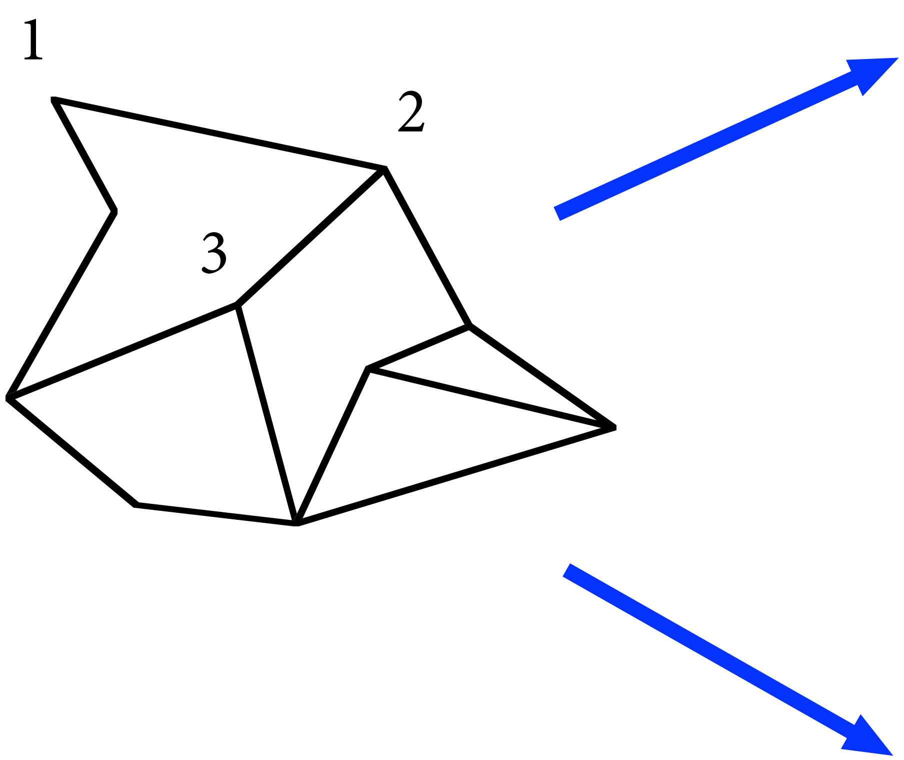
Reconfiguration of Graph Drawings

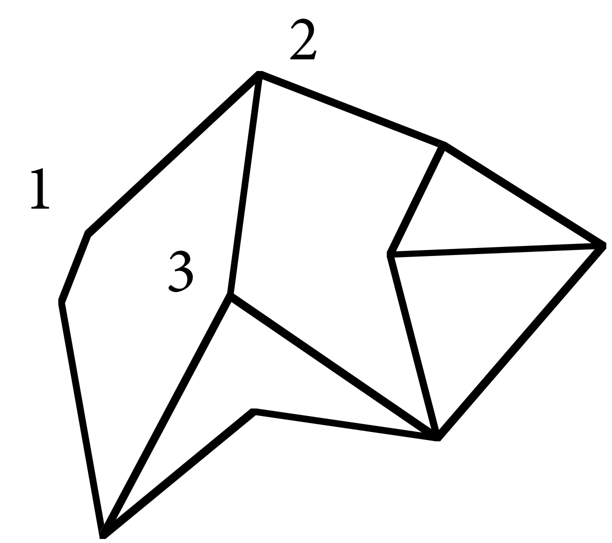
Anna Lubiw

University of Waterloo

Reconfiguring a Graph Drawing

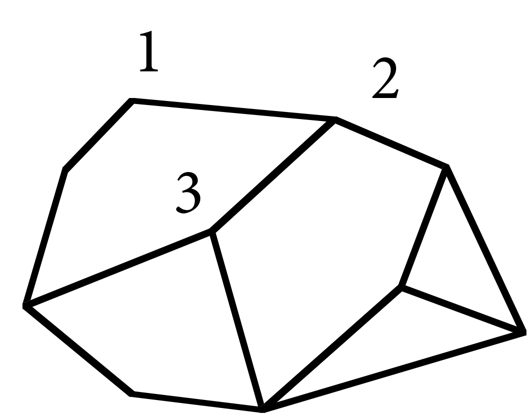
Given a planar drawing of a graph, transform it,
preserving planarity + other structure





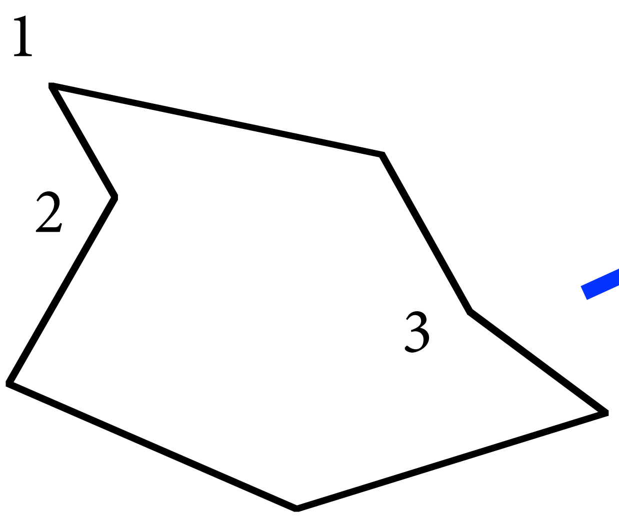
Transform to a specific target
or to attain some structure.

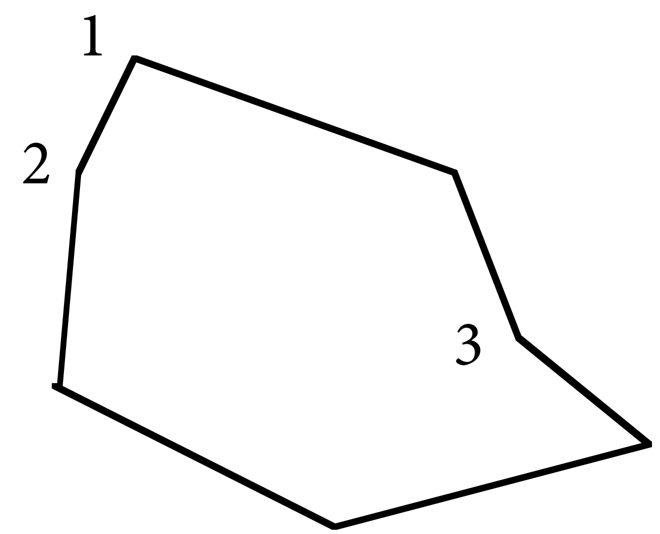
convex faces



Reconfiguring a Graph Drawing

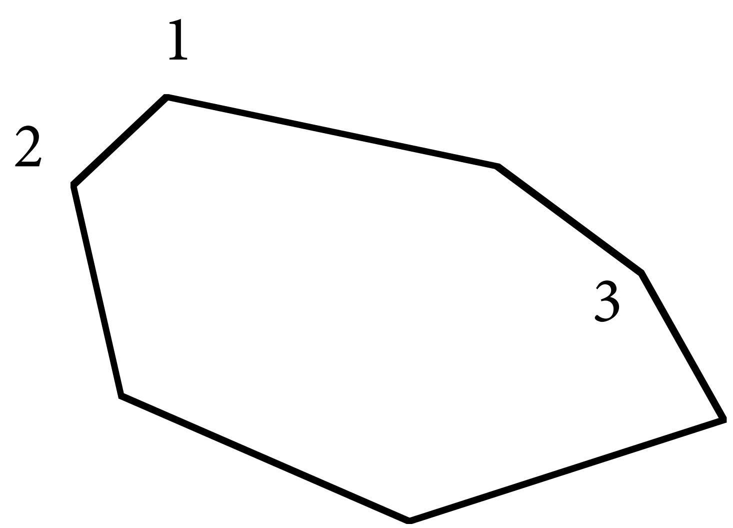
Given a planar drawing of a cycle ~~graph~~, transform it,
preserving planarity + ~~other structure~~ edge lengths





Transform to a specific target
or to attain some structure.

convex faces



Carpenter's Rule

[Connelly, Demaine, Rote, 2003]

Outline

- transform planar graph drawing to specific target ("morphing")
    - straight line edges
    - edges are [orthogonal] poly-lines
    - morphing preserving lengths, directions, etc.

- transform planar graph drawing to attain convex faces
    - polygon, with increasing visibility

Morphing Graph Drawings

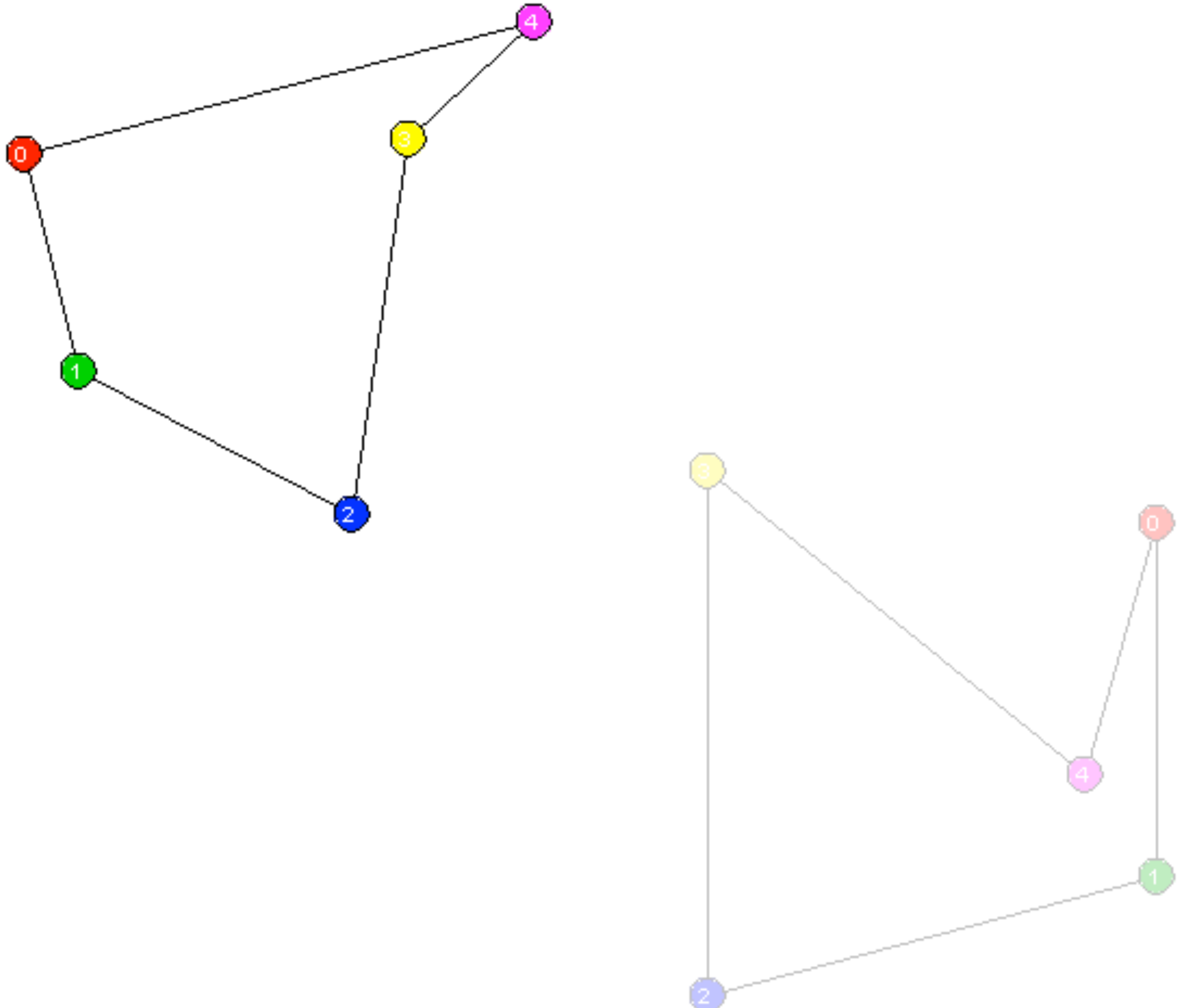
Definition. Let P and Q be two drawings of graph G. A morph from P to Q is a continuous family of drawings $P(t)$, $0 \leq t \leq 1$ with $P(0) = P$ and $P(1) = Q$.

Erten, Kobourov, Pitta,

GMorph, 2004

Morphing Graph Drawings

Definition. Let P and Q be two drawings of graph G. A morph from P to Q is a continuous family of drawings P(t), $0 \le t \le 1$ with P(0) = P and P(1) = Q.

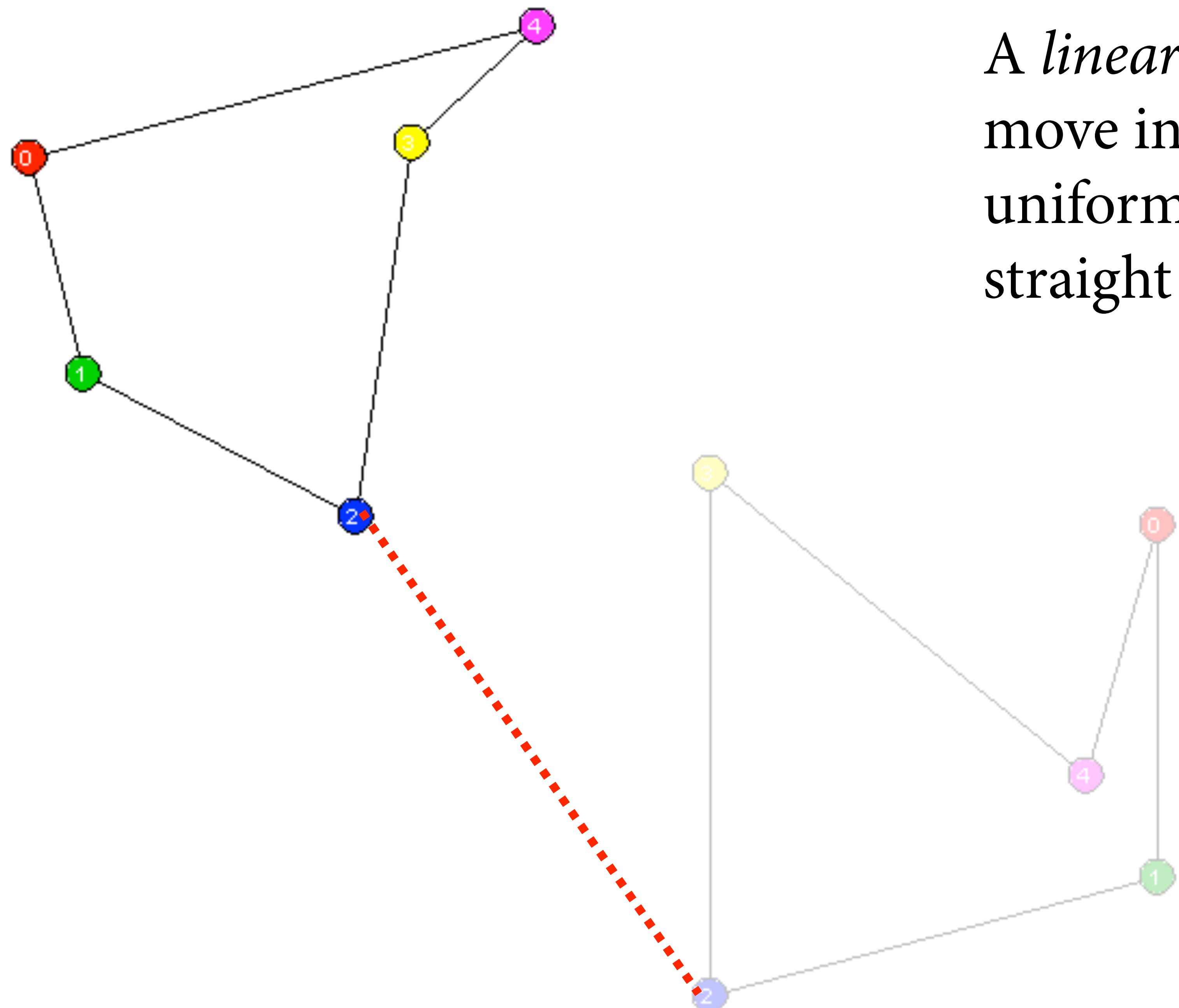


Erten, Kobourov, Pitta,

GMorph, 2004

Morphing Graph Drawings

Definition. Let P and Q be two drawings of graph G. A morph from P to Q is a continuous family of drawings P(t), $0 \le t \le 1$ with P(0) = P and P(1) = Q.

