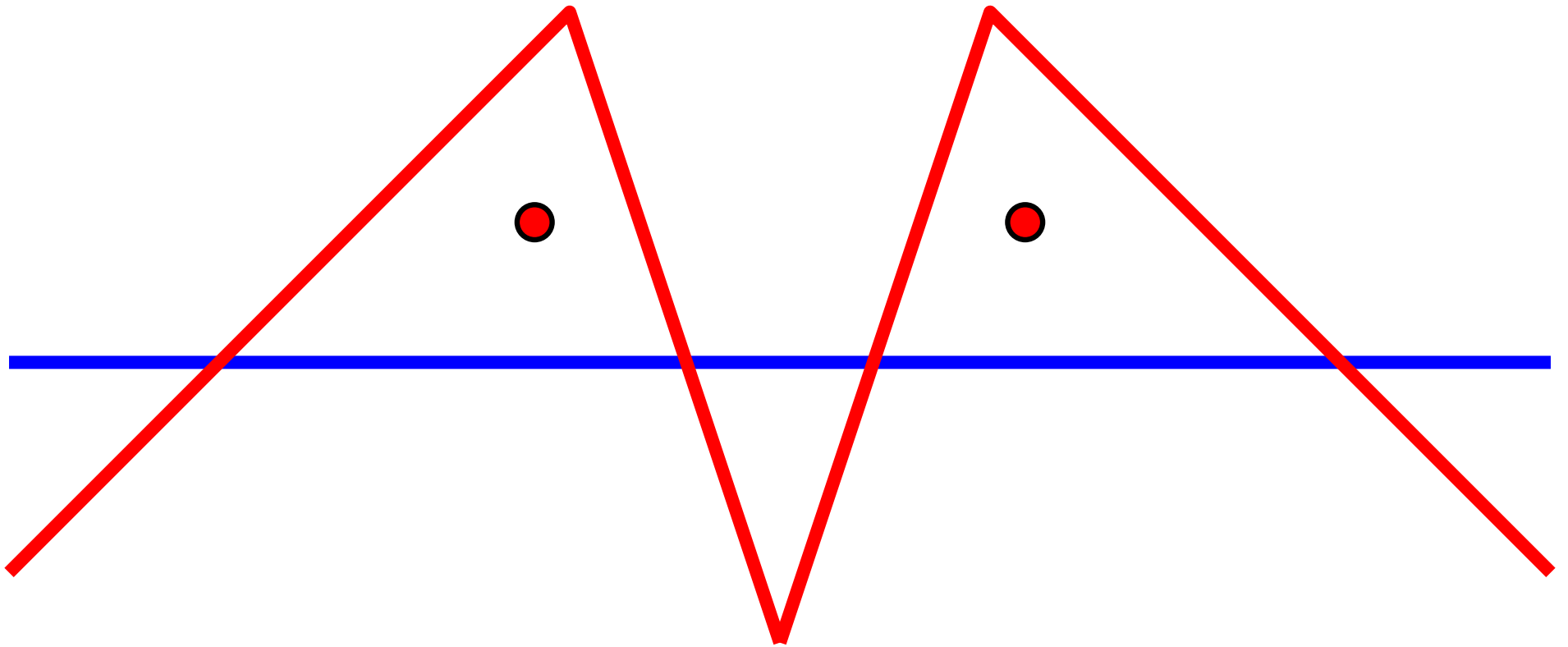
Application 2: Morphing



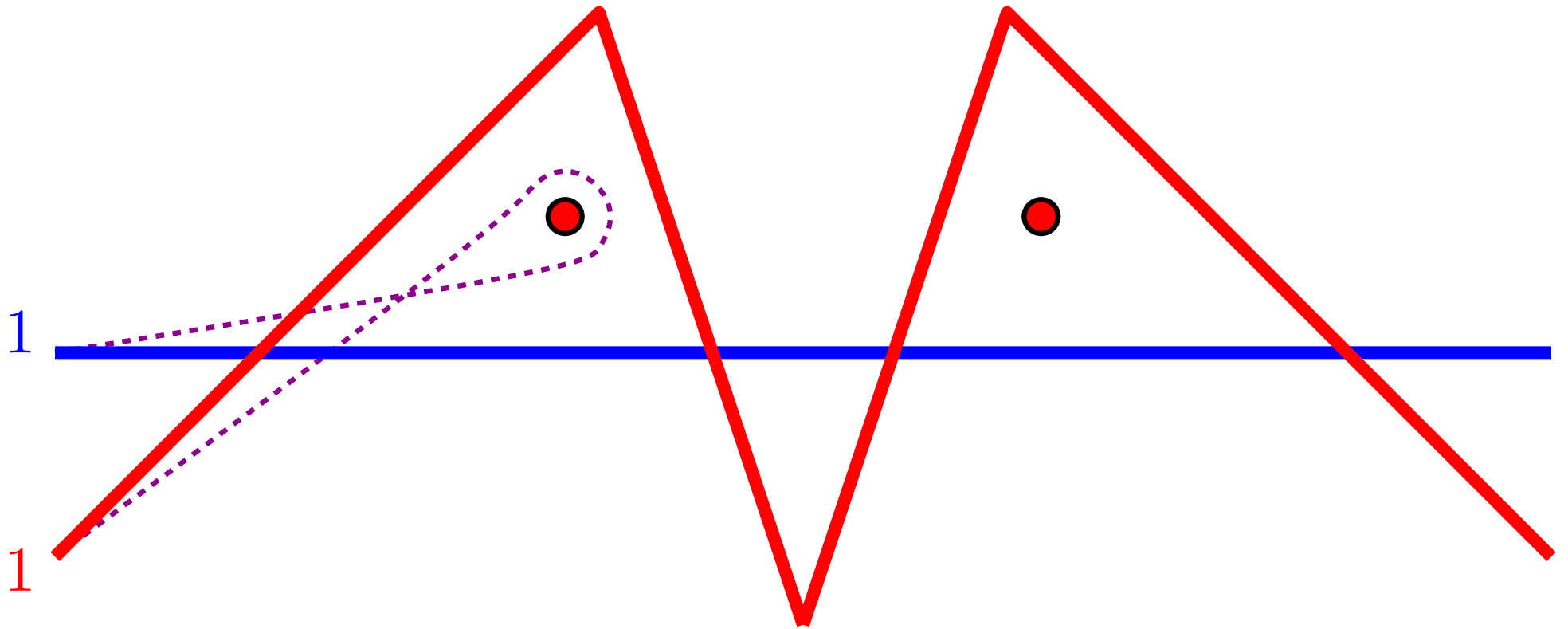
Leash motion encodes a continuous deformation between A and B

The “cost” of the deformation is the maximum distance any point has to travel, i.e., $\max_{t \in [0,1]} \text{dist}(A(t), B(t))$

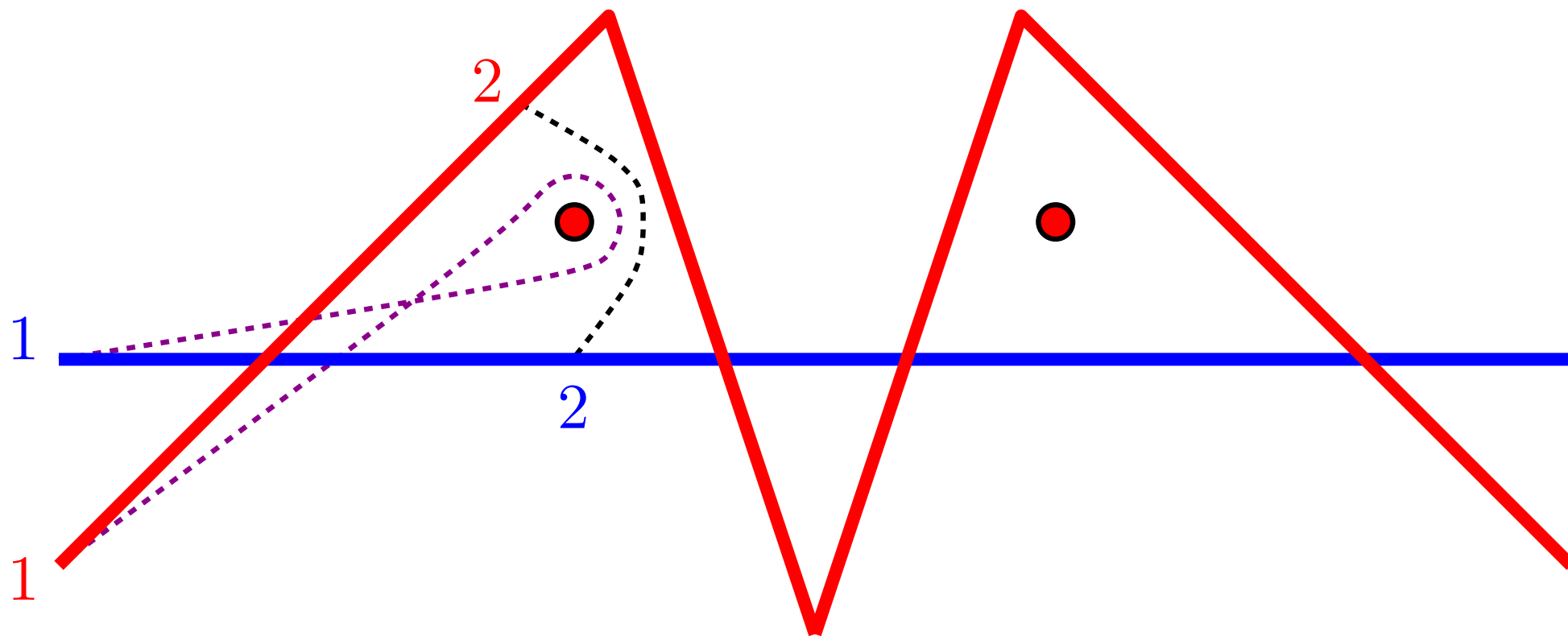
Example 1



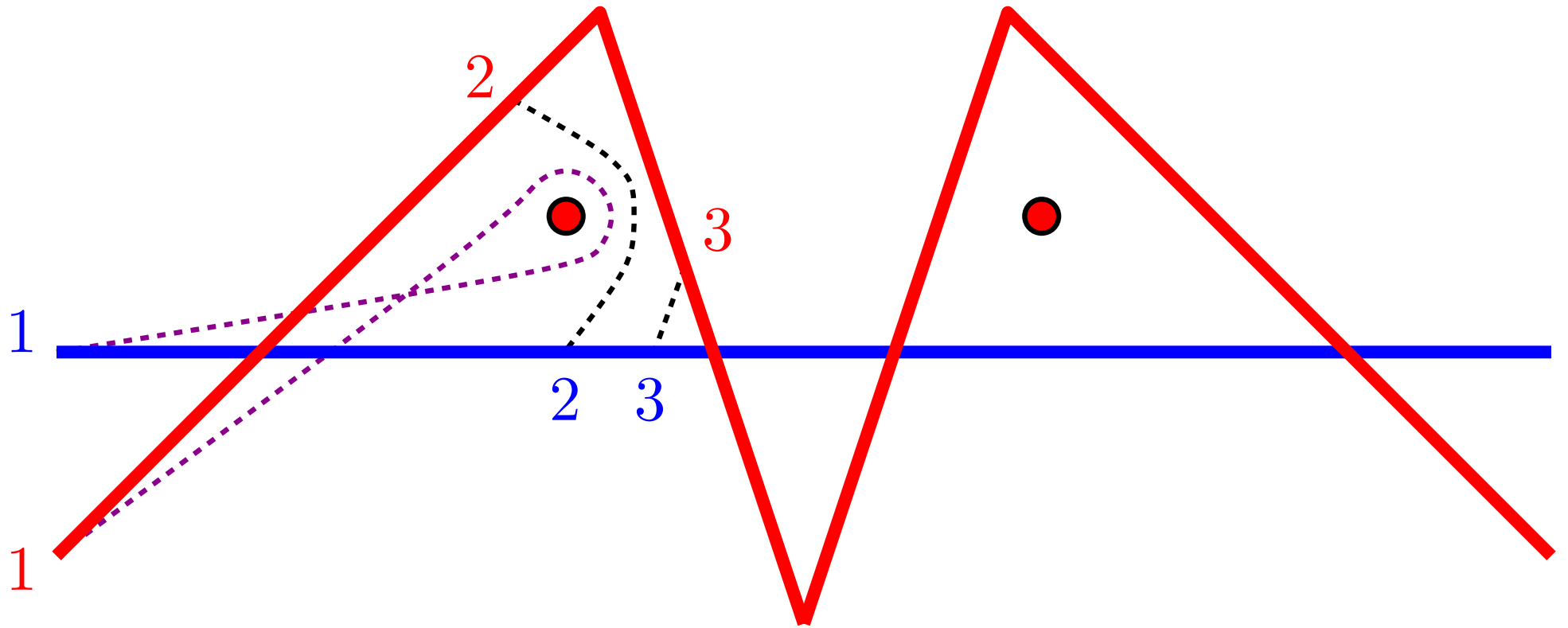
Example 1



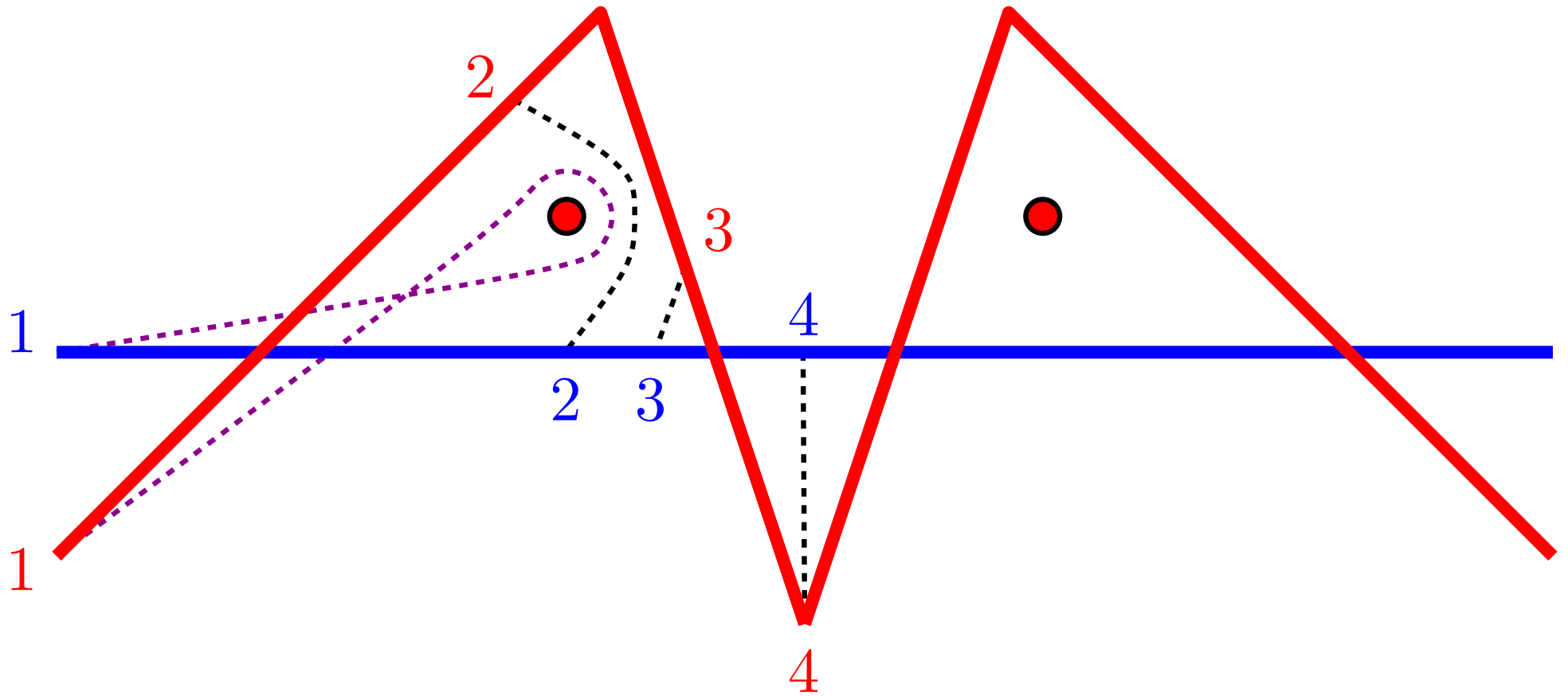
Example 1



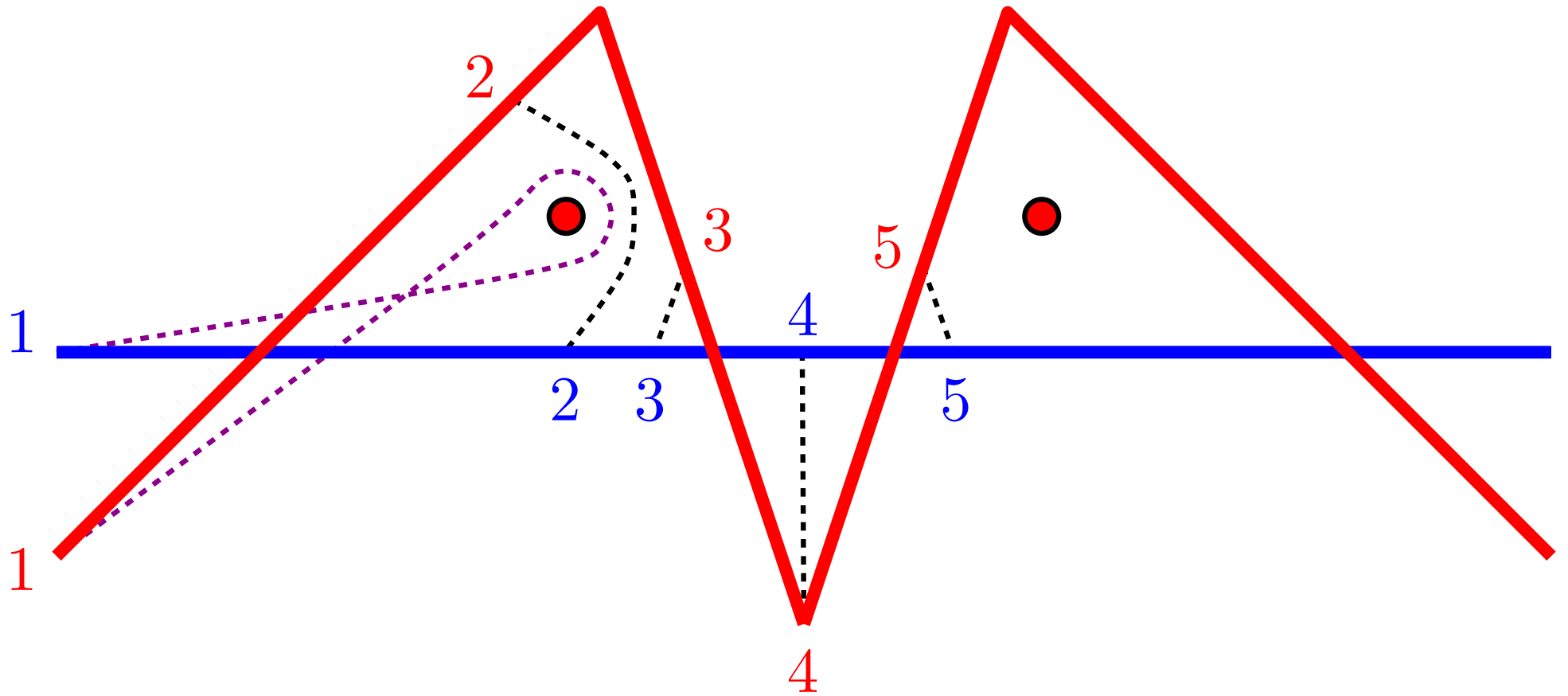
Example 1



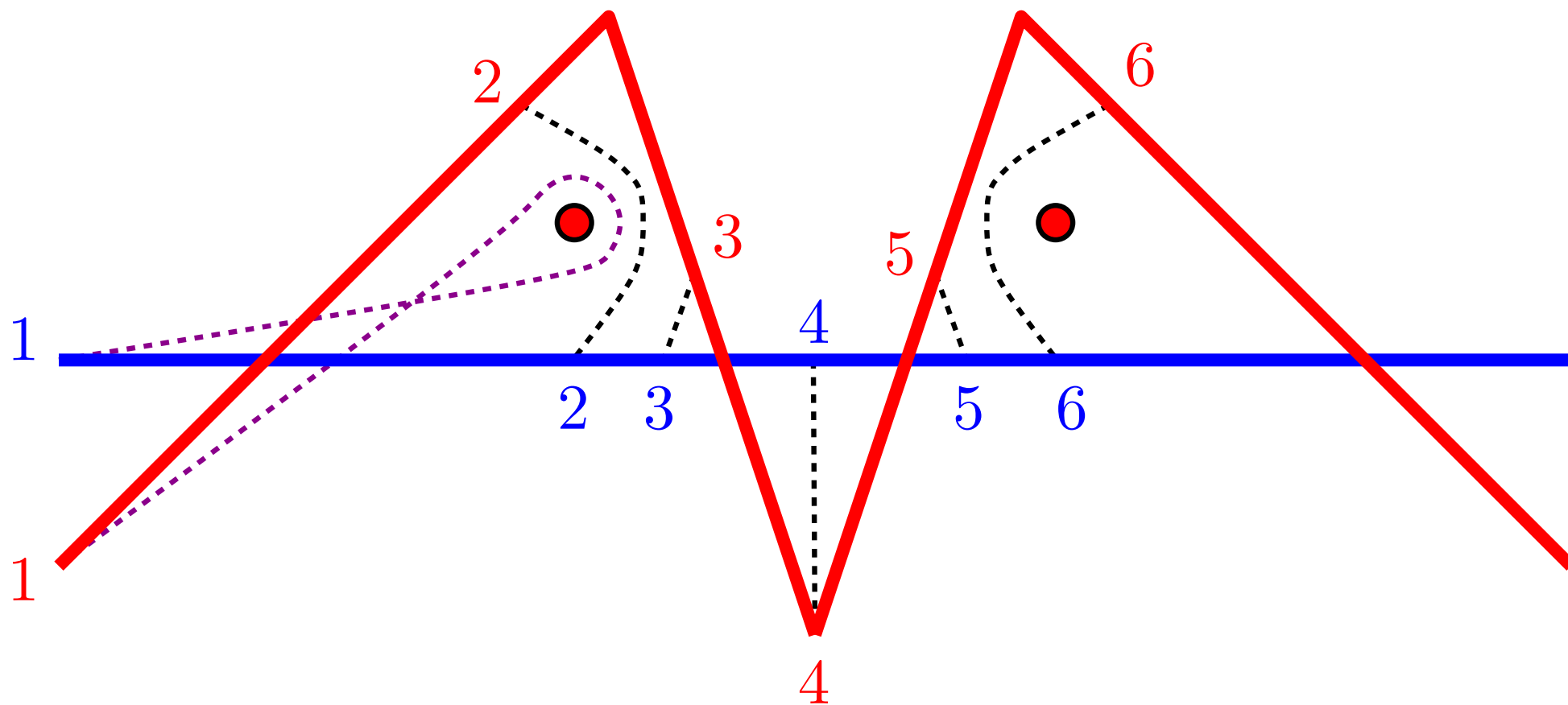
Example 1



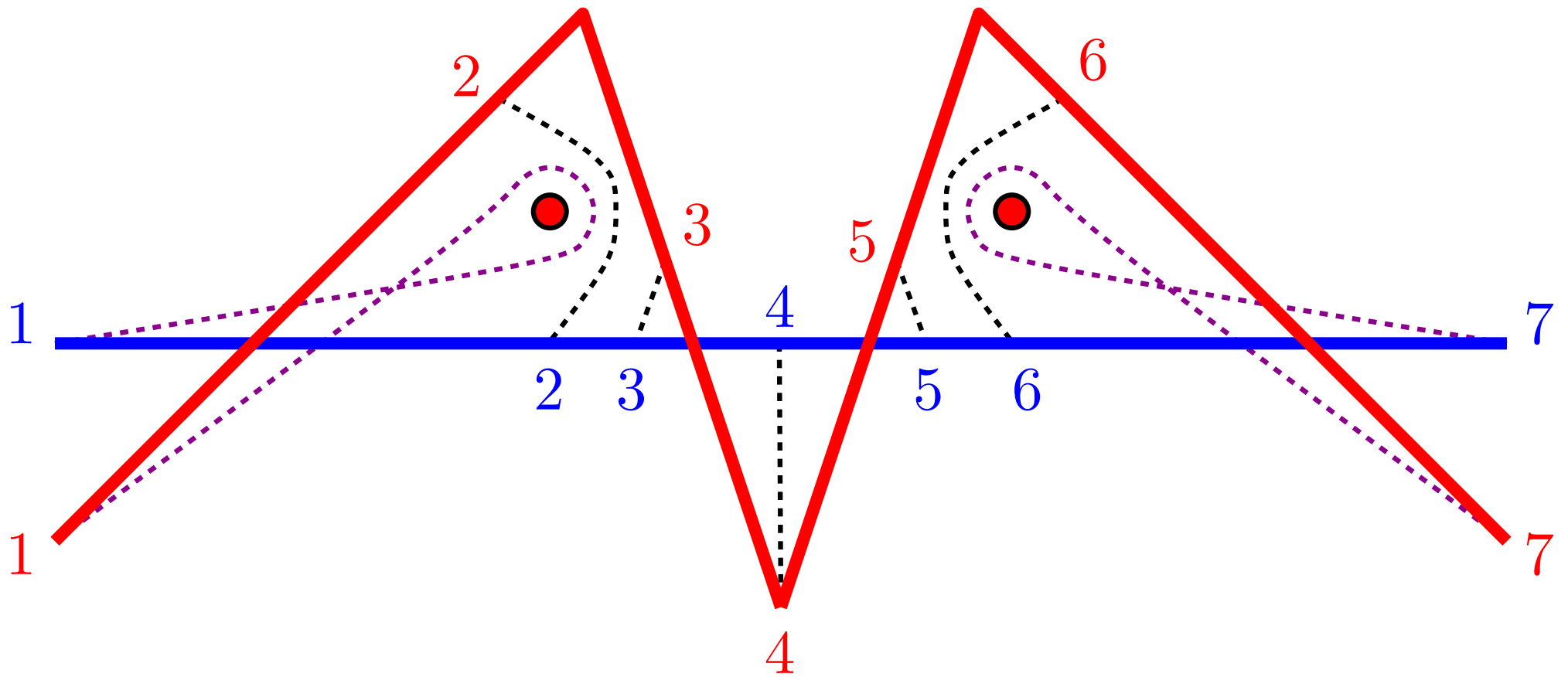
Example 1



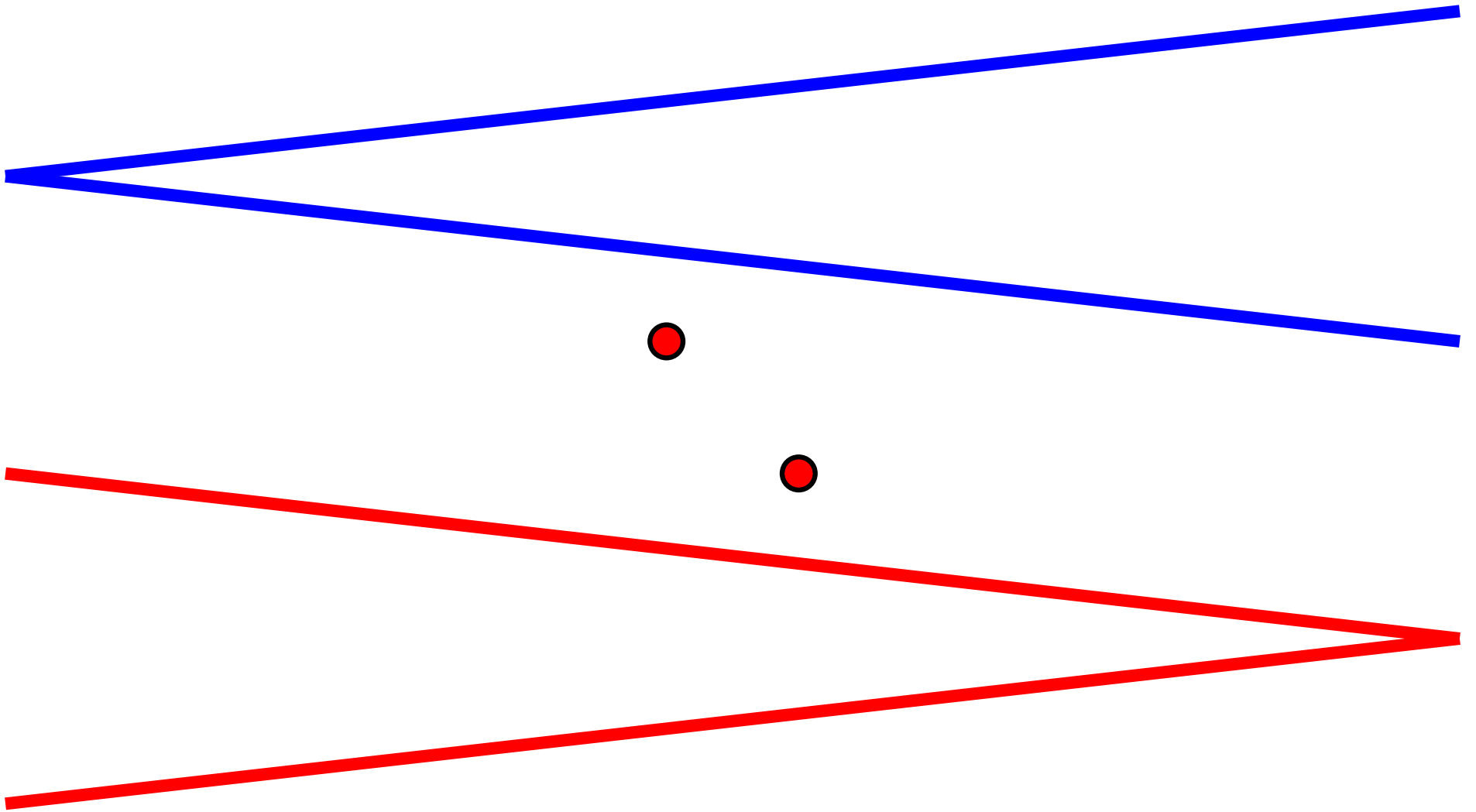
Example 1



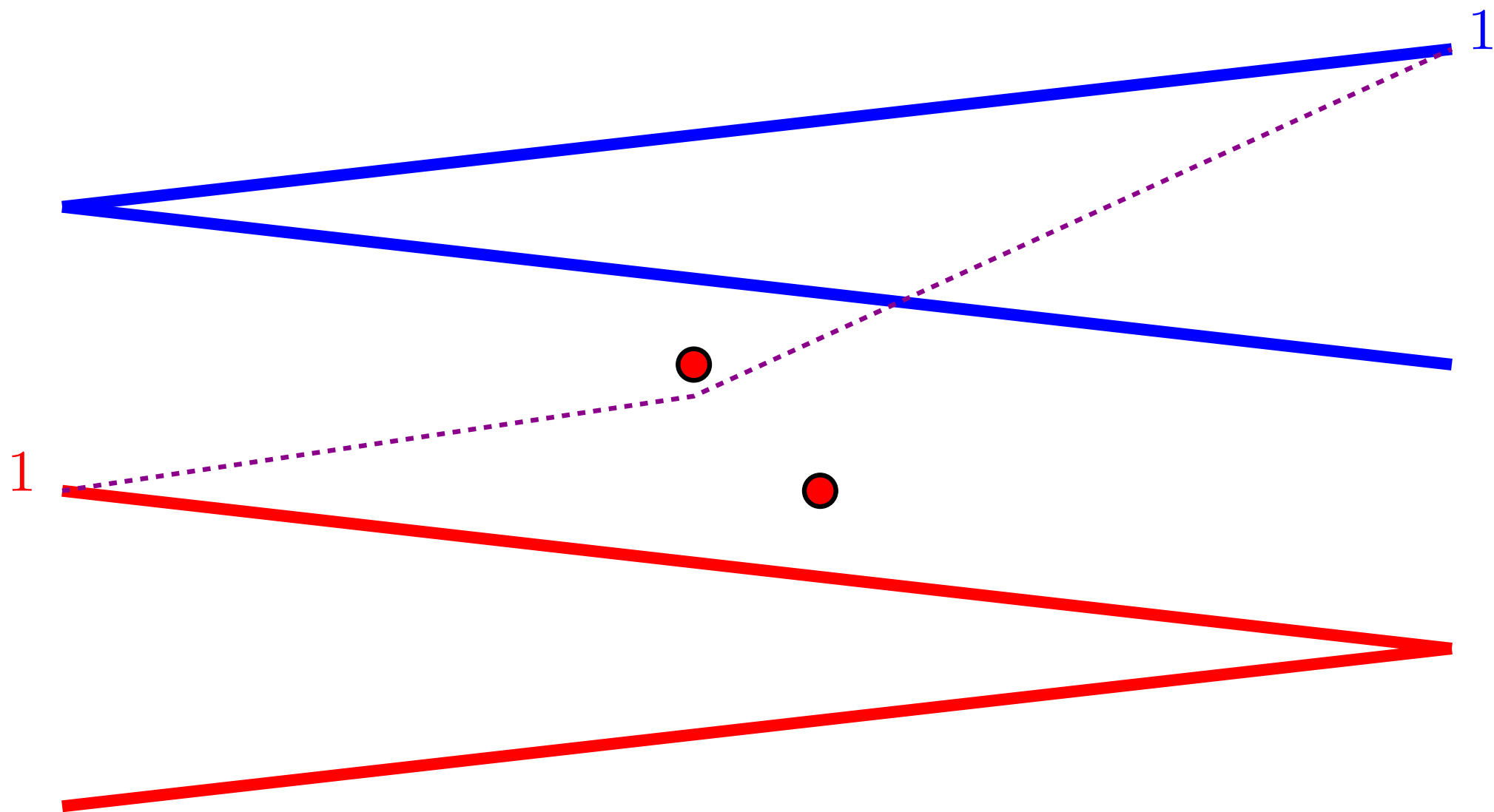
Example 1



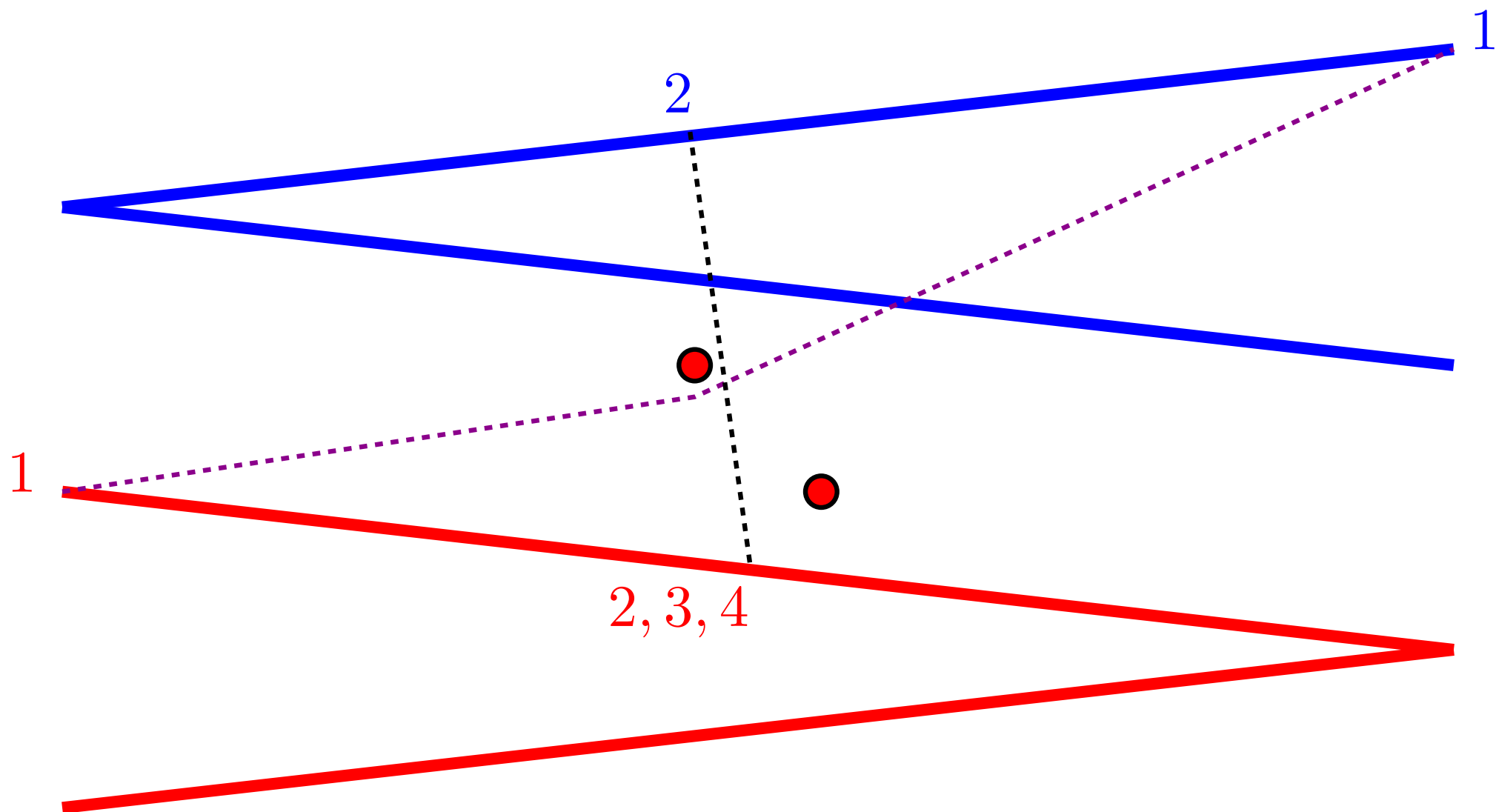
Example 2



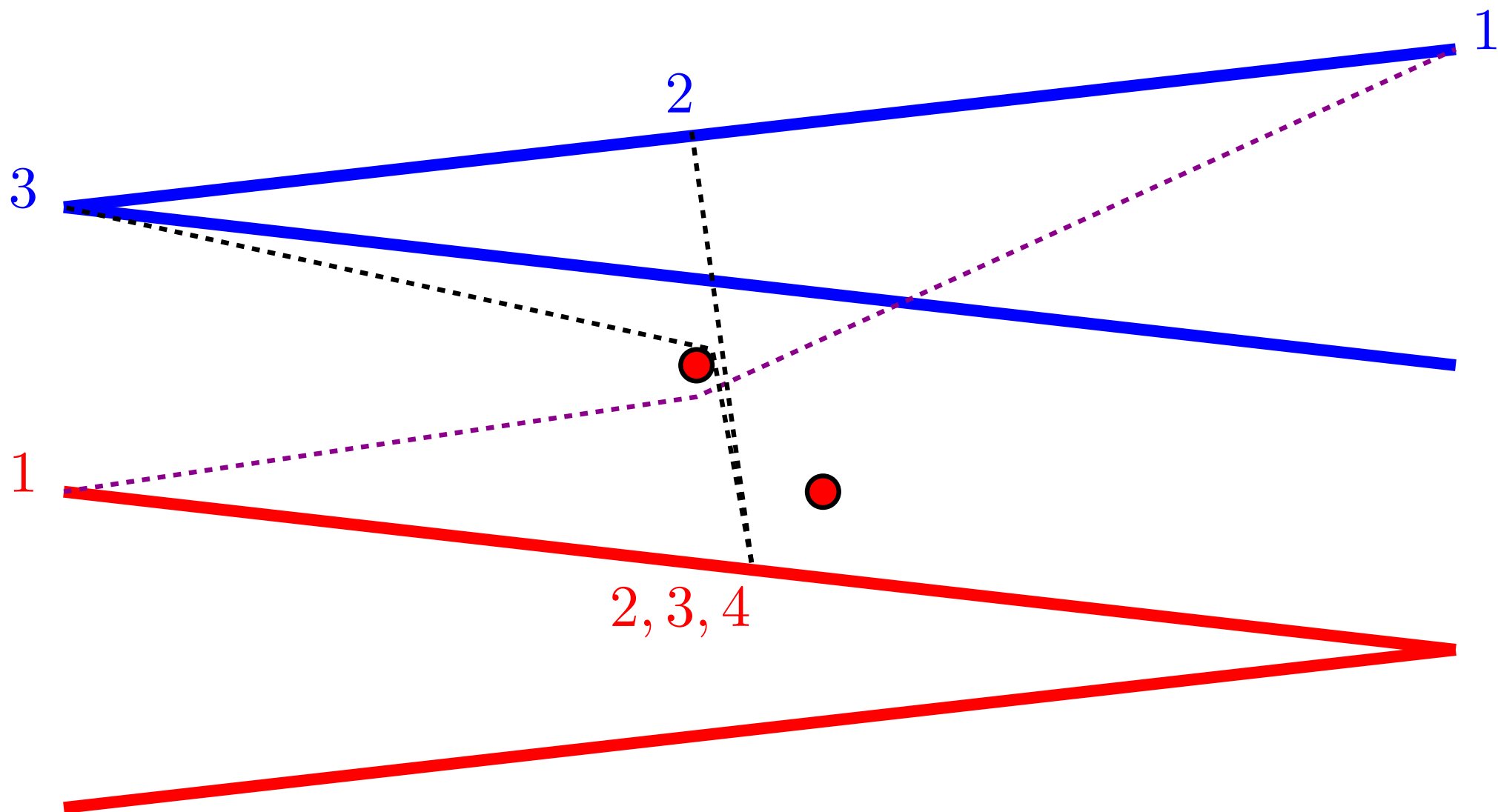
Example 2



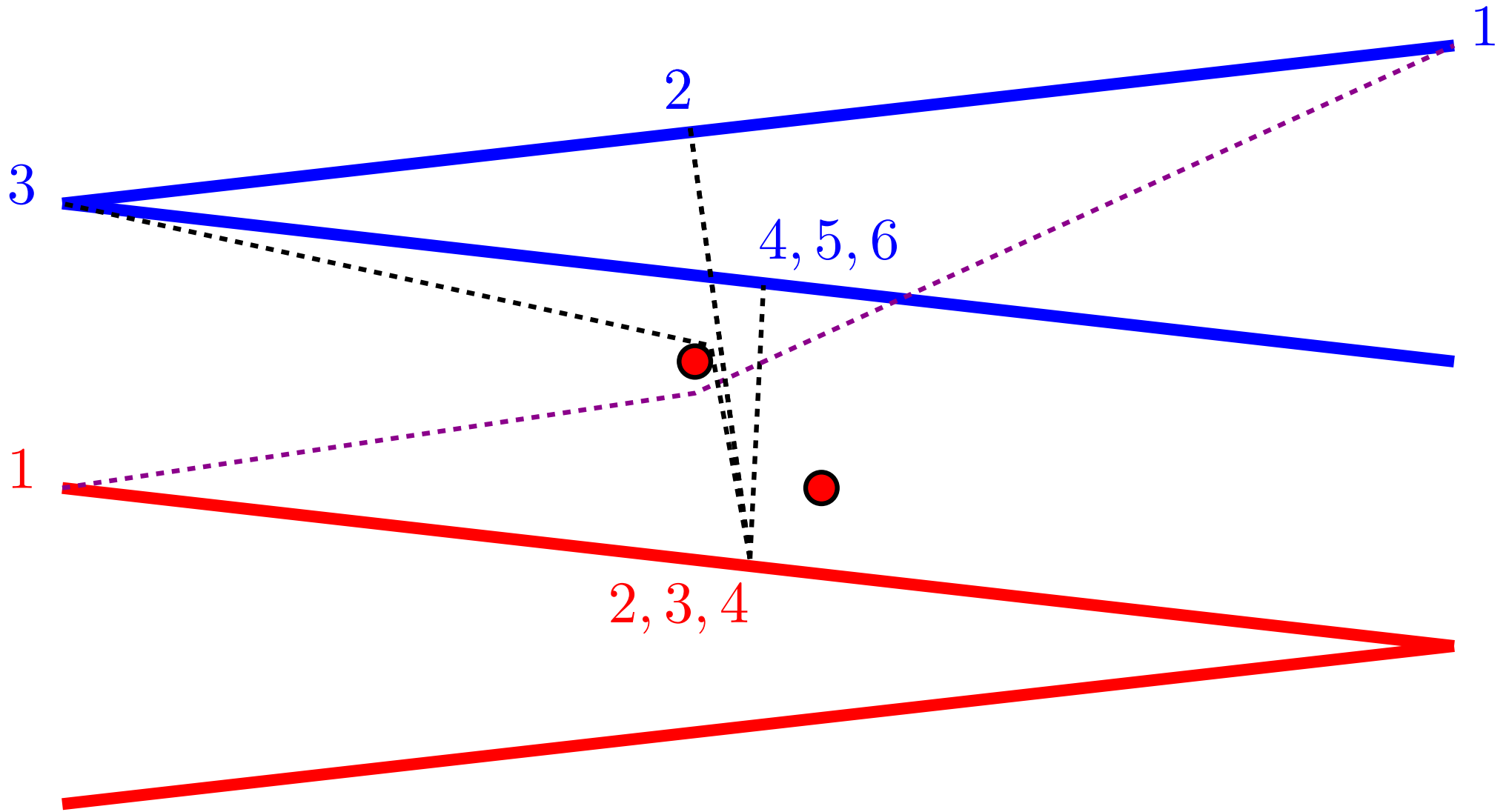
Example 2



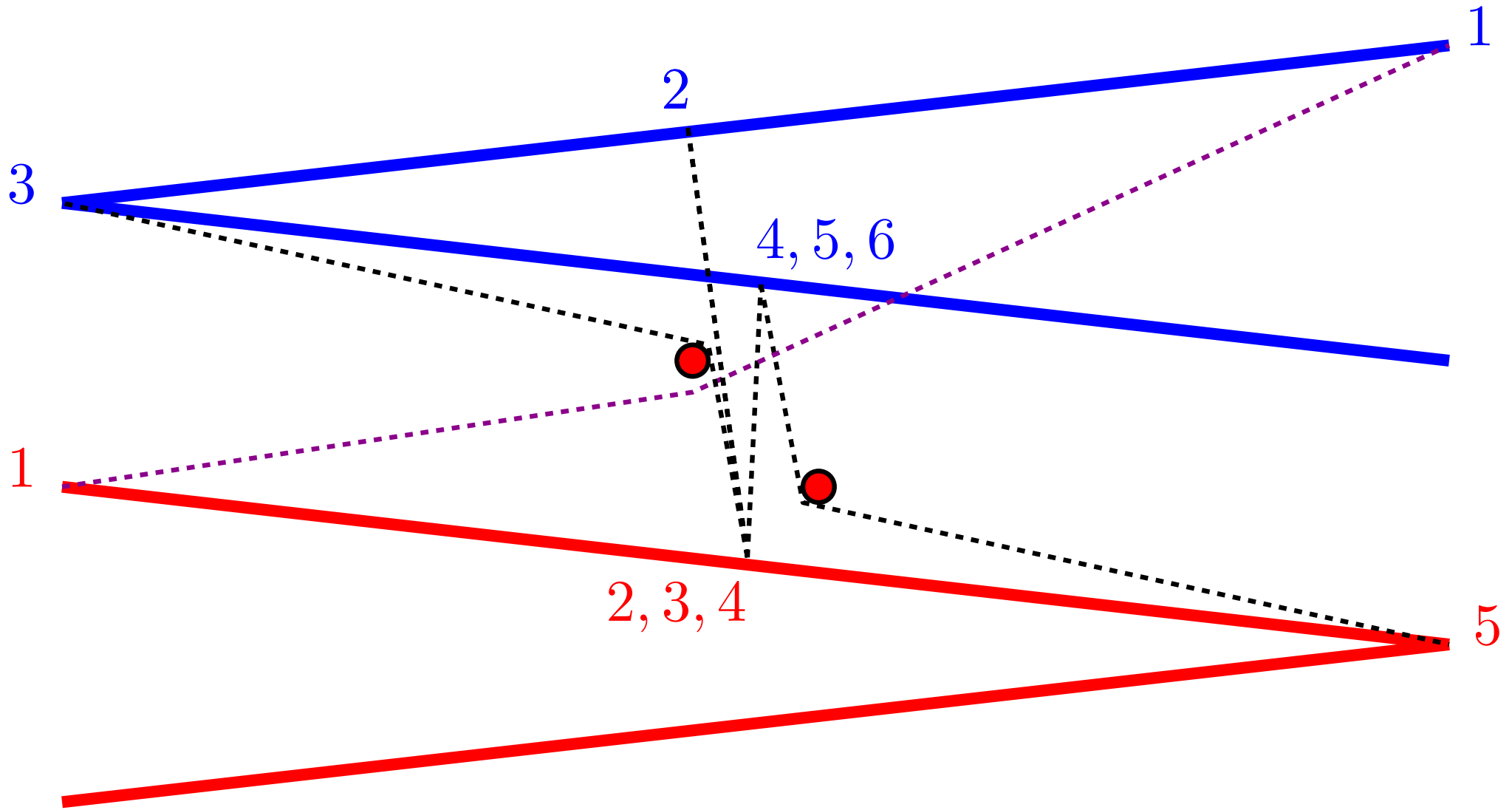
Example 2



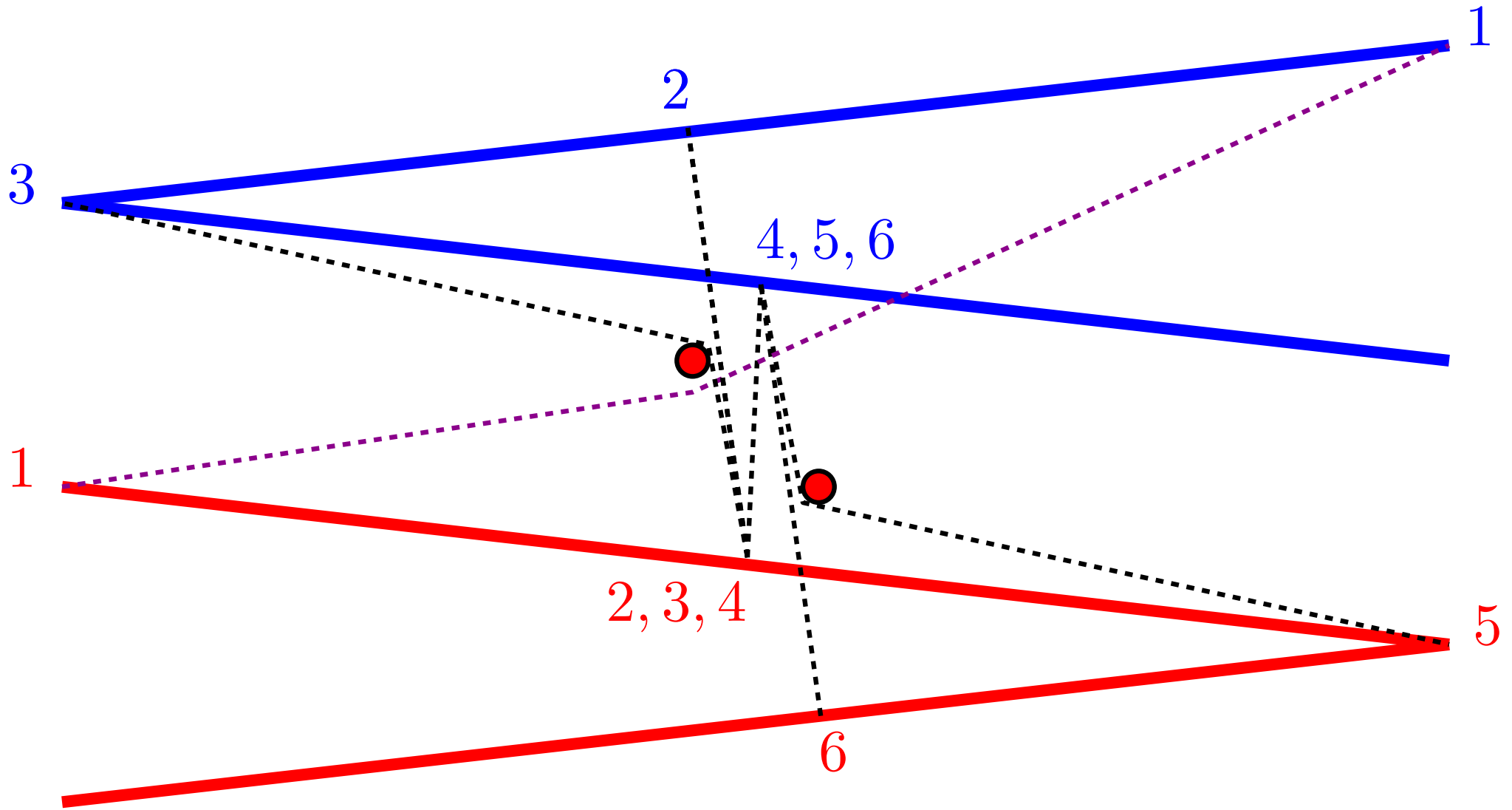
Example 2



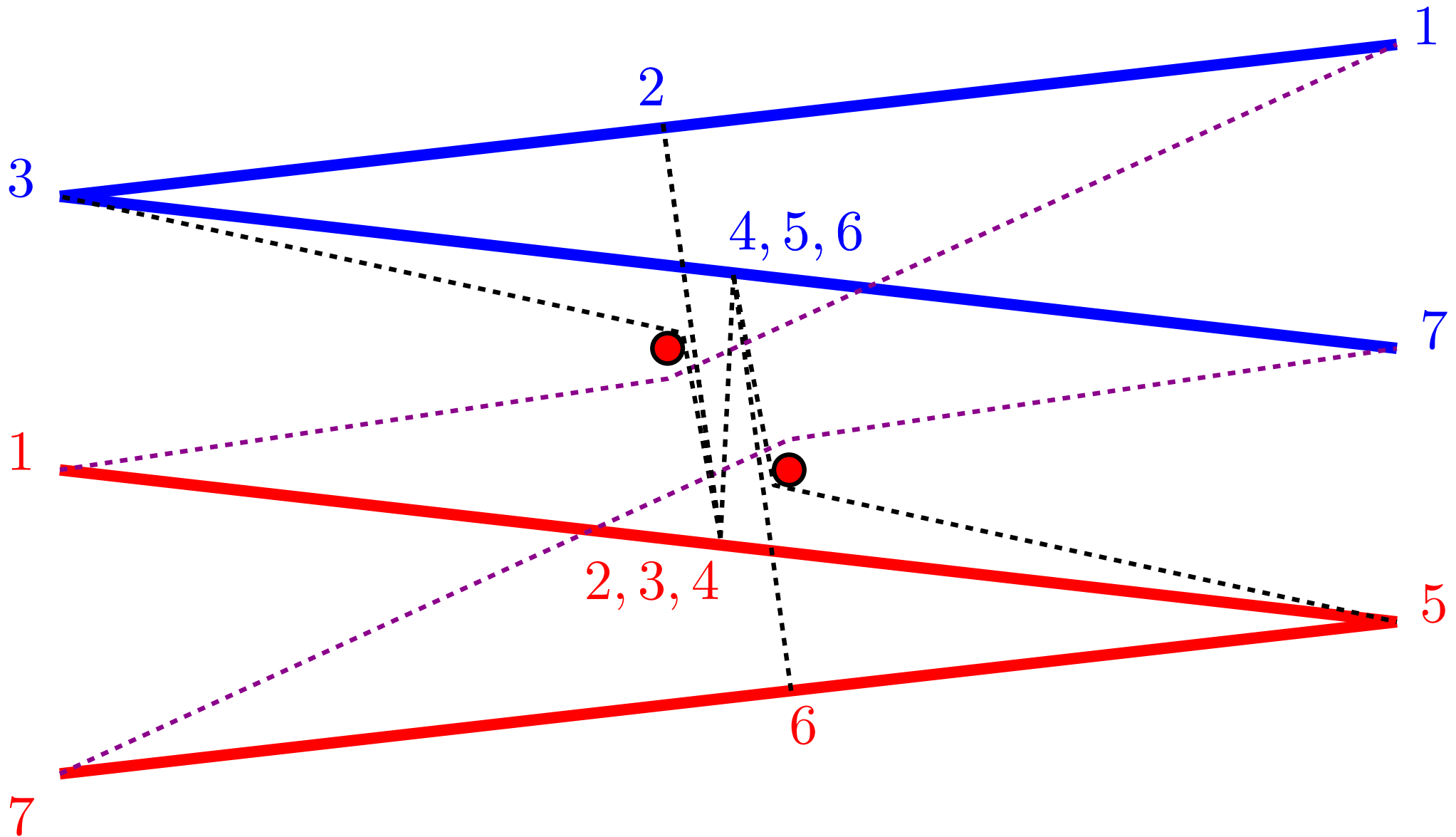