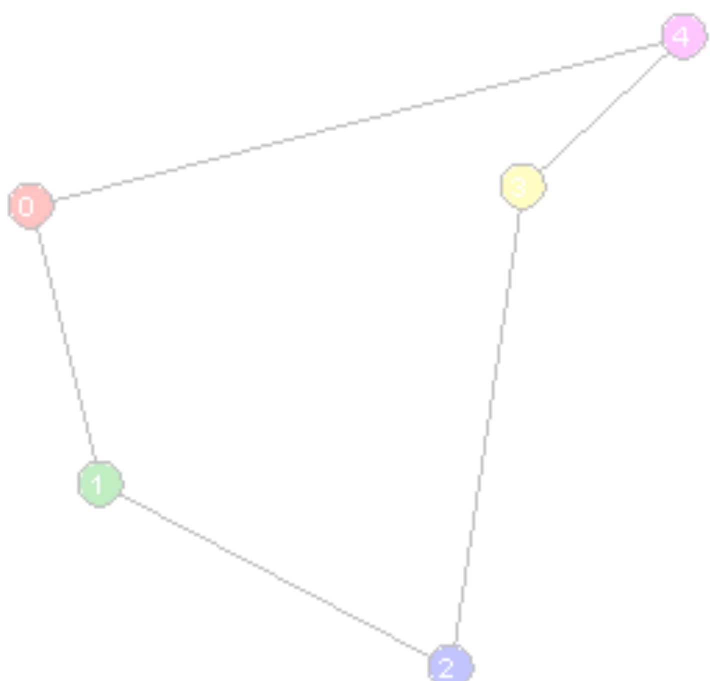


A *linear morph* — vertices move in straight lines at uniform speed, edges are straight line segments.

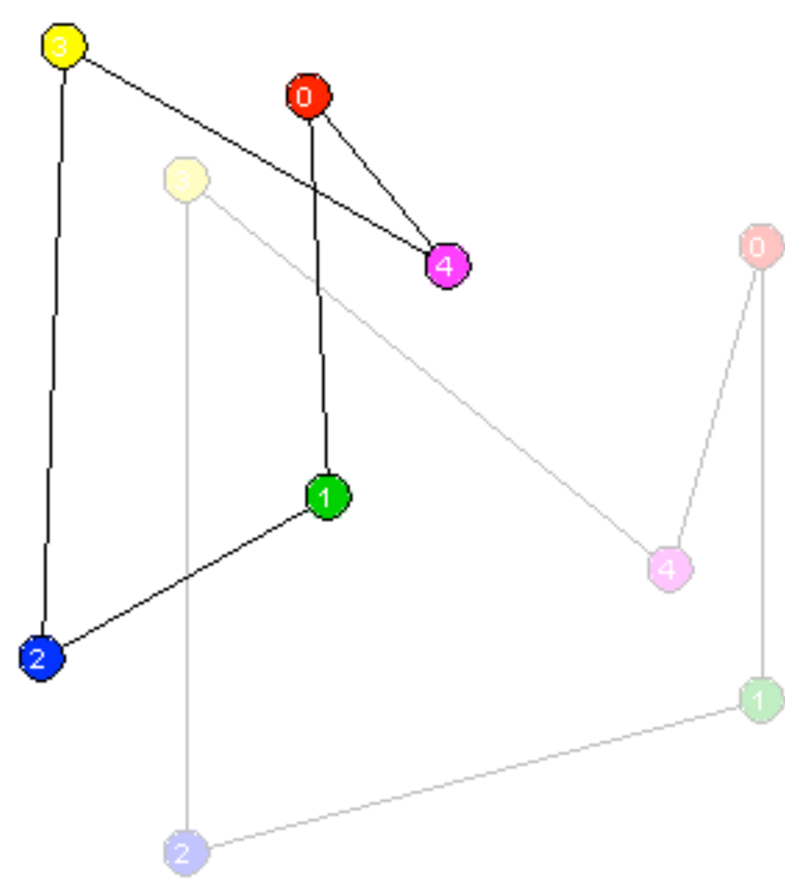
Erten, Kobourov, Pitta, GMorph, 2004

Morphing Graph Drawings

Definition. Let P and Q be two drawings of graph G. A morph from P to Q is a continuous family of drawings P(t), $0 \leq t \leq 1$ with P(0) = P and P(1) = Q.



A *linear morph* — vertices move in straight lines at uniform speed, edges are straight line segments.



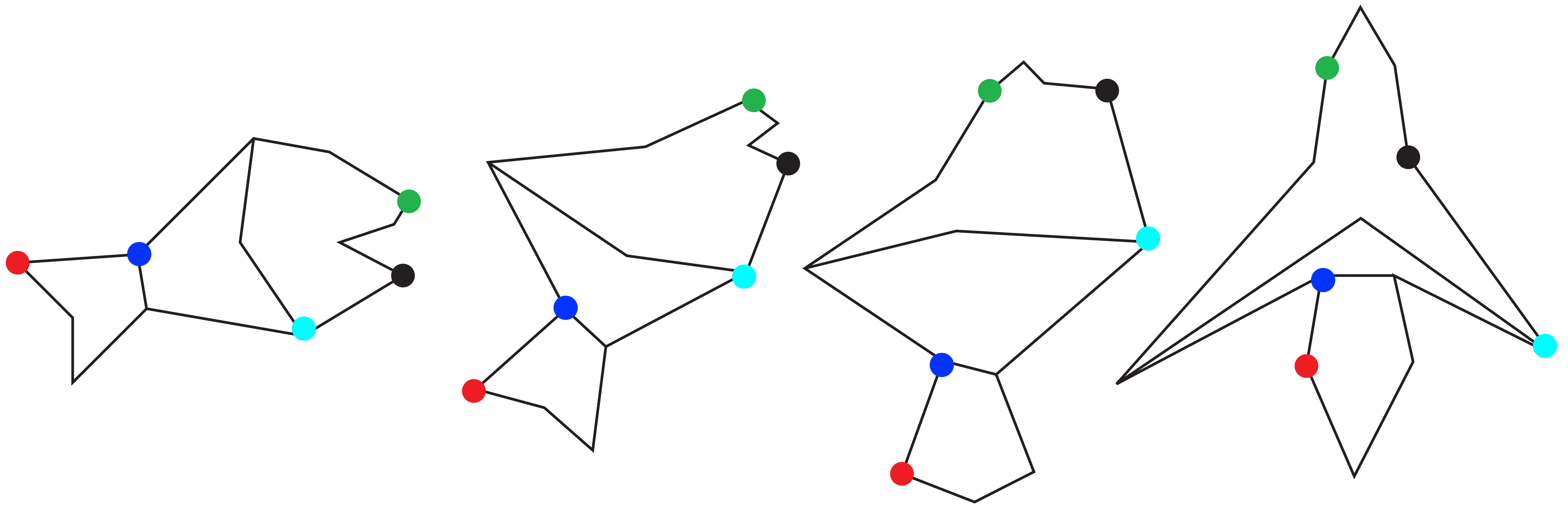
NOT a planar morph.

Erten, Kobourov, Pitta, GMorph, 2004

Planar Morphing

Every intermediate drawing is planar. Note that P =P(0) and Q =P(1) must represent the same embedding)

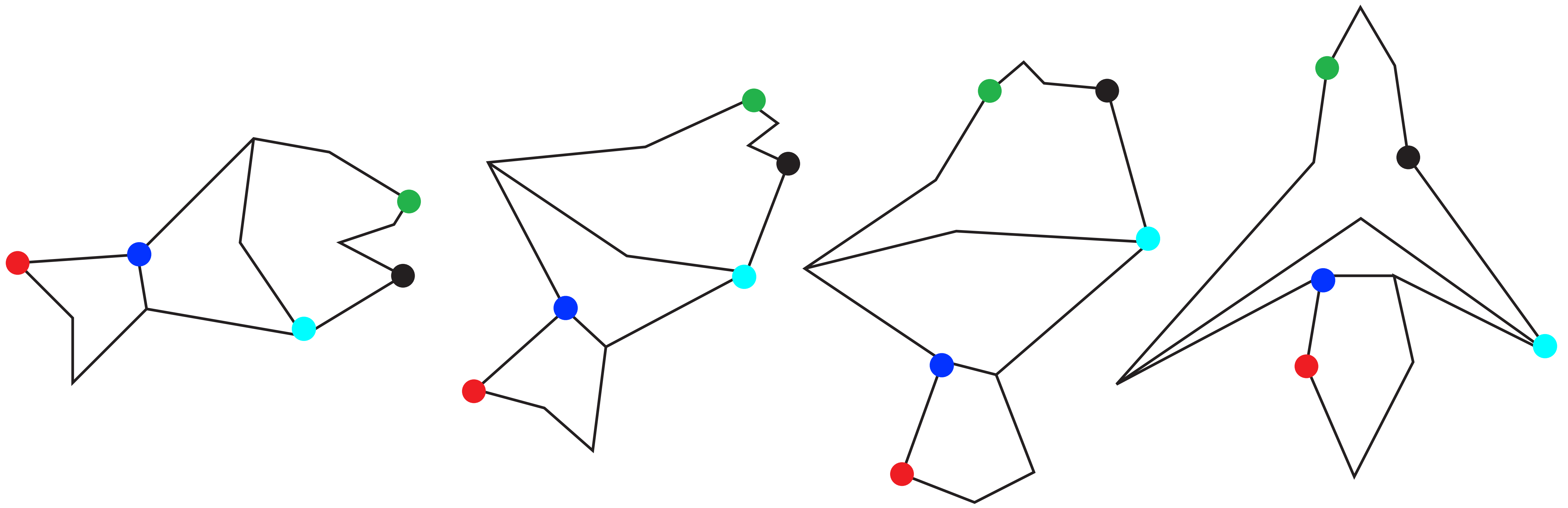
Given two planar drawings of a graph, find a [straight-line] planar morph between them.



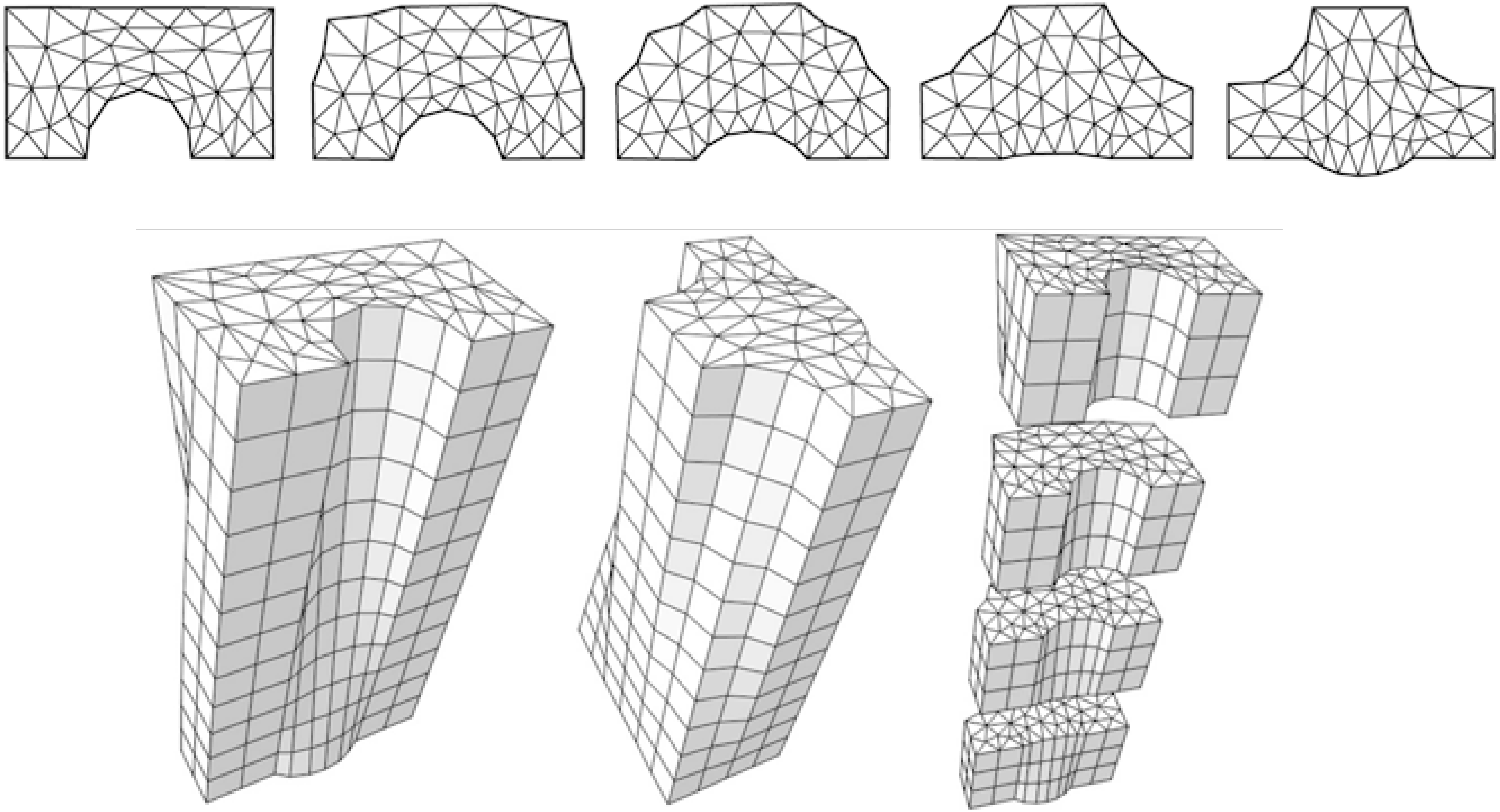
Planar Morphing

Every intermediate drawing is planar. Note that P =P(0) and Q =P(1) must represent the same embedding)

Given two planar drawings of a graph, find a (straight-line) planar morph between them.



Application: 2D Morphing as 3D Shape Reconstruction



Surazhsky, Gotsman, High quality compatible triangulations, 2002

Morphing Planar Straight-Line Graph Drawings

Planar Graph Drawing

- existence of straight-line drawing [Wagner, Koebe 1936, Fáry 1948, Stein 1951]
- an algorithm [Tutte 1963]
- polynomial size grid [de Fraysseix, Pach, Pollack; Schnyder 1990]

Planar Graph Morphing

- existence of morph preserving straight-line [Cairns 1944, Thomassen 1983]
- an algorithm [Floater and Gotsman 1999, Gotsman and Surazhsky 2001]
- polynomial size ??

Morphing Planar Straight-Line Graph Drawings

Planar Graph Drawing

➡ • existence of straight-line drawing [Wagner, Koebe 1936, Fáry 1948, Stein 1951]

• an algorithm [Tutte 1963]

• polynomial size grid [de Fraysseix, Pach, Pollack; Schnyder 1990]

Planar Graph Morphing

➡ • existence of morph preserving straight-line [Cairns 1944, Thomassen 1983]

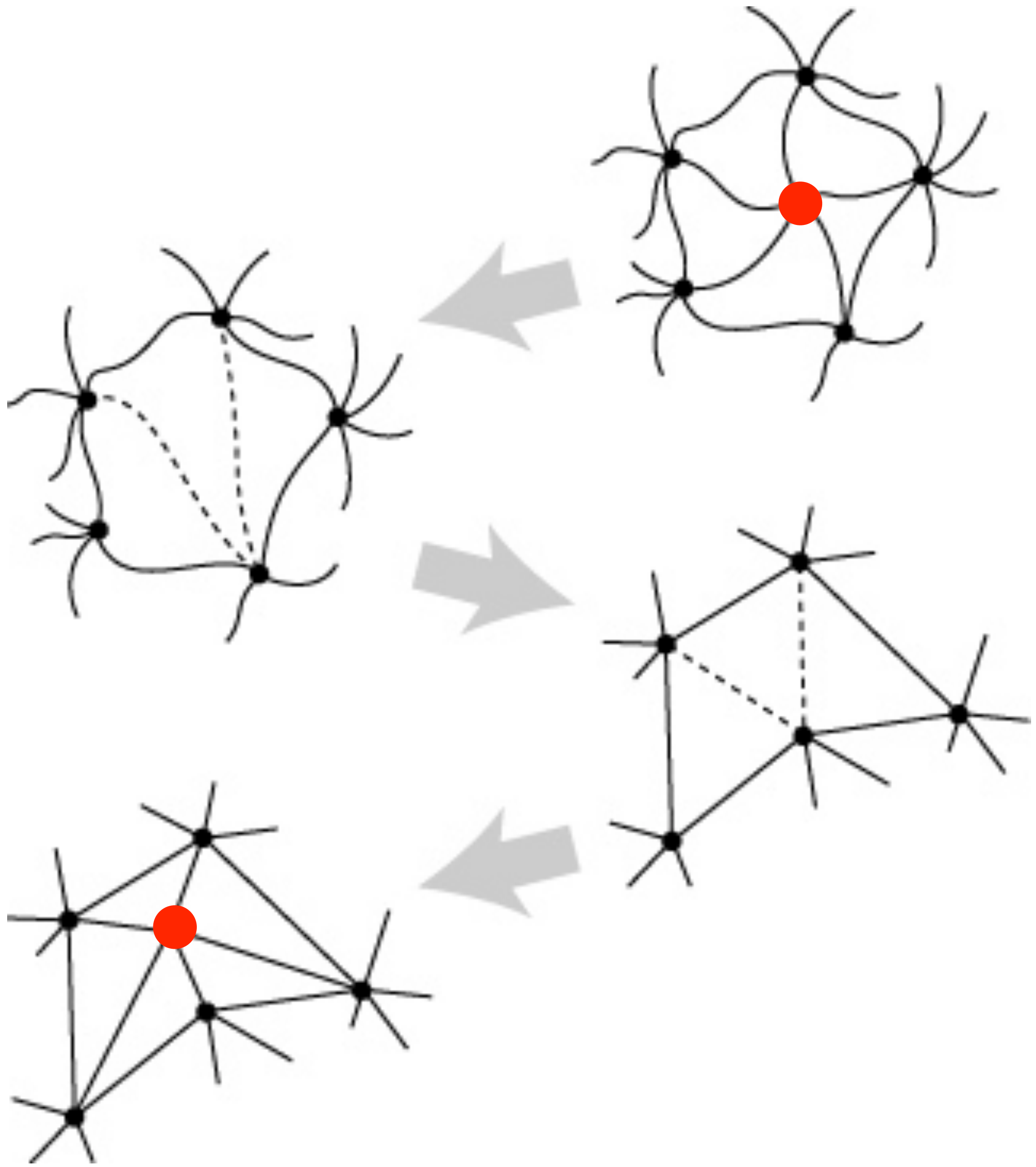
• an algorithm [Floater and Gotsman 1999, Gotsman and Surazhsky 2001]

• polynomial size ??

Planar Graph Drawing: Existence

Every planar graph has a drawing with straight lines for edges.

[Wagner, Koebe 1936, Fáry 1948, Stein 1951]



remove a vertex of degree ≤ 5

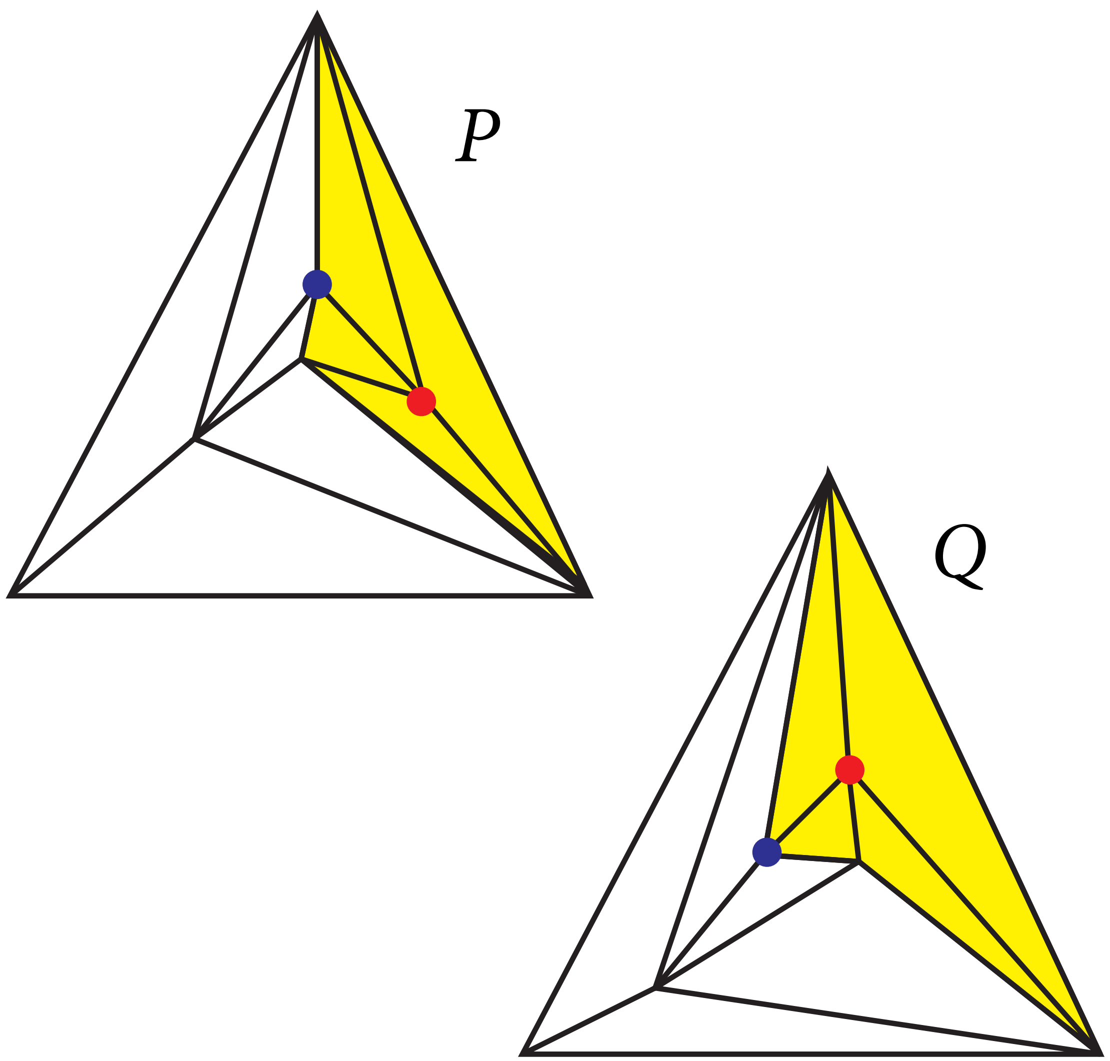
apply induction

Fact: a ≤ 5-gon has a point that sees all vertices

replace missing vertex

Planar Graph Morphing: Existence

There is a planar morph between any two straight-line embeddings of a triangulation. [Cairns 1944]



in P contract a vertex v of degree ≤ 5 to neighbour u that sees same

Fact: a ≤ 5-gon has a ~~point~~ vertex that sees all vertices

Complication: cannot use same u in Q

extra recursive call to make face convex
⇒ exponential number of steps

Extended to planar graphs [Thomassen 1983].

Morphing Planar Straight-Line Graph Drawings

Planar Graph Drawing

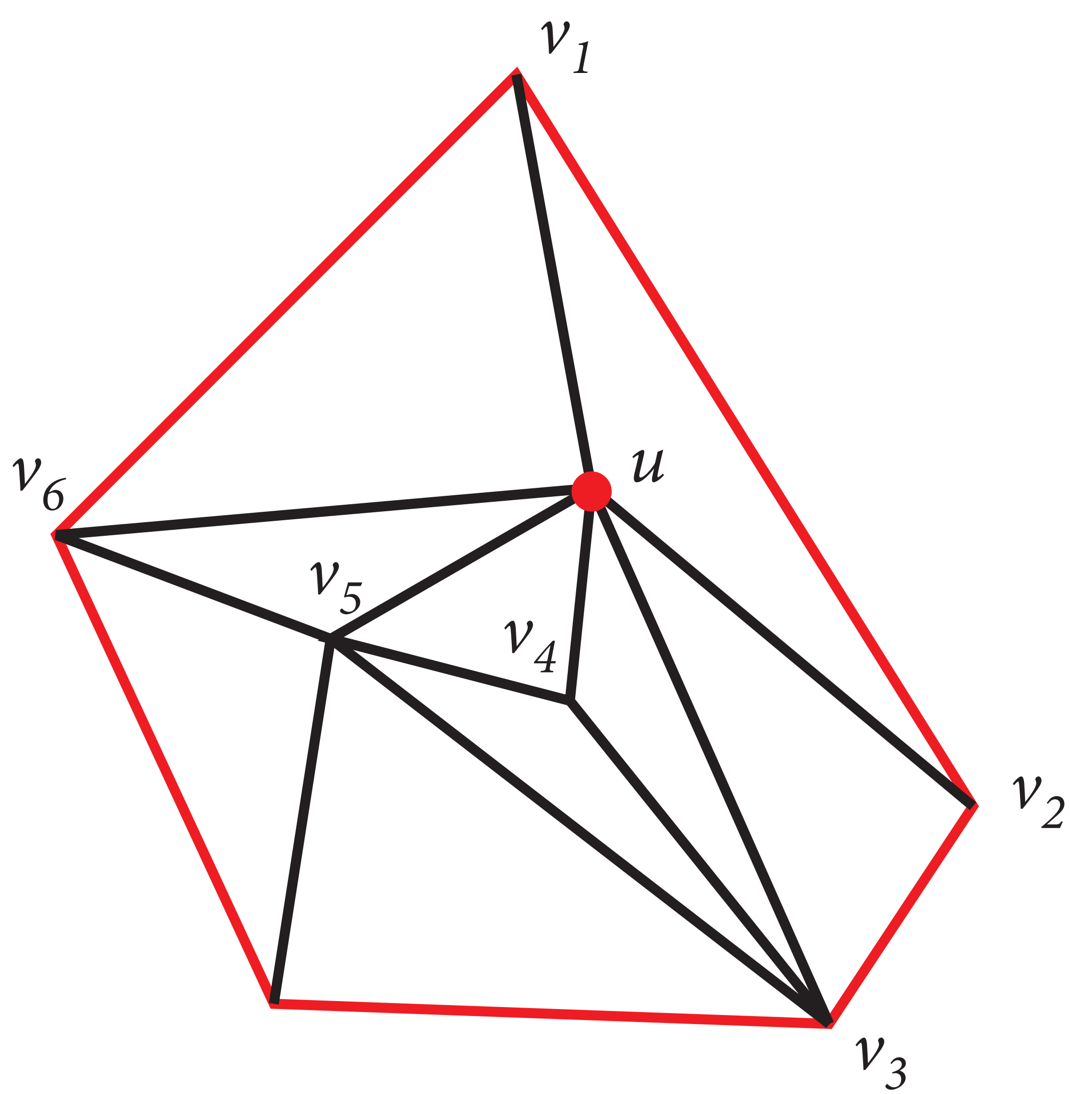
- existence of straight-line drawing [Wagner, Koebe 1936, Fáry 1948, Stein 1951]
- ➡ an algorithm [Tutte 1963]
- polynomial size grid [de Fraysseix, Pach, Pollack; Schnyder 1990]

Planar Graph Morphing

- existence of morph preserving straight-line [Cairns 1944, Thomassen 1983]
- ➡ an algorithm [Floater and Gotsman 1999, Gotsman and Surazhsky 2001]
- polynomial size ??

Planar Graph Drawing: Algorithm

Can find a planar straight line drawing in polynomial time by solving a linear system to find coordinates. [Tutte 1963]



fix convex outer face

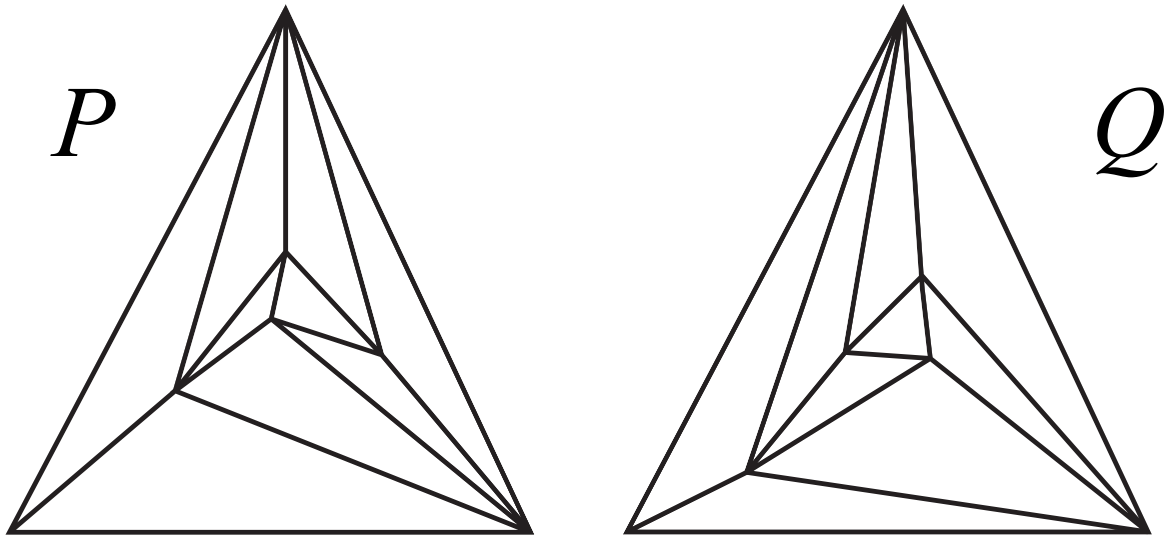
one equation for each interior vertex:

$$(x(u), y(u)) = \frac{1}{6} \sum_{i=1}^{6} (x(v_i), y(v_i))$$

Planar Graph Morphing: Algorithm

A morphing algorithm that computes a "snapshot" at any time t, $0 \leq t \leq 1$.