Example 2



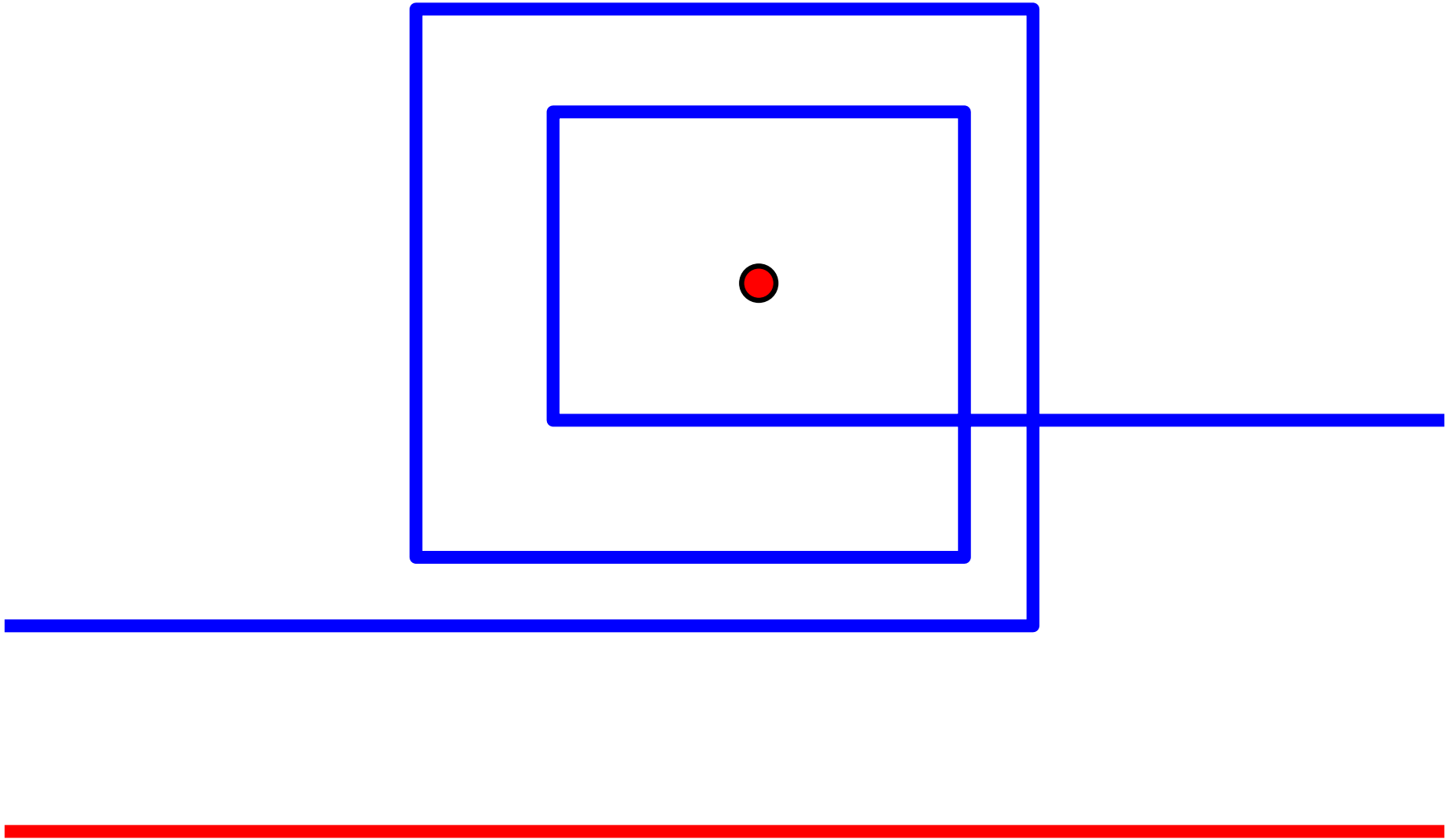
Example 2



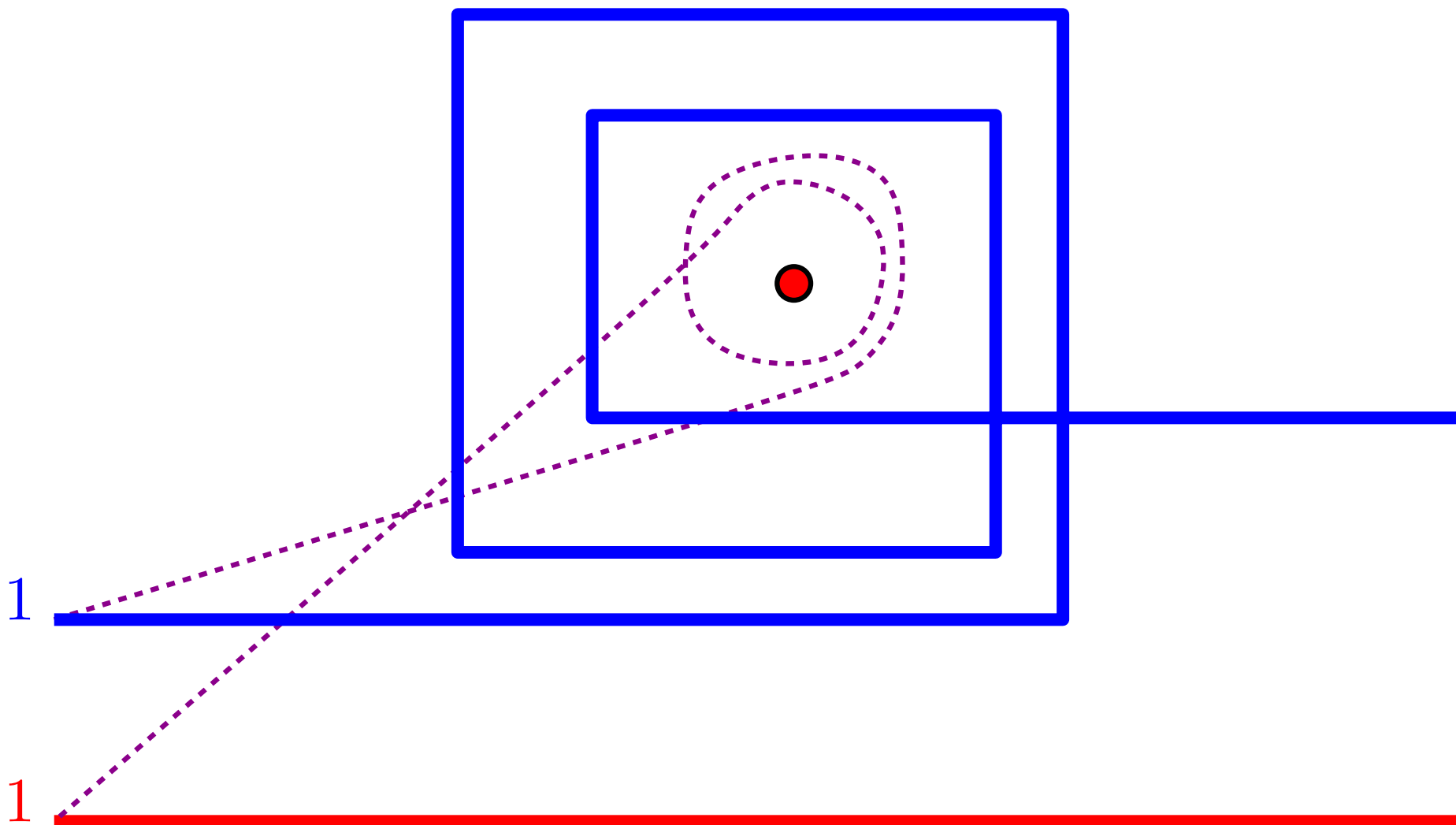
Example 2



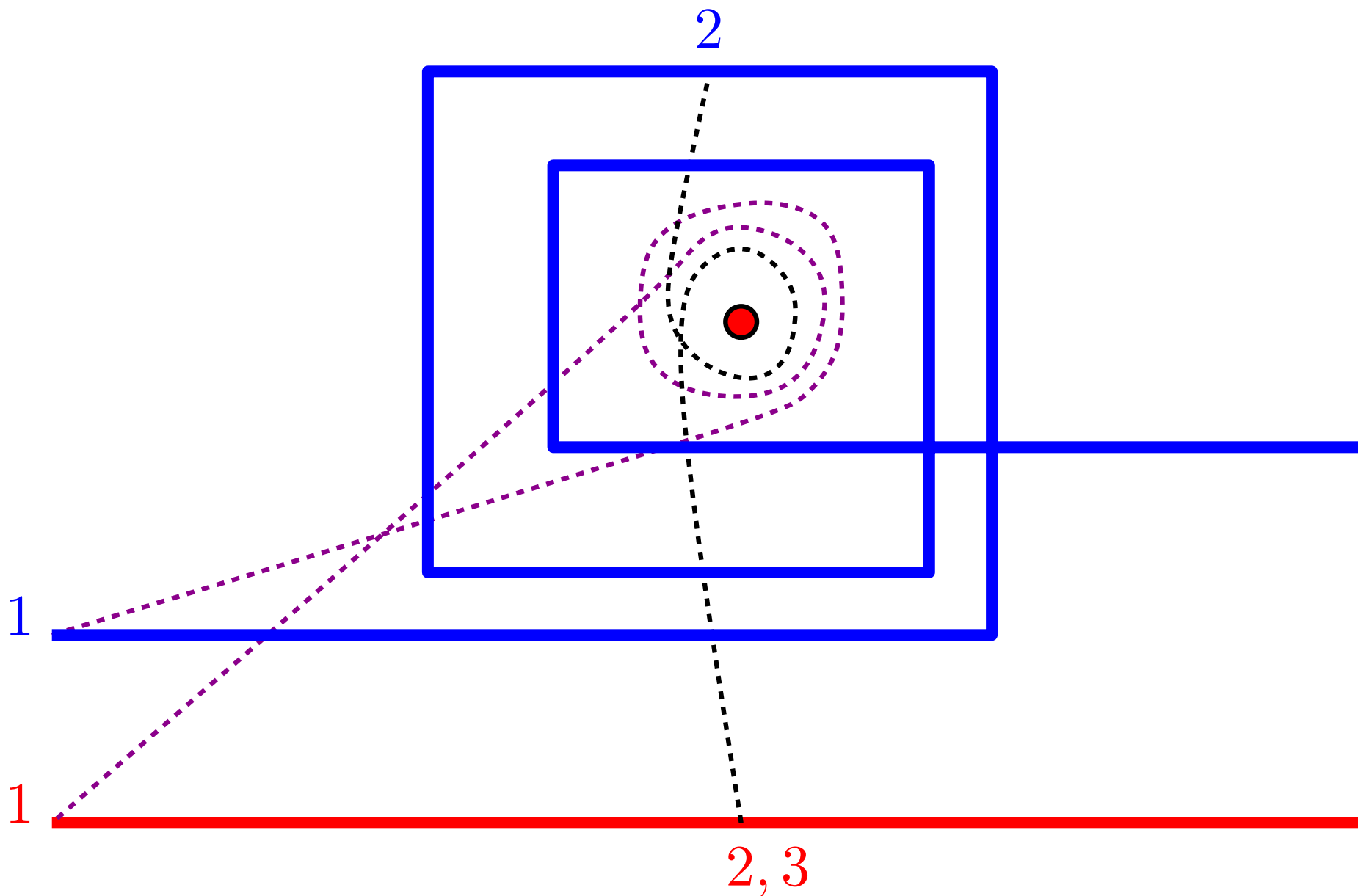
Example 3



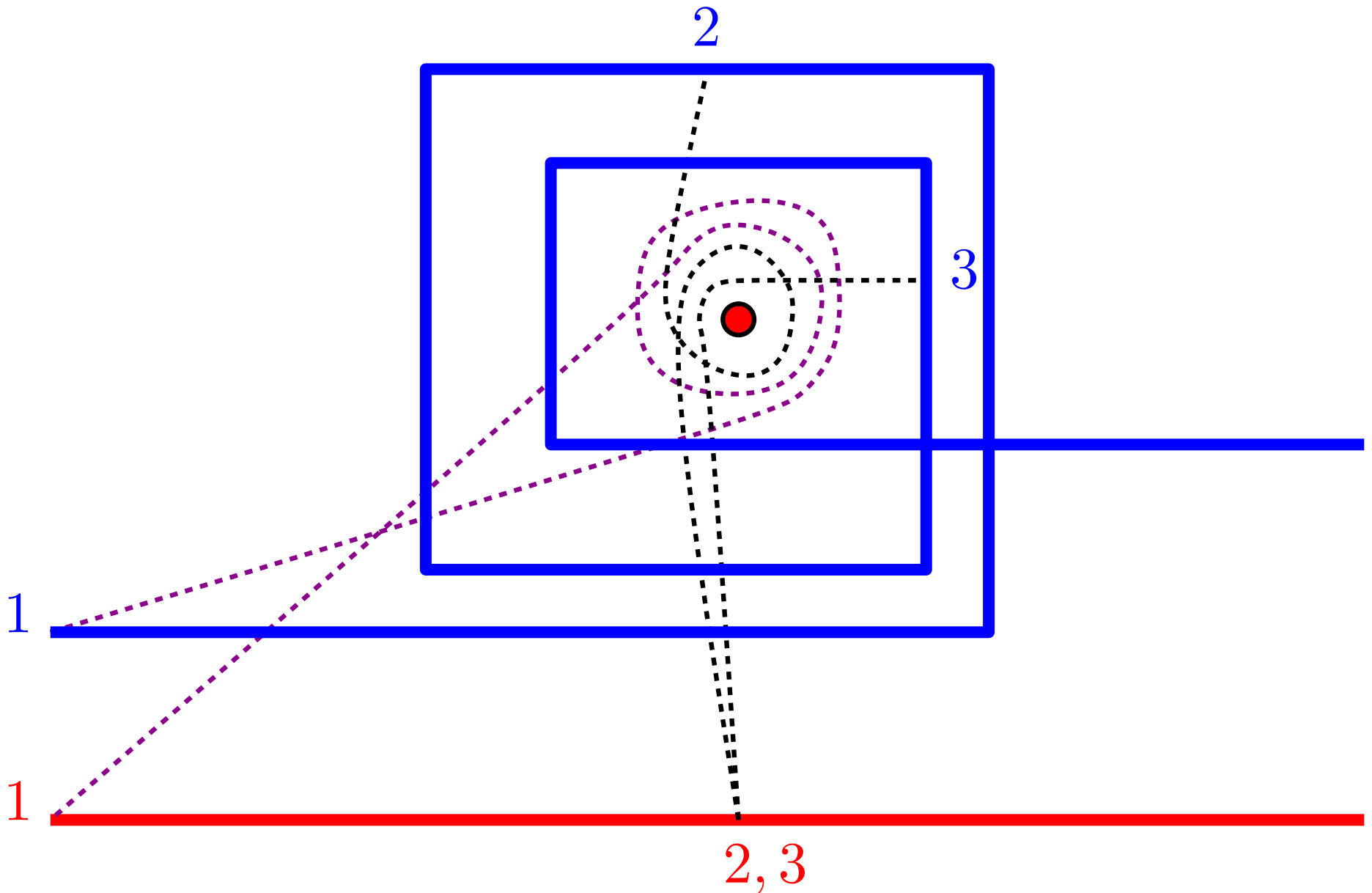
Example 3



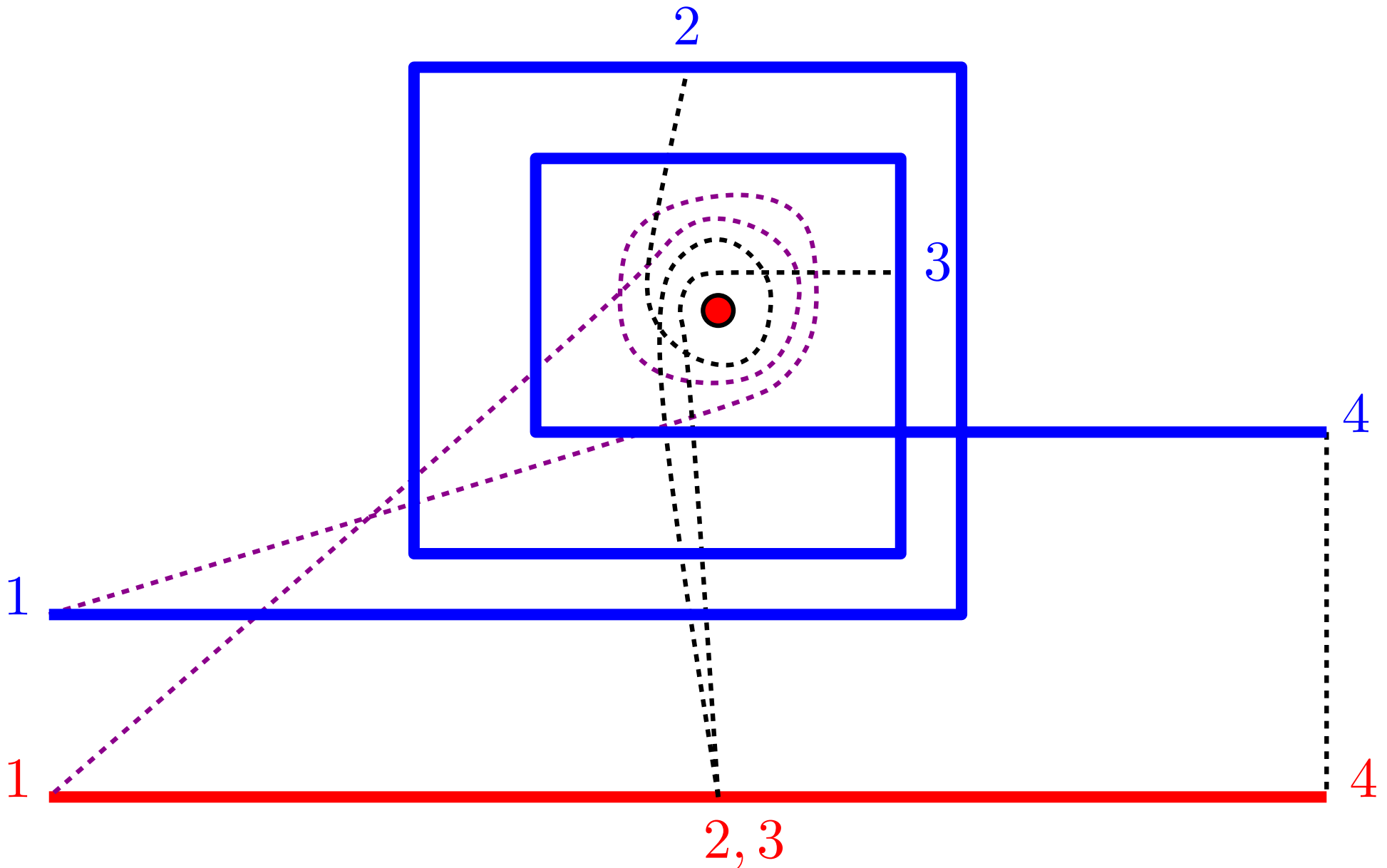
Example 3



Example 3



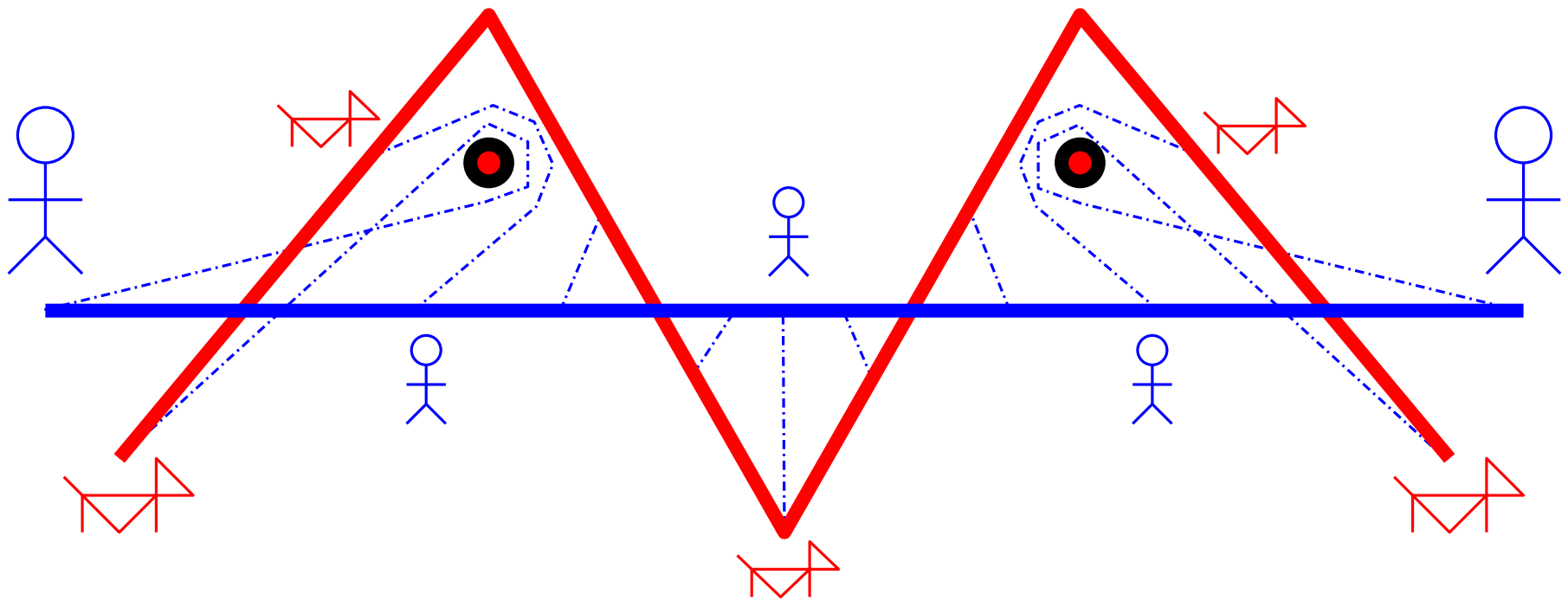
Example 3



Punctured plane

Let P be a set of points in the plane = obstacles

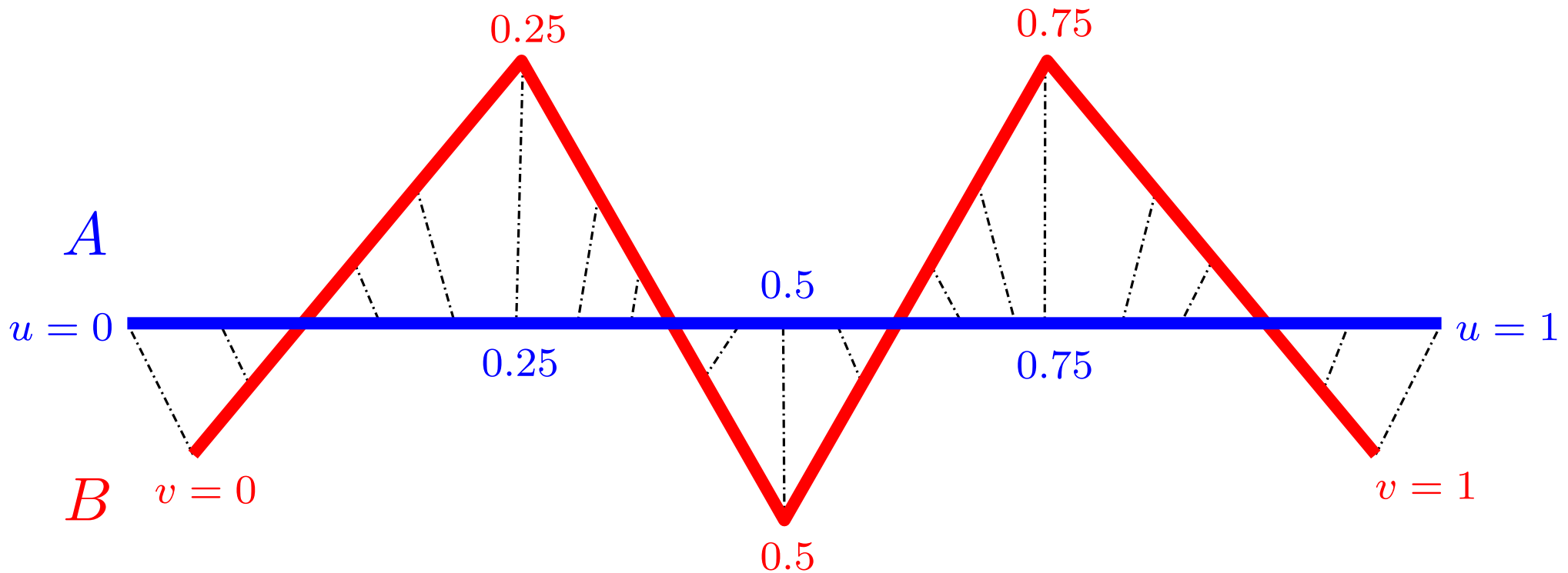
Punctured plane = $\mathbb{E}^2 \setminus P$



Leash must move continuously in the punctured plane, so cannot jump over obstacles

Leash map

Continuous function $\ell : [0, 1] \times [0, 1] \rightarrow \mathbb{E}^2 \setminus P$