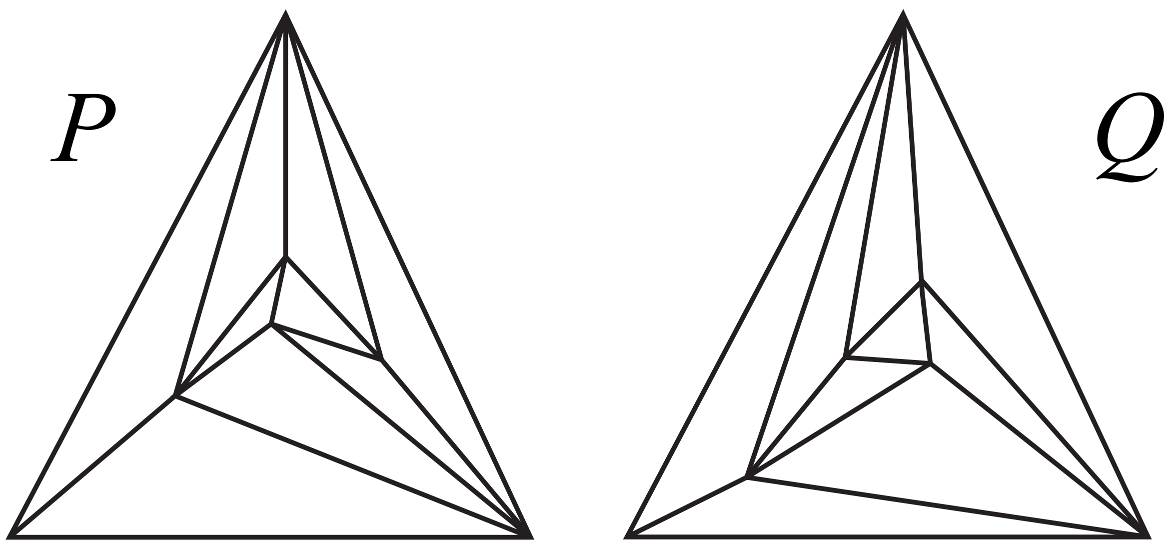
[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]

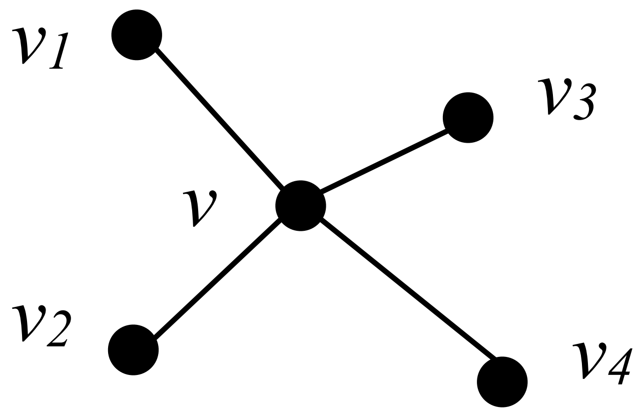


Planar Graph Morphing: Algorithm

A morphing algorithm that computes a "snapshot" at any time t, 0 ≤ t ≤ 1.

[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]



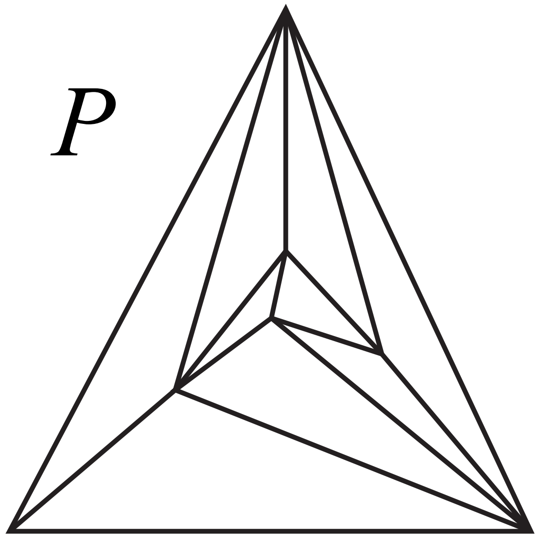


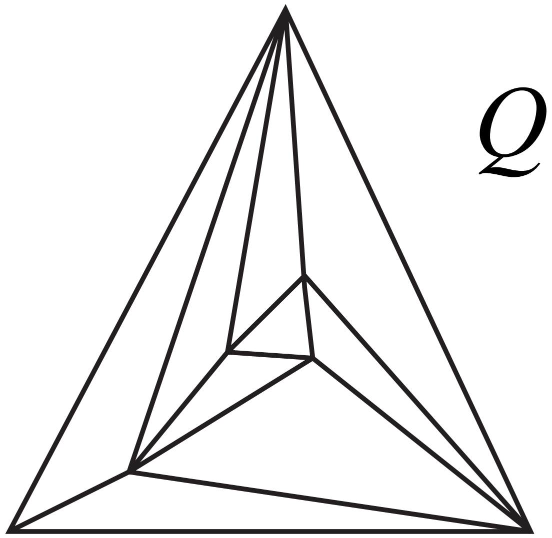
$$(x(v), y(v)) = \sum_{i=1}^{4} \frac{1}{4}(x(v_i), y(v_i))$$

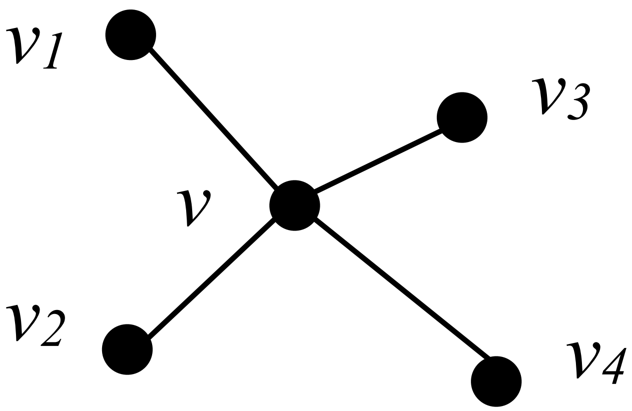
Planar Graph Morphing: Algorithm

A morphing algorithm that computes a "snapshot" at any time t, 0 ≤ t ≤ 1.

[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]







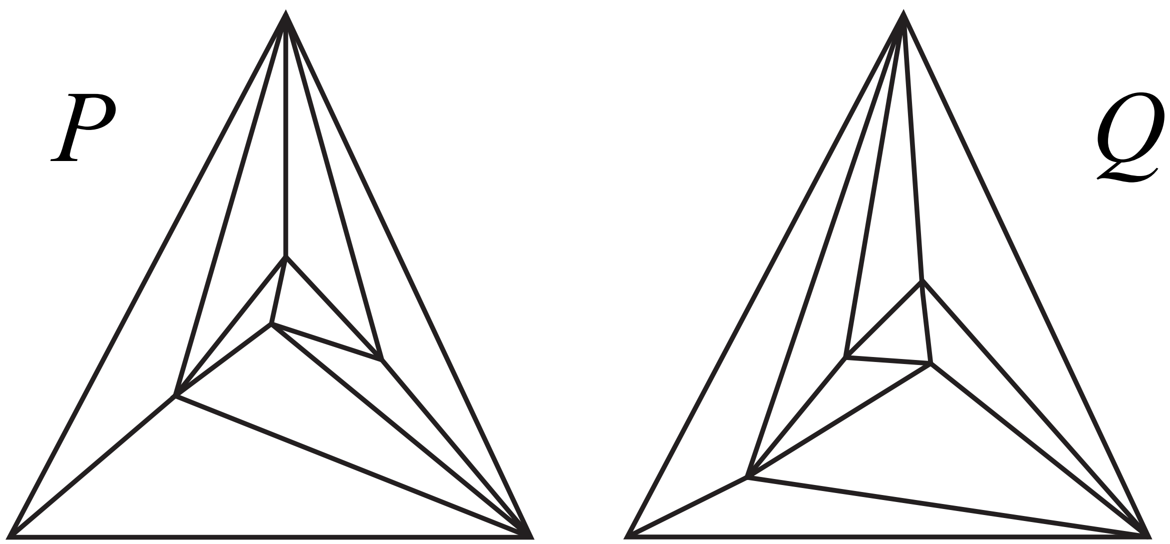
$$(x(v), y(v)) = \sum_{i=1}^{4} \frac{1}{4}(x(v_i), y(v_i))$$

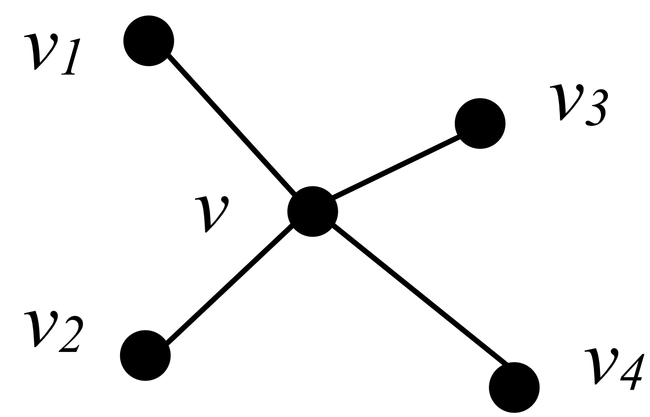
$m(v, v_i)$

Planar Graph Morphing: Algorithm

A morphing algorithm that computes a "snapshot" at any time t, $0 \leq t \leq 1$.

[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]





$$M_P = [m(u,v)]_{n \times n}$$

$$(x(v), y(v)) = \sum_{i=1}^{4} \frac{1}{4} (x(v_i), y(v_i))$$

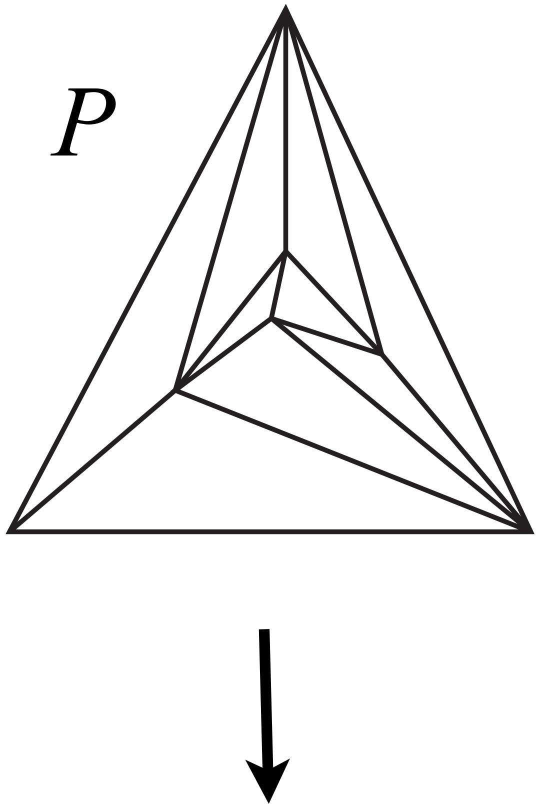
$m(v, v_i)$

Planar Graph Morphing: Algorithm

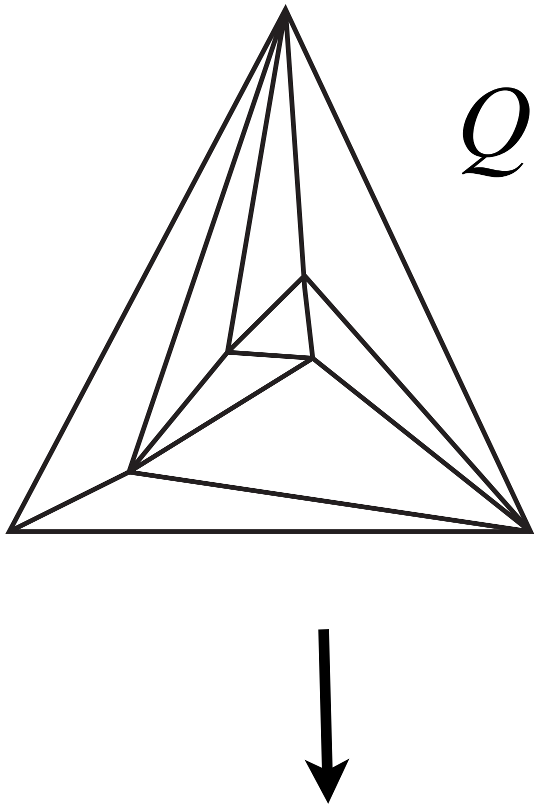
A morphing algorithm that computes a “snapshot” at any time t, 0 ≤ t ≤ 1.

[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]

P



Q



$M_P = [m(u,v)]_{n \times n}$

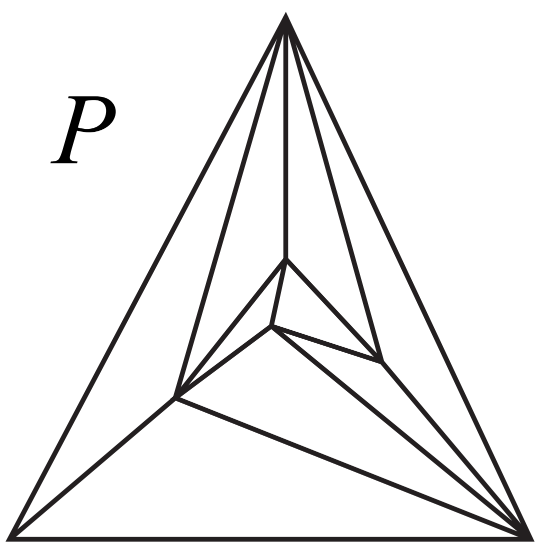
M_Q

Planar Graph Morphing: Algorithm

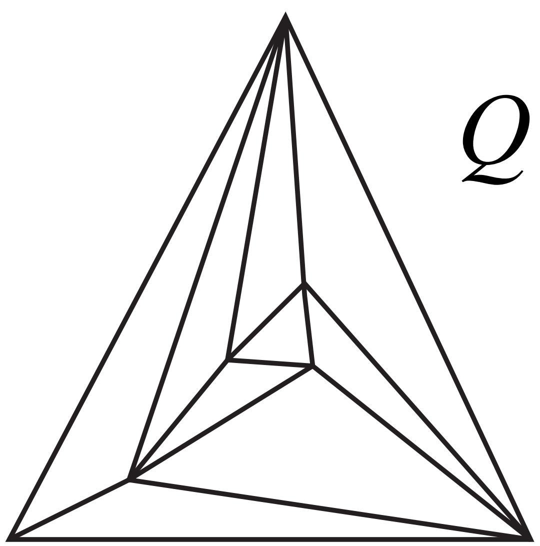
A morphing algorithm that computes a "snapshot" at any time t, 0 ≤ t ≤ 1.