ℓ gives re-parameterizations $\exists u, v$ of A and B such that $\ell(\cdot, t)$ is the leash at time t joining $A(u(t))$ and $B(v(t))$

Homotopic Fréchet distance

The **cost** of a leash map ℓ is the longest length of the leash at any time during the leash motion:

$$\text{cost}(\ell) := \sup_{t \in [0,1]} \{ \text{Length of } \ell(\cdot, t) \}$$

Homotopic Fréchet distance

The **cost** of a leash map ℓ is the longest length of the leash at any time during the leash motion:

$$\text{cost}(\ell) := \sup_{t \in [0,1]} \{ \text{Length of } \ell(\cdot, t) \}$$

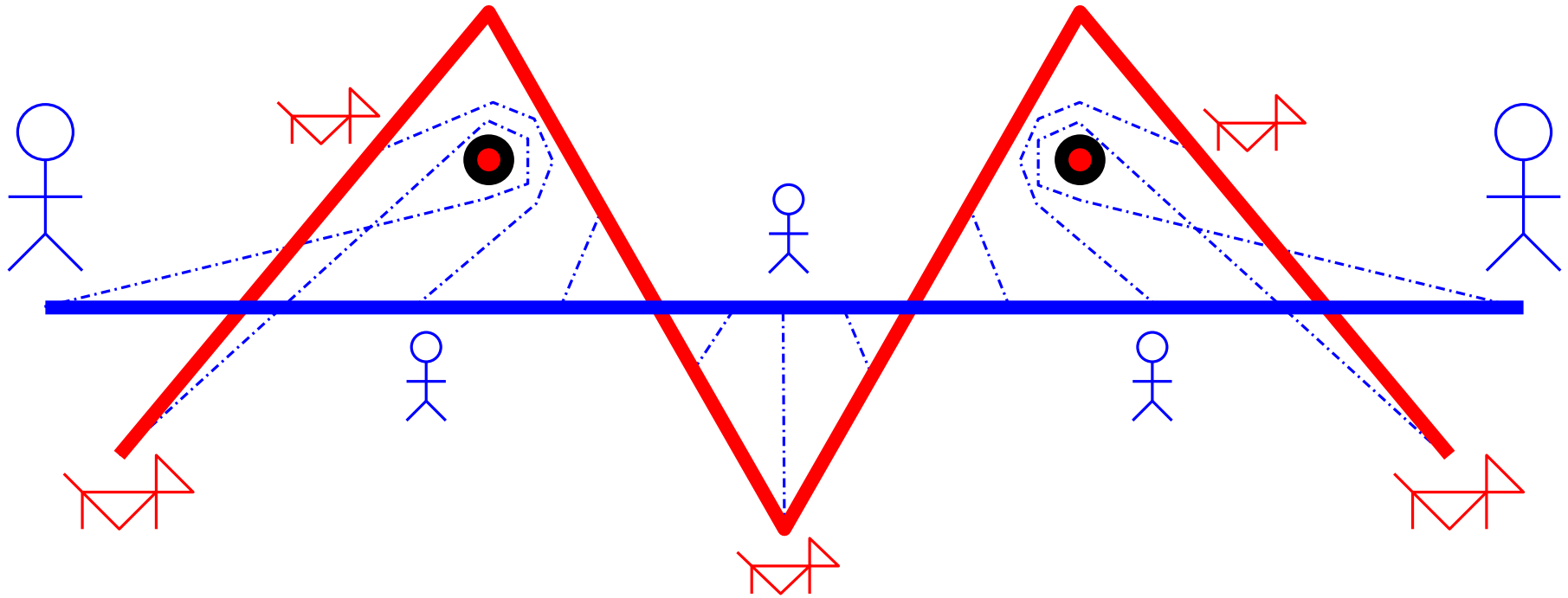
The **homotopic Fréchet distance** is the minimum cost of any leash map:

$$F(A, B) := \inf_{\text{leash map } \ell} \{ \text{cost}(\ell) \}$$

Relative homotopy

Curves $A, B = \text{continuous functions } [0, 1] \rightarrow \mathbb{E}^2$

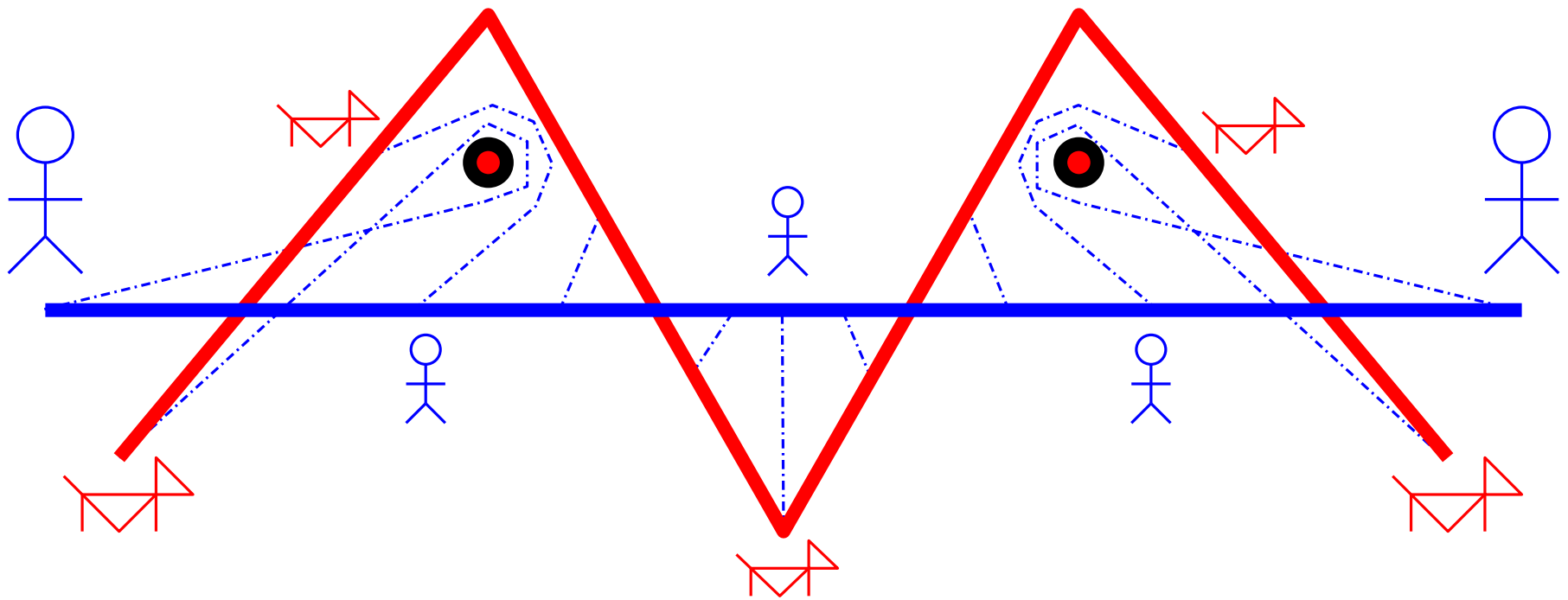
A **leash** is a curve joining a point of A and a point of B



Two leashes are **relatively homotopic** if they are continuously deformable in the punctured plane *while keeping their endpoints on the respective curves*

Relative homotopy

Every leash map ℓ_h in the punctured plane describes a set of leashes belonging to some relative homotopy class h



Homotopic Fréchet distance redux

Let h be a relative homotopy class

Homotopic Fréchet distance redux

Let h be a relative homotopy class

$$\ell_h : [0, 1] \times [0, 1] \rightarrow \mathbb{E}^2 \setminus P$$

such that $\ell_h(\cdot, t)$ is a leash in homotopy class h