[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]

P



Q



$M_P = [m(u,v)]_{n \times n}$

M_Q

$t = 0$

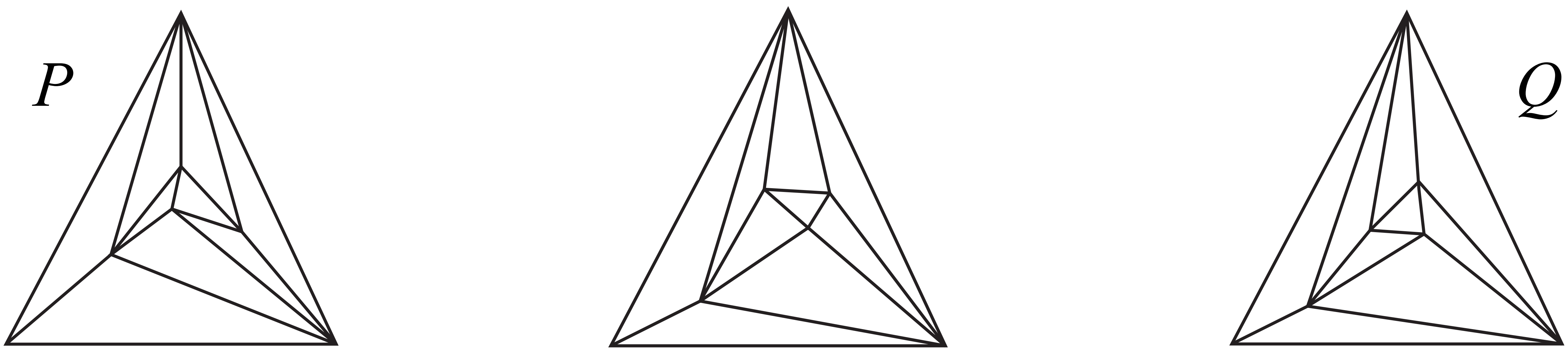
$$M(t) = (1-t)M_P + tM_Q$$

$t = 1$

Planar Graph Morphing: Algorithm

A morphing algorithm that computes a “snapshot” at any time t, 0 ≤ t ≤ 1.

[Floater and Gotsman 1999, Gotsman and Surazhsky 2001]



$$M_P = [m(u,v)]_{n \times n}$$

$$M_Q$$

$$t = 0$$

$$M(t) = (1-t)M_P + tM_Q$$

$$t = 1$$

Morphing Planar Straight-Line Graph Drawings

Planar Graph Drawing

- existence of straight-line drawing [Wagner, Koebe 1936, Fáry 1948, Stein 1951]
- an algorithm [Tutte 1963]
- polynomial size grid [de Fraysseix, Pach, Pollack; Schnyder 1990]

Planar Graph Morphing

- existence of morph preserving straight-line [Cairns 1944, Thomassen 1983]
- an algorithm [Floater and Gotsman 1999, Gotsman and Surazhsky 2001]
- polynomial size ??

Open Problem

Given two straight line planar drawings of a graph, find a polynomial size planar morph between them.

Requirements:

- straight line edges
- piece-wise linear

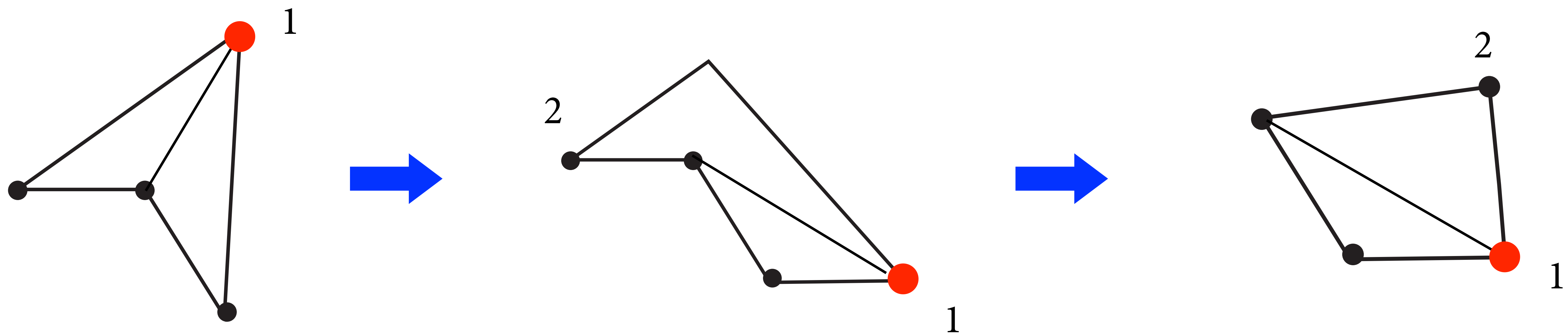
Outline

- transform planar graph drawing to specific target ("morphing")
    - straight line edges
    - edges are [orthogonal] poly-lines
    - morphing preserving lengths, directions, etc.

- transform planar graph drawing to attain convex faces
    - polygon, with increasing visibility

Morphing Planar ~~Straight-Line~~ Graph Drawings

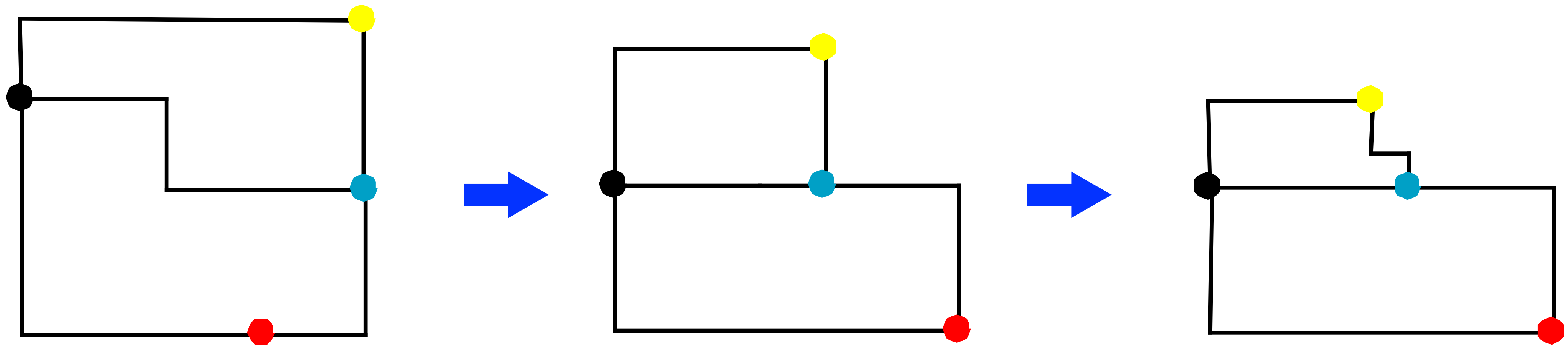
Morphing (with Bent Edges) [Lubiw, Petrick, 2011]



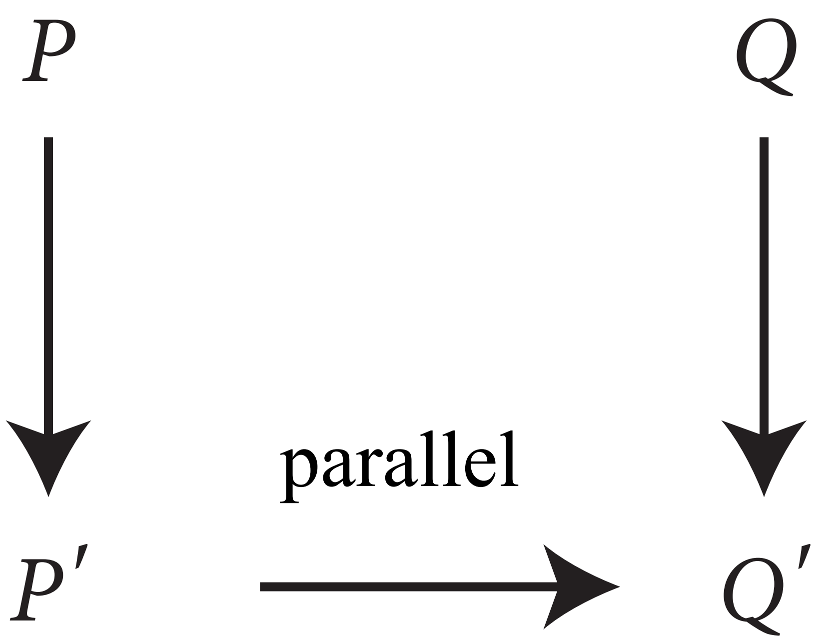
Theorem. There is a polynomial time algorithm to compute a planar morph between two planar straight-line drawings P and Q (of the same graph) that

- is composed of $O(n^6)$ linear morphs
- uses a polynomial size grid

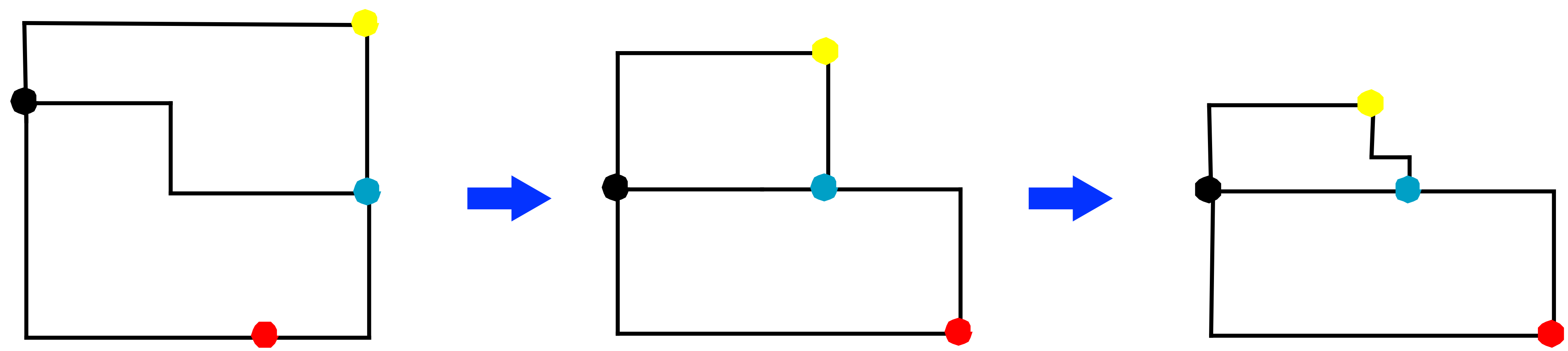
Morphing Orthogonal Graph Drawings [Biedl, Lubiw, Petrick, Spriggs]



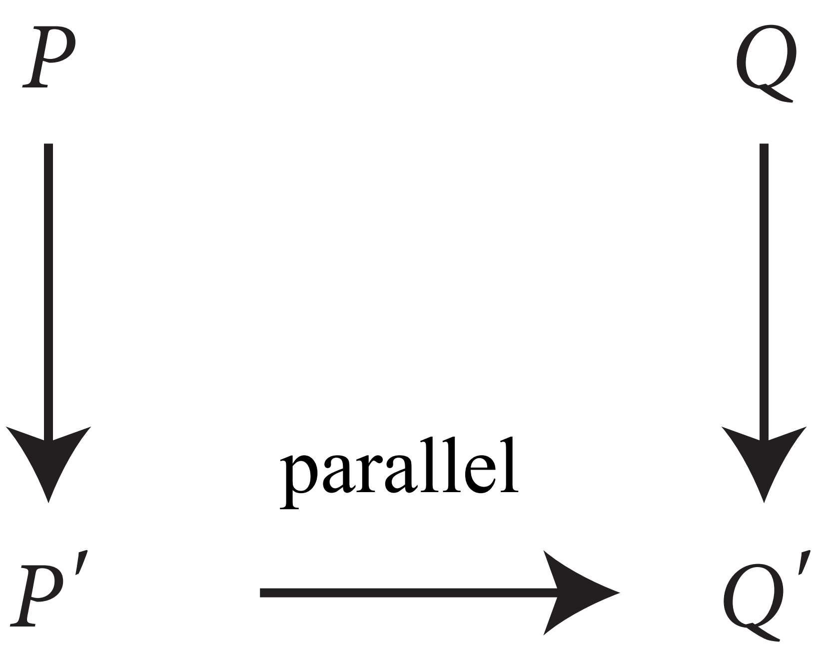
Main idea: reduce to the case of *parallel* orthogonal graph drawings



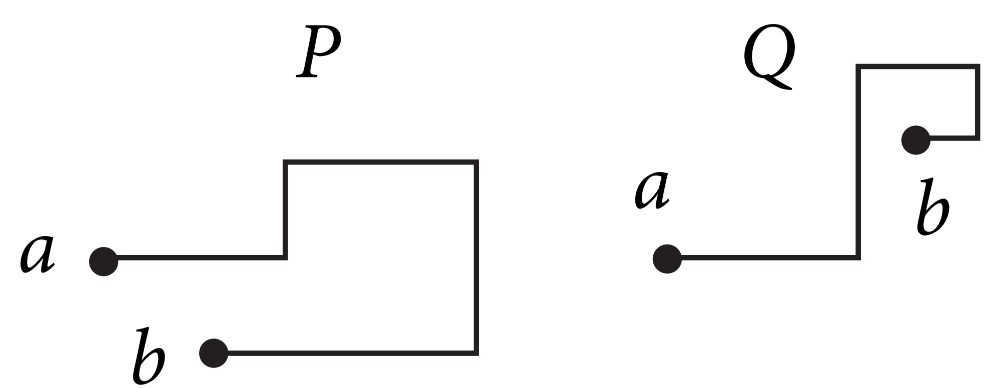
Morphing Orthogonal Graph Drawings [Biedl, Lubiw, Petrick, Spriggs]



Main idea: reduce to the case of *parallel* orthogonal graph drawings



(a,b) is *parallel* in P and Q:



direction sequence: E N E S W

Morphing Orthogonal Graph Drawings [Biedl, Lubiw, Petrick, Spriggs]

Theorem. There is a polynomial time algorithm to compute a morph between two orthogonal drawings of the same graph that

- preserves planar, othogonal
- is composed of $O(n^4)$ linear morphs
- $O(n) \times O(n)$ grid
- constant minimum feature size