Homotopic Fréchet distance redux

Let h be a relative homotopy class

$$\ell_h : [0, 1] \times [0, 1] \rightarrow \mathbb{E}^2 \setminus P$$

such that $\ell_h(\cdot, t)$ is a leash in homotopy class h

$$F_h(A, B) := \inf_{\ell_h} \sup_{t \in [0, 1]} \{ \text{Length of } \ell_h(\cdot, t) \}$$

Homotopic Fréchet distance redux

Let h be a relative homotopy class

$$\ell_h : [0, 1] \times [0, 1] \rightarrow \mathbb{E}^2 \setminus P$$

such that $\ell_h(\cdot, t)$ is a leash in homotopy class h

$$F_h(A, B) := \inf_{\ell_h} \sup_{t \in [0, 1]} \{ \text{Length of } \ell_h(\cdot, t) \}$$

Homotopic Fréchet distance

$$F(A, B) := \min_h \{ F_h(A, B) \}$$

Geodesic leashes

Lemma: There exists an optimum leash map such that the leash at every time is the shortest path in its homotopy class

Geodesic leashes

Lemma: There exists an optimum leash map such that the leash at every time is the shortest path in its homotopy class

Hence, w.l.o.g., $\ell_h(\cdot, t)$ is the (unique) shortest path in homotopy class h between its endpoints

Geodesic leashes

Lemma: There exists an optimum leash map such that the leash at every time is the shortest path in its homotopy class

Hence, w.l.o.g., $\ell_h(\cdot, t)$ is the (unique) shortest path in homotopy class h between its endpoints

We allow the leash to pass through obstacle points. A turning angle at every obstacle point uniquely identifies the homotopy class of the leash. Now, unique shortest paths exist in every homotopy class.

Key lemma

The optimum homotopy class h^* must contain a straight-line leash

Key lemma

The optimum homotopy class h^* must contain a straight-line leash

Step 1: Prove that there is no obstacle p such that the leash always bends around p during ℓ_{h^*}

Key lemma

The optimum homotopy class h^* must contain a straight-line leash

Step 1: Prove that there is no obstacle p such that the leash always bends around p during ℓ_{h^*}

Step 2: Prove that we can straighten the initial leash to a line segment

Our algorithm

We enumerate all homotopy classes h that contain a straight-line leash and compute $F_h(A, B)$

Our algorithm

We enumerate all homotopy classes h that contain a straight-line leash and compute $F_h(A, B)$

Let $N =$ total input complexity

There are $O(N^4)$ such homotopy classes h

$F_h(A, B)$ can be computed in $O(N^3 \log N)$ time

Computing F_h

Decision problem: Given a real $d \geq 0$, is $F_h(A, B) \leq d$?

Computing F_h

Decision problem: Given a real $d \geq 0$, is $F_h(A, B) \leq d$?

Observation: There are polynomially many **critical values** of d at which the answer may change from 'no' to 'yes'

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq d_{i+1} \leq d_{i+2} \leq \dots$$

Computing F_h

Decision problem: Given a real $d \geq 0$, is $F_h(A, B) \leq d$?

Observation: There are polynomially many **critical values** of d at which the answer may change from 'no' to 'yes'

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq d_{i+1} \leq d_{i+2} \leq \dots$$

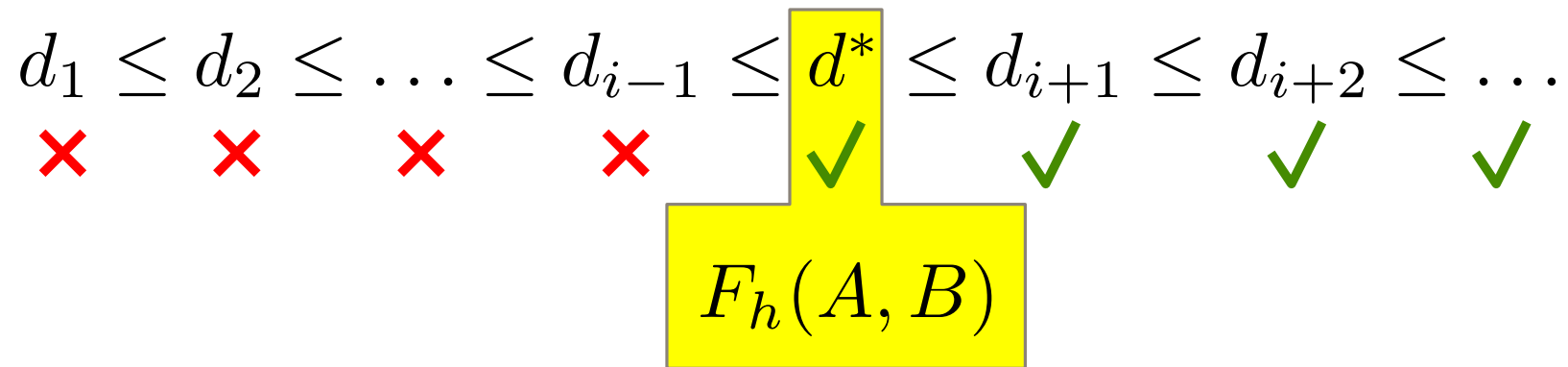
✗ ✗ ✗ ✗ ✓ ✓ ✓ ✓

Computing F_h

Decision problem: Given a real $d \geq 0$, is $F_h(A, B) \leq d$?

Observation: There are polynomially many **critical values** of d at which the answer may change from 'no' to 'yes'

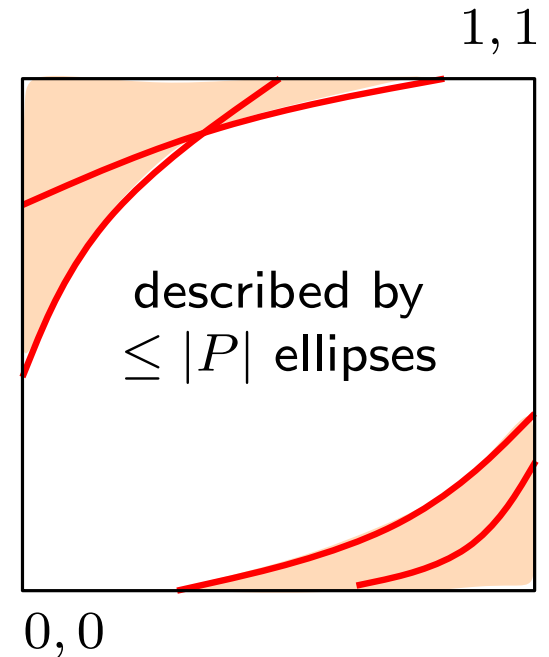
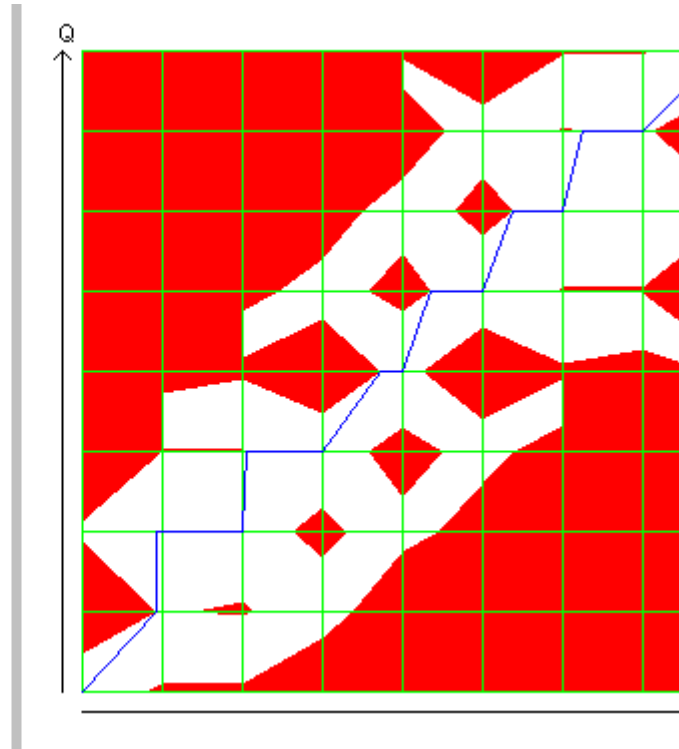
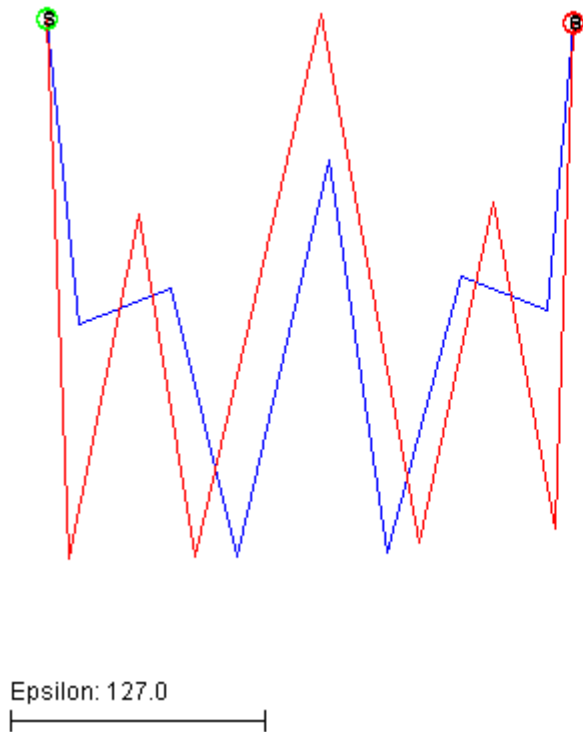
$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq d_{i+1} \leq d_{i+2} \leq \dots$$



$F_h(A, B)$

Goal: Find the smallest critical value d for which the answer above is 'yes'

Is $F_h \leq d$?



<http://www.cim.mcgill.ca/~stephane/cs507/Project.html>, Stéphane Pelletier, 2002

Is there a monotone path from $(0, 0)$ to (m, n) in free space?

Lemma: In each cell C_{ij} , the free space is convex

Parametric search

[Megiddo '83]

Let A_s be an algorithm to decide, given a critical value d_i , whether $F_h(A, B) \leq d_i$, with running time $O(T_s)$

Ask me later if you want me to describe A_s

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

✗ ✗ ✗ ✗ ✓ ✓ ✓ ✓

Parametric search

[Megiddo '83]

Let A_s be an algorithm to decide, given a critical value d_i , whether $F_h(A, B) \leq d_i$, with running time $O(T_s)$

Ask me later if you want me to describe A_s

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_s on input d^* , with d^* as a symbolic variable

Parametric search

[Megiddo '83]

Let A_s be an algorithm to decide, given a critical value d_i , whether $F_h(A, B) \leq d_i$, with running time $O(T_s)$

Ask me later if you want me to describe A_s

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_s on input d^* , with d^* as a symbolic variable

The control flow of A_s depends on comparisons of the form $d^* \leq d_j$ where d_j is a critical value

Each d_j is a distance, i.e., a quadratic function of input coordinates.

Parametric search

[Megiddo '83]

Let A_s be an algorithm to decide, given a critical value d_i , whether $F_h(A, B) \leq d_i$, with running time $O(T_s)$

Ask me later if you want me to describe A_s

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_s on input d^* , with d^* as a symbolic variable

The control flow of A_s depends on comparisons of the form $d^* \leq d_j$ where d_j is a critical value

Each d_j is a distance, i.e., a quadratic function of input coordinates.

$d^* \leq d_j$? Run A_s on input d_j , in $O(T_s)$ time

Parametric search

[Megiddo '83]

Let A_s be an algorithm to decide, given a critical value d_i , whether $F_h(A, B) \leq d_i$, with running time $O(T_s)$

Ask me later if you want me to describe A_s

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_s on input d^* , with d^* as a symbolic variable

The control flow of A_s depends on comparisons of the form $d^* \leq d_j$ where d_j is a critical value

Each d_j is a distance, i.e., a quadratic function of input coordinates.

$d^* \leq d_j$? Run A_s on input d_j , in $O(T_s)$ time

Total running time = $O(T_s^2)$

Parametric search on steroids [M'83]

Let A_p be a **parallel** algorithm to decide, given d_i , whether $F_h(A, B) \leq d_i$, with parallel running time $O(T_p)$ on k processors

Ask me later if you want me to describe A_p

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

Parametric search on steroids [M'83]

Let A_p be a **parallel** algorithm to decide, given d_i , whether $F_h(A, B) \leq d_i$, with parallel running time $O(T_p)$ on k processors

Ask me later if you want me to describe A_p

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_p **sequentially** on input d^* , with d^* as a symbolic variable

Parametric search on steroids [M'83]

Let A_p be a **parallel** algorithm to decide, given d_i , whether $F_h(A, B) \leq d_i$, with parallel running time $O(T_p)$ on k processors

Ask me later if you want me to describe A_p

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_p **sequentially** on input d^* , with d^* as a symbolic variable

The control flow of A_p depends on comparisons of the form $d^* \leq d_j$ where d_j is a critical value

Parametric search on steroids [M'83]

Let A_p be a **parallel** algorithm to decide, given d_i , whether $F_h(A, B) \leq d_i$, with parallel running time $O(T_p)$ on k processors

Ask me later if you want me to describe A_p

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

✗ ✗ ✗ ✗ ✓ ✓ ✓ ✓

We simulate A_p **sequentially** on input d^* , with d^* as a symbolic variable

The control flow of A_p depends on comparisons of the form $d^* \leq d_j$ where d_j is a critical value

In each parallel stage, do binary search on $O(k)$ values d_j

Parametric search on steroids [M'83]

Let A_p be a **parallel** algorithm to decide, given d_i , whether $F_h(A, B) \leq d_i$, with parallel running time $O(T_p)$ on k processors

Ask me later if you want me to describe A_p

$$d_1 \leq d_2 \leq \dots \leq d_{i-1} \leq d^* \leq \dots \leq d_j \leq \dots$$

× × × × ✓ ✓ ✓ ✓

We simulate A_p **sequentially** on input d^* , with d^* as a symbolic variable

The control flow of A_p depends on comparisons of the form $d^* \leq d_j$ where d_j is a critical value

In each parallel stage, do binary search on $O(k)$ values d_j

Total running time = $O(T_s T_p \log k + k T_p)$ *better than $O(T_s^2)$*

Open: On a convex polyhedron

Leash is not always a geodesic!

e.g., leash must have enough slack to cross over a vertex
(a 'mountain')

Challenge: Characterize an optimum leash map

Thank you!

Extra slides