Outline

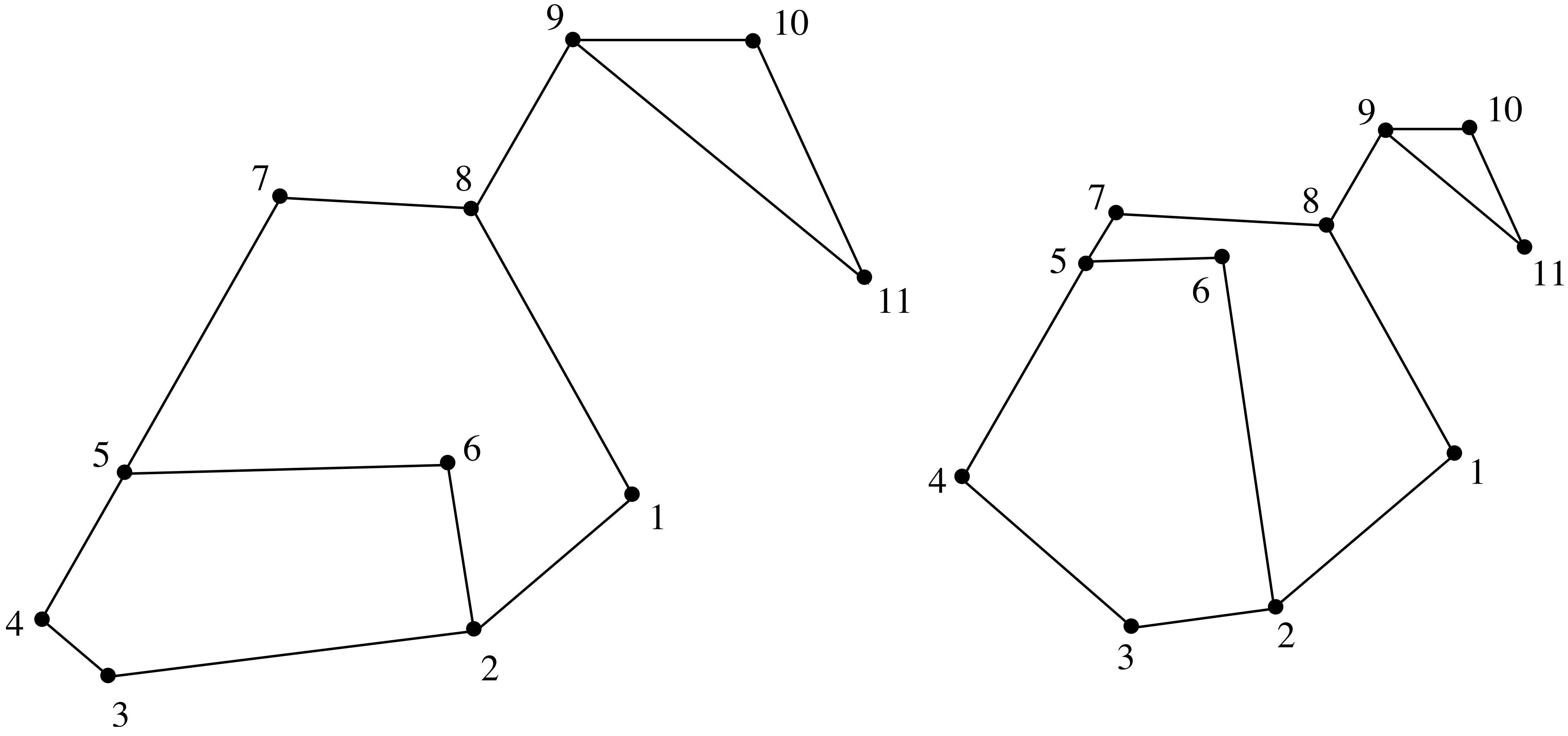
- transform planar graph drawing to specific target ("morphing")
  - straight line edges
  - edges are [orthogonal] poly-lines
  - ➡ morphing preserving lengths, directions, etc.
- transform planar graph drawing to attain convex faces
  - polygon, with increasing visibility

Morphing Preserving Other Properties

- planar (non-intersecting)
- directions ("parallel")
- edge lengths
- angles

} or change these monotonically

Morphing Preserving Directions ("Parallel")



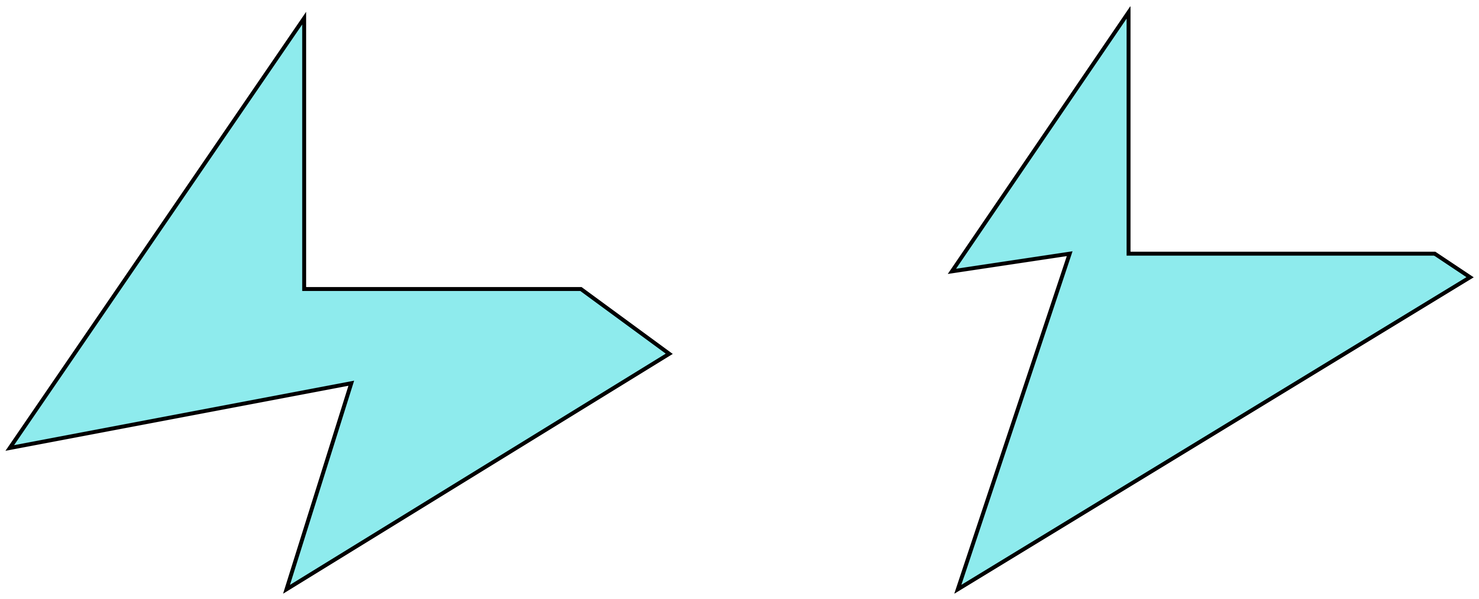
parallel planar graph drawings

Morphing Preserving Directions ("Parallel")

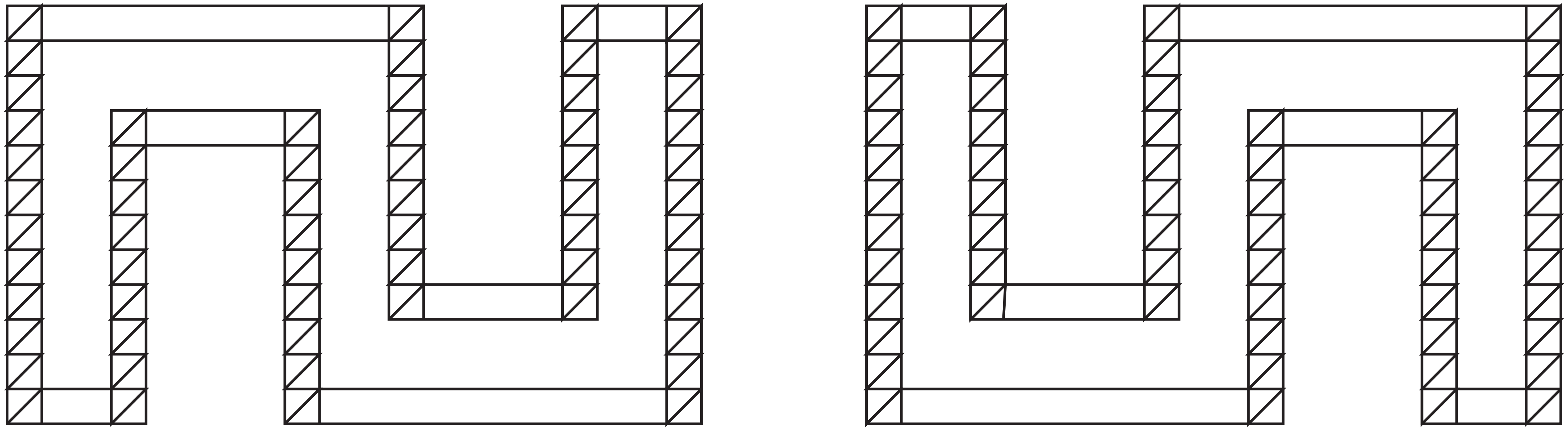
- parallel orthogonal graphs always have a parallel morph

- parallel cycles always have a parallel morph [Guibas, Hershberger, Suri, 2000]

$O(n \log n)$ steps but terrible edge lengths



Morphing Preserving Directions ("Parallel")

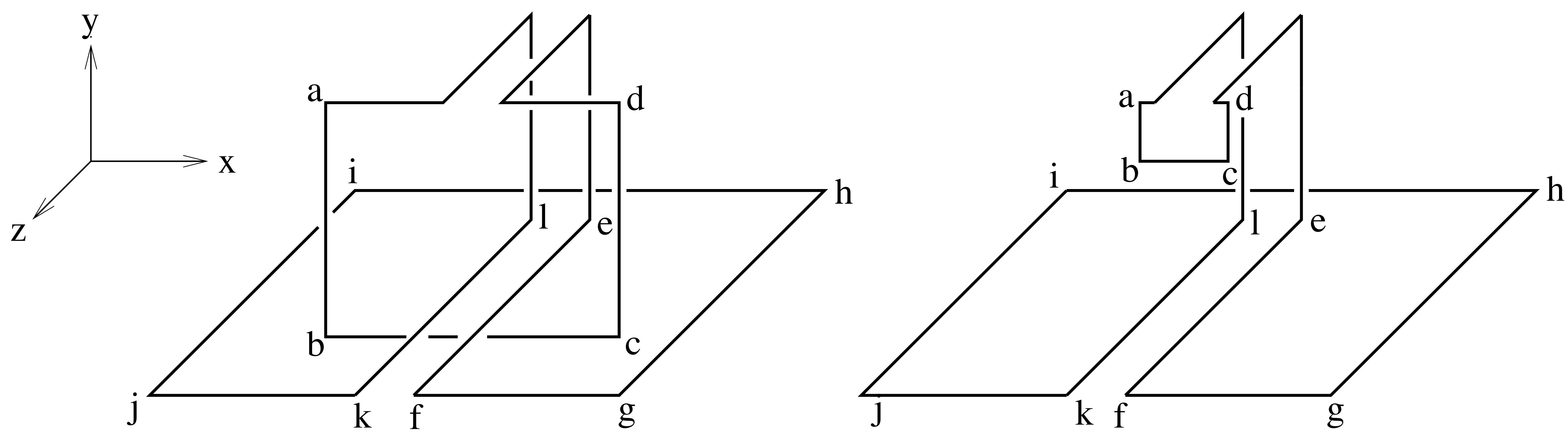


parallel planar graph drawings with no parallel morph.

Decision problem is NP-hard [Biedl, Lubiw, Spriggs].

Morphing Preserving Directions ("Parallel")

Orthogonal 3D Graph Drawings



Parallel orthogonal cycles with no parallel morph.

Decision problem PSPACE-hard for parallel orthogonal 3D graphs

[Biedl, Lubiw, Spriggs].

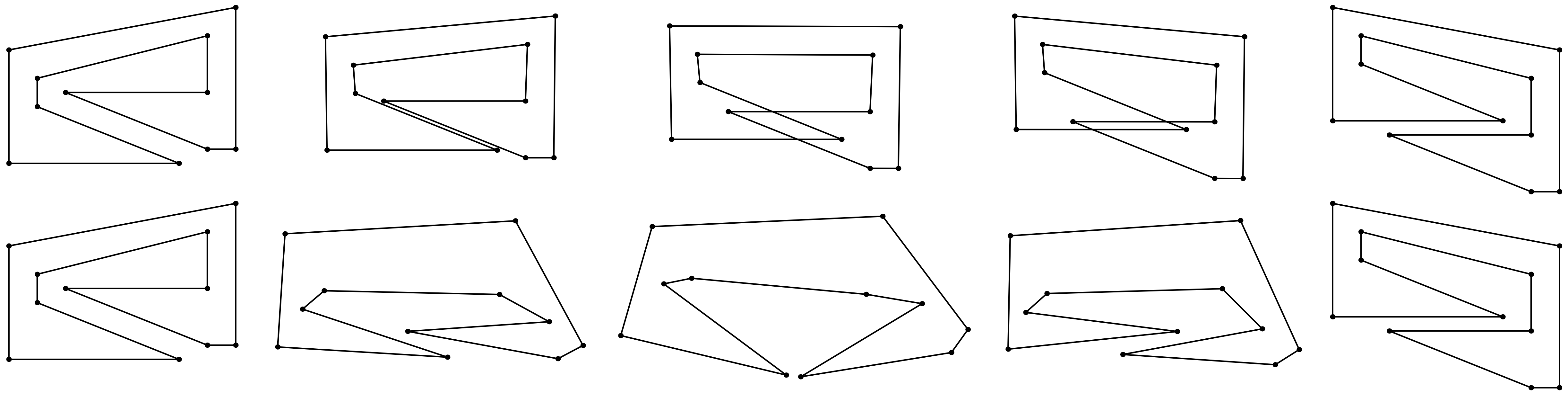
Open for cycles.

Morphing Preserving Edge Lengths

- non-intersecting morph between polygons, preserving edge lengths

[the Carpenter's Rule Theorem: Connelly, Demaine, Rote, 2003]

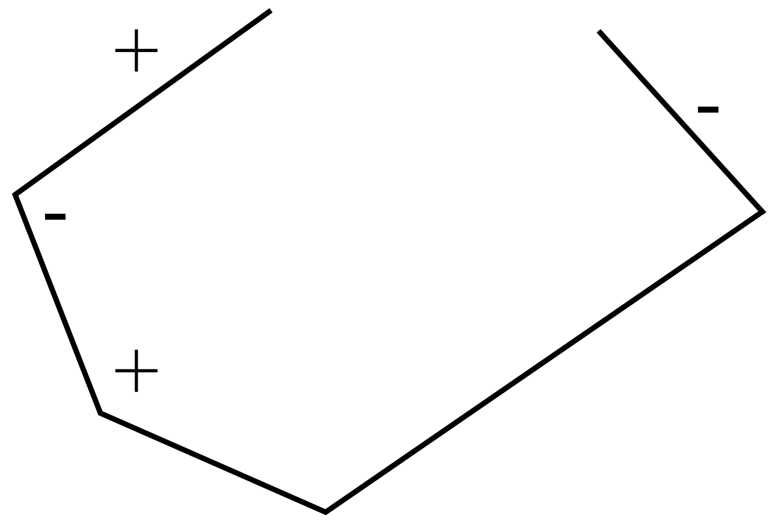
- non-intersecting morph between polygons, edge lengths change monotonically [Iben, O'Brien, Demaine, 2006]



Not possible for graphs. Not possible for parallel morphs of polygons.

Morphing Preserving Edge Lengths and Angles

Conjecture. Open convex chains can be morphed with edge lengths and angles changing monotonically.



Outline

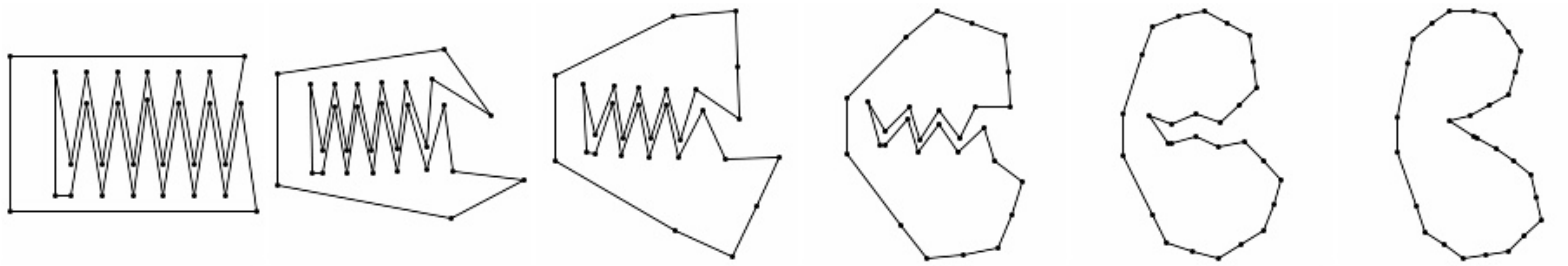
- transform planar graph drawing to specific target (“morphing”)
  - straight line edges
  - edges are [orthogonal] poly-lines
  - morphing preserving lengths, directions, etc.

➡
- transform planar graph drawing to attain convex faces
  - polygon, with increasing visibility

Convexifying Polygons

Convexifying is easy.

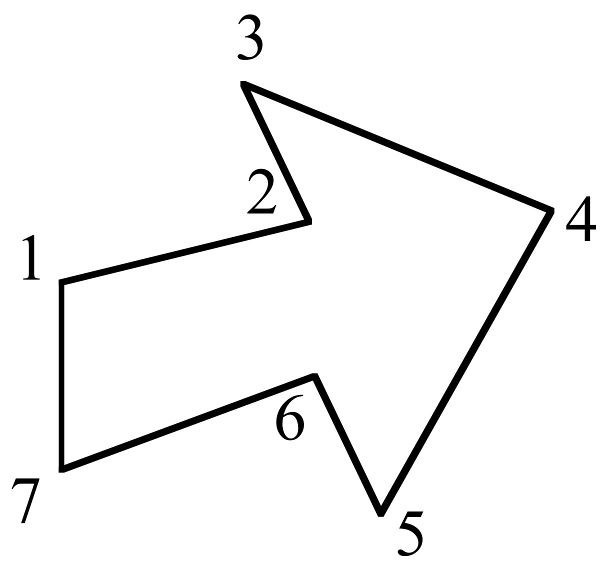
Convexifying preserving edge lengths is always possible:



the Carpenter's Rule Theorem: Connelly, Demaine, Rote, 2003

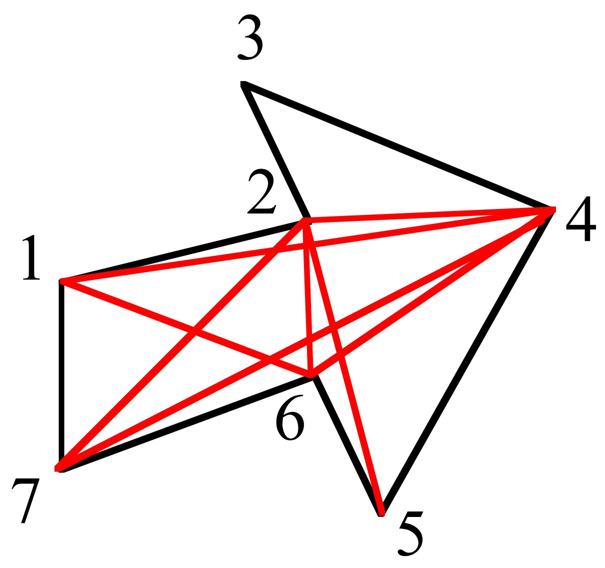
Convexifying without losing visibilities

Given a simple polygon, convexify without losing visibilities. [Devadoss, 2008]



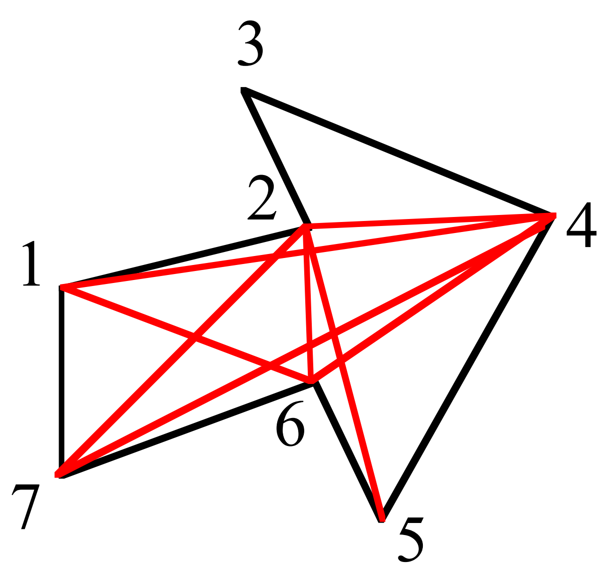
Convexifying without losing visibilities

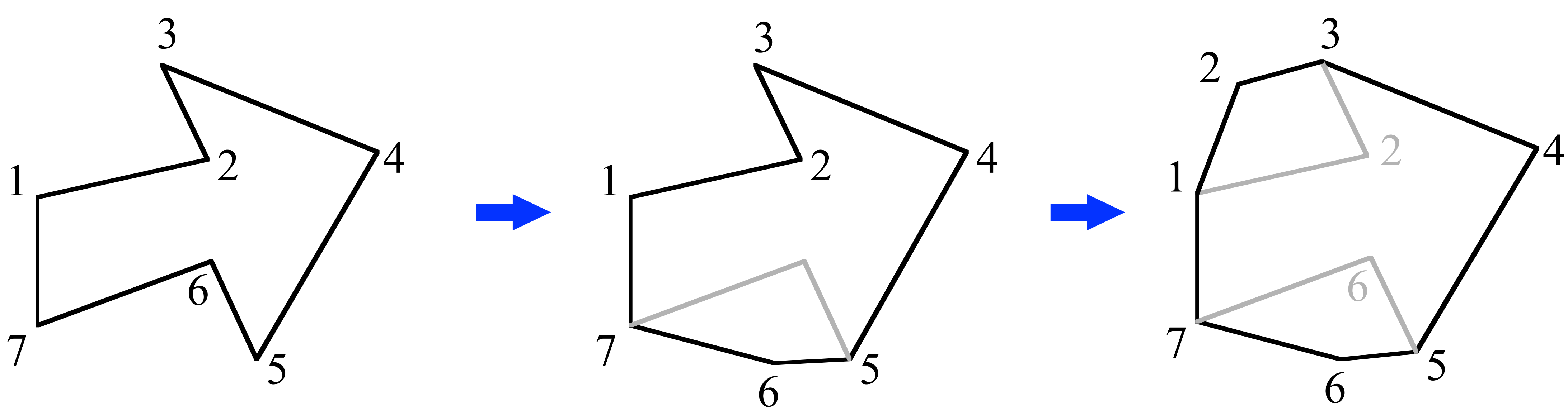
Given a simple polygon, convexify without losing visibilities. [Devadoss, 2008]



Convexifying without losing visibilities

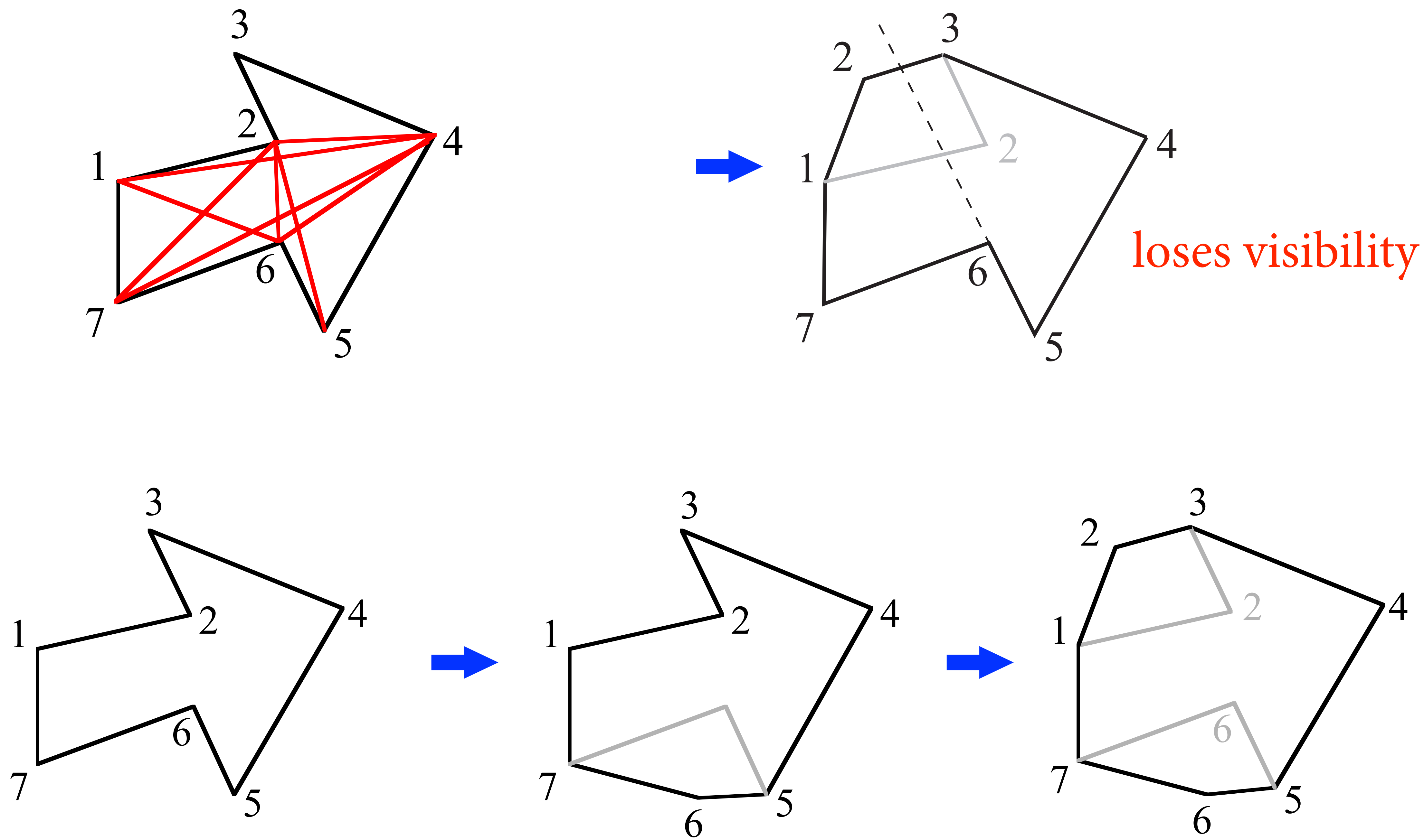
Given a simple polygon, convexify without losing visibilities. [Devadoss, 2008]





Convexifying without losing visibilities

Given a simple polygon, convexify without losing visibilities. [Devadoss, 2008]

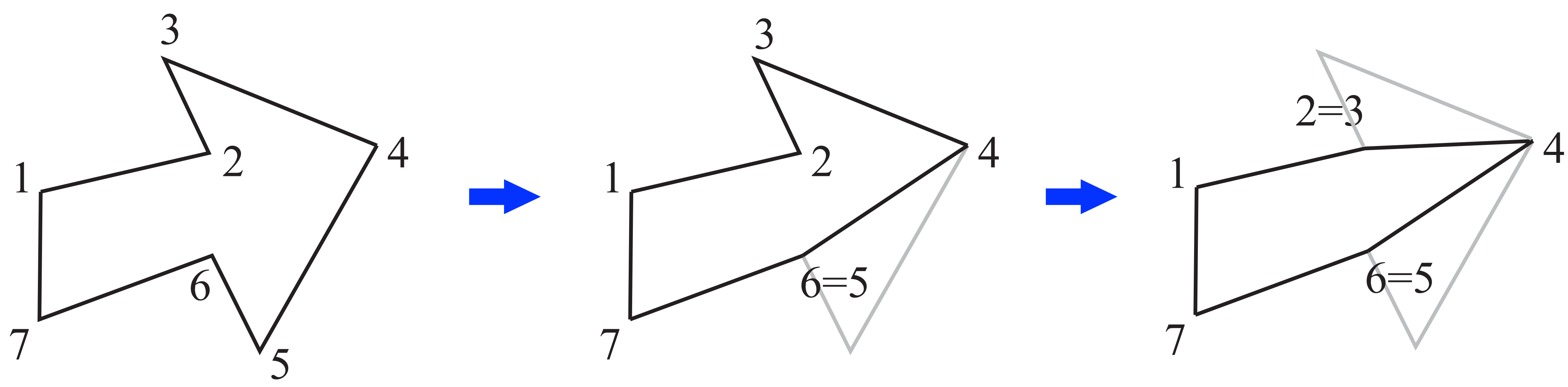


Convexifying without losing visibilities

[Aichholzer, Hurtado, Aloupis, Lubiw, Demaine, Demaine, Dujmović, Rote, Schulz, Souvaine, Winslow, 2011]

Theorem. Can convexify any polygon in *n* moves where

- every move increases visibility
- a move translates one vertex along a polygon edge to a neighbour



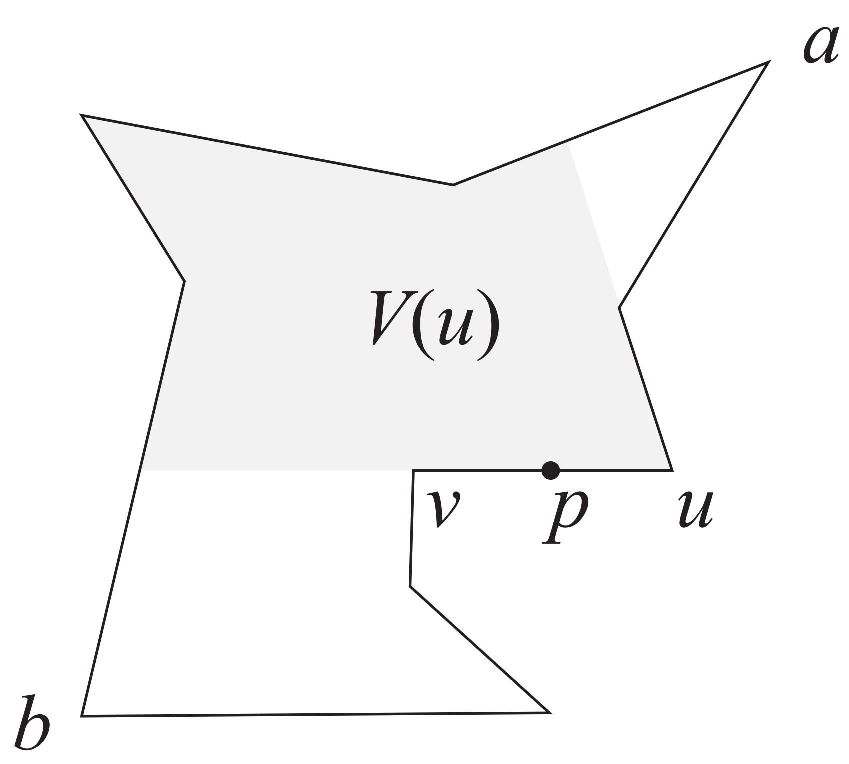
Note that a vertex of the current polygon represents a set of vertices of the original polygon.

How to Convexify in n moves

Lemma. Every non-convex polygon has a *visibility-increasing edge*:
an edge (u,v) such that for every point p along the edge (u,v),

$$V(u) \subseteq V(p) \subseteq V(v)$$

and there is a vertex in $V(v) - V(u)$.

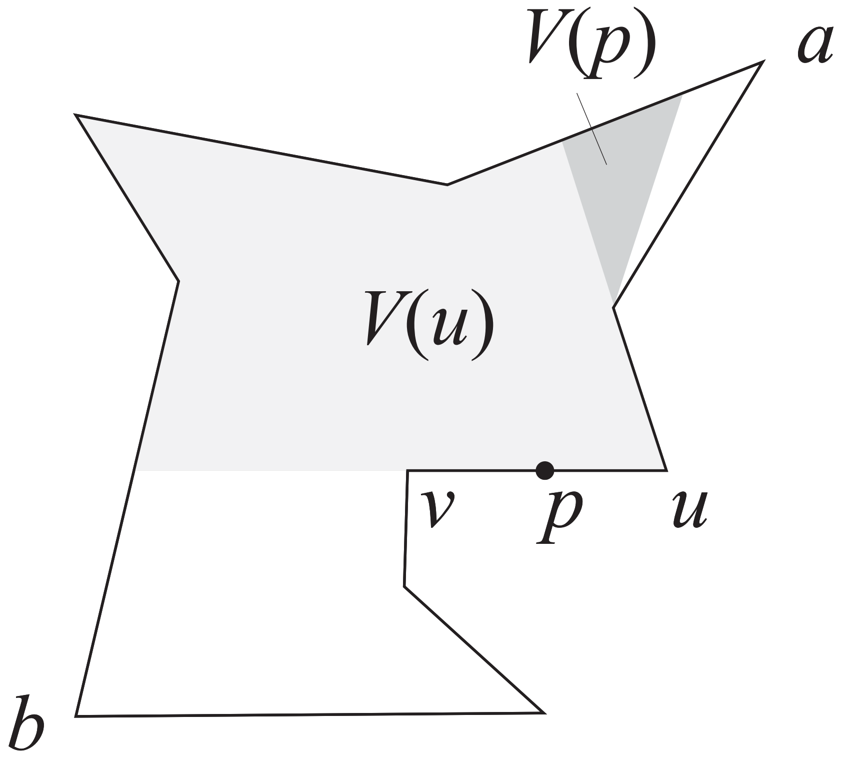


How to Convexify in n moves

Lemma. Every non-convex polygon has a *visibility-increasing edge*: an edge (u,v) such that for every point p along the edge (u,v),

$$V(u) \subseteq V(p) \subseteq V(v)$$

and there is a vertex in $V(v) - V(u)$.

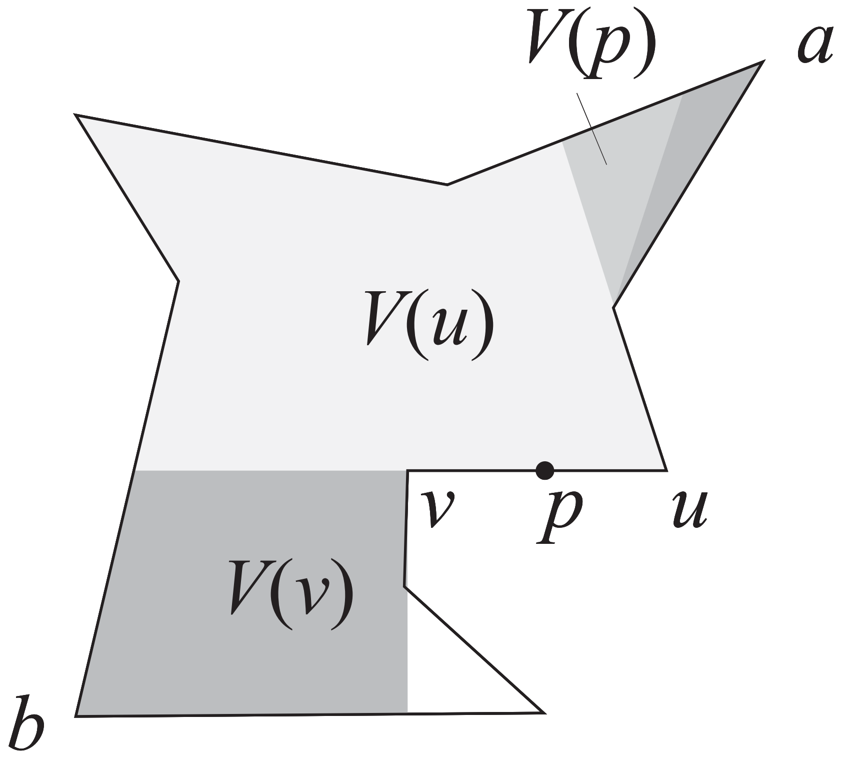


How to Convexify in n moves

Lemma. Every non-convex polygon has a *visibility-increasing edge*: an edge (u,v) such that for every point p along the edge (u,v),

$$V(u) \subseteq V(p) \subseteq V(v)$$

and there is a vertex in $V(v) - V(u)$.

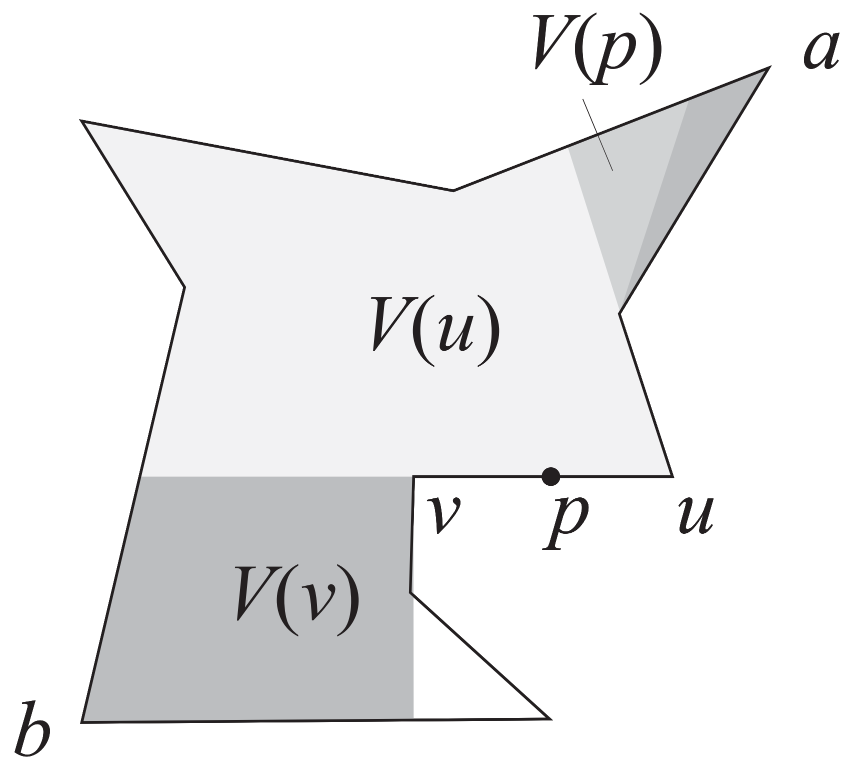


How to Convexify in n moves

Lemma. Every non-convex polygon has a *visibility-increasing edge*: an edge (u,v) such that for every point p along the edge (u,v),

$$V(u) \subseteq V(p) \subseteq V(v)$$

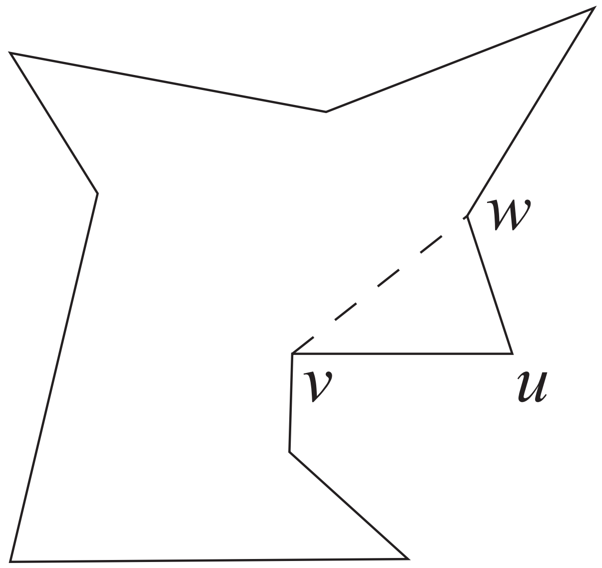
and there is a vertex in $V(v) - V(u)$.



Lemma ⇒ Theorem 1

Move u to v.

Note u convex, (w,v) a chord.



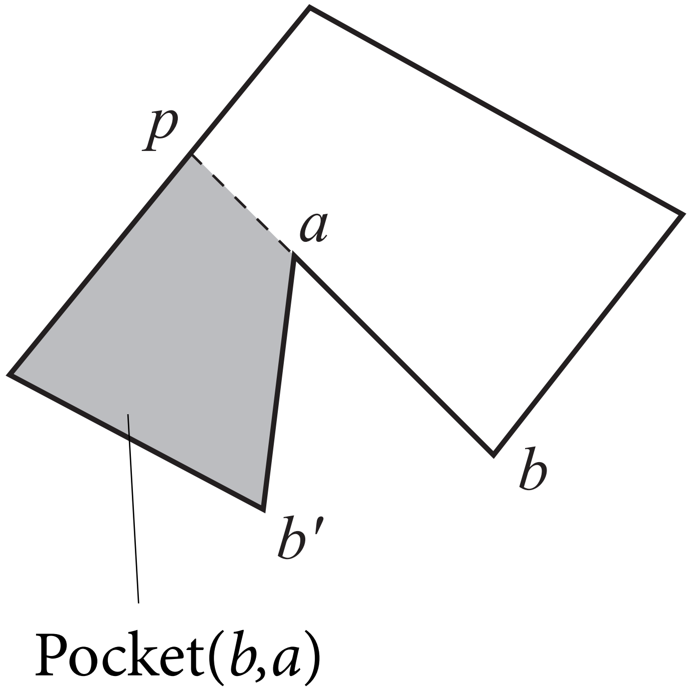
How to Convexify in n moves

Lemma. Every non-convex polygon has a *visibility-increasing edge*: an edge (u,v) such that for every point p along the edge (u,v),

$$V(u) \subseteq V(p) \subseteq V(v)$$

and there is a vertex in $V(v) - V(u)$.

Proof. For any edge (b,a) with a reflex, there is a visibility-increasing edge outside Pocket(b,a). By induction on # vertices outside Pocket(b,a).



How to Convexify in n moves

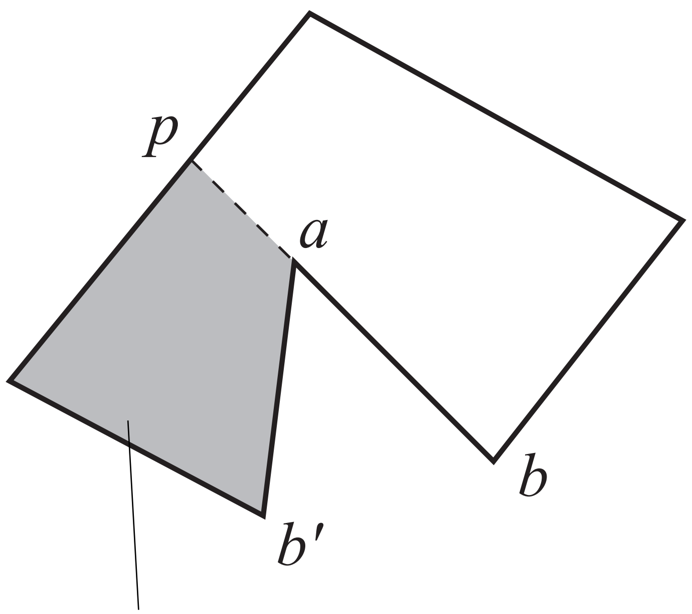
Lemma. Every non-convex polygon has a *visibility-increasing edge*:

an edge (u,v) such that for every point p along the edge (u,v),

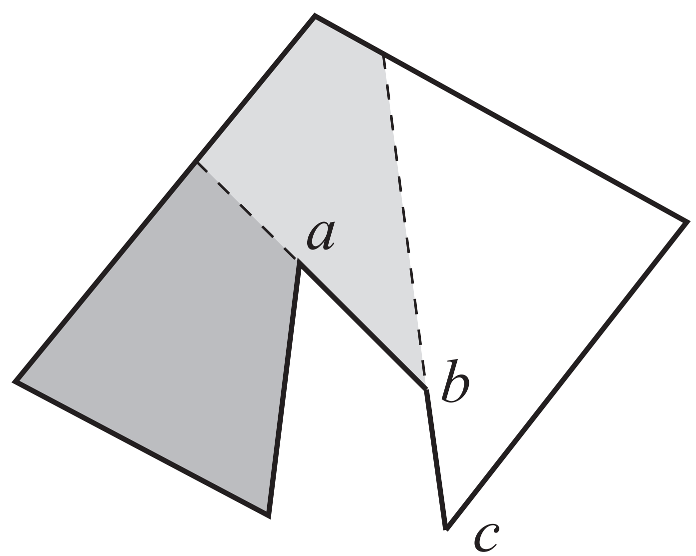
$$V(u) \subseteq V(p) \subseteq V(v)$$

and there is a vertex in $V(v)$ - $V(u)$.

Proof. For any edge (b,a) with a reflex, there is a visibility-increasing edge outside Pocket(b,a). By induction on # vertices outside Pocket(b,a).



Pocket(b,a)



induction on (c,b)

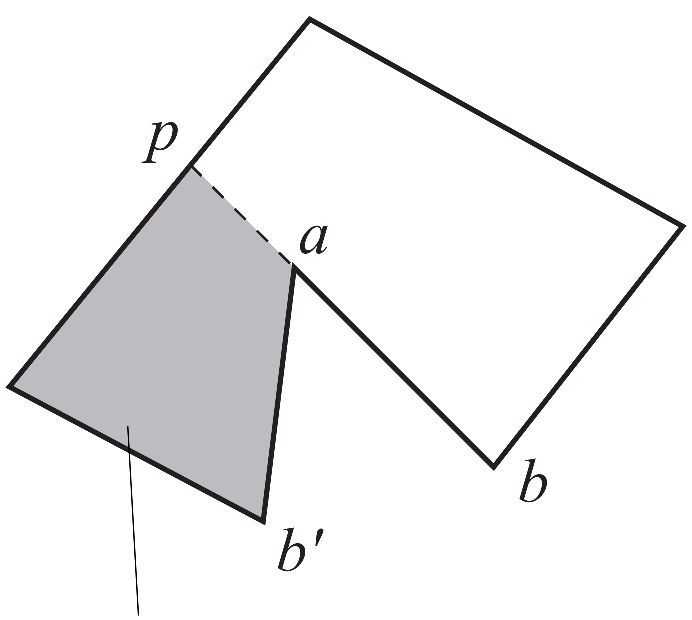
How to Convexify in n moves

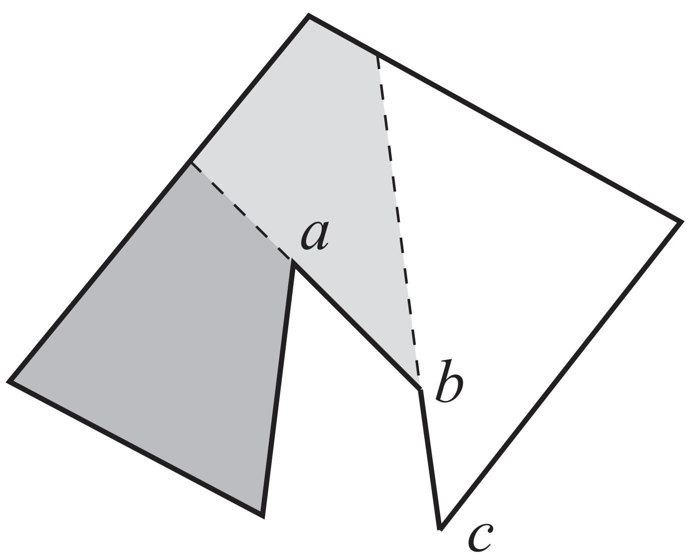
Lemma. Every non-convex polygon has a *visibility-increasing edge*:
an edge (u,v) such that for every point p along the edge (u,v),

$$V(u) \subseteq V(p) \subseteq V(v)$$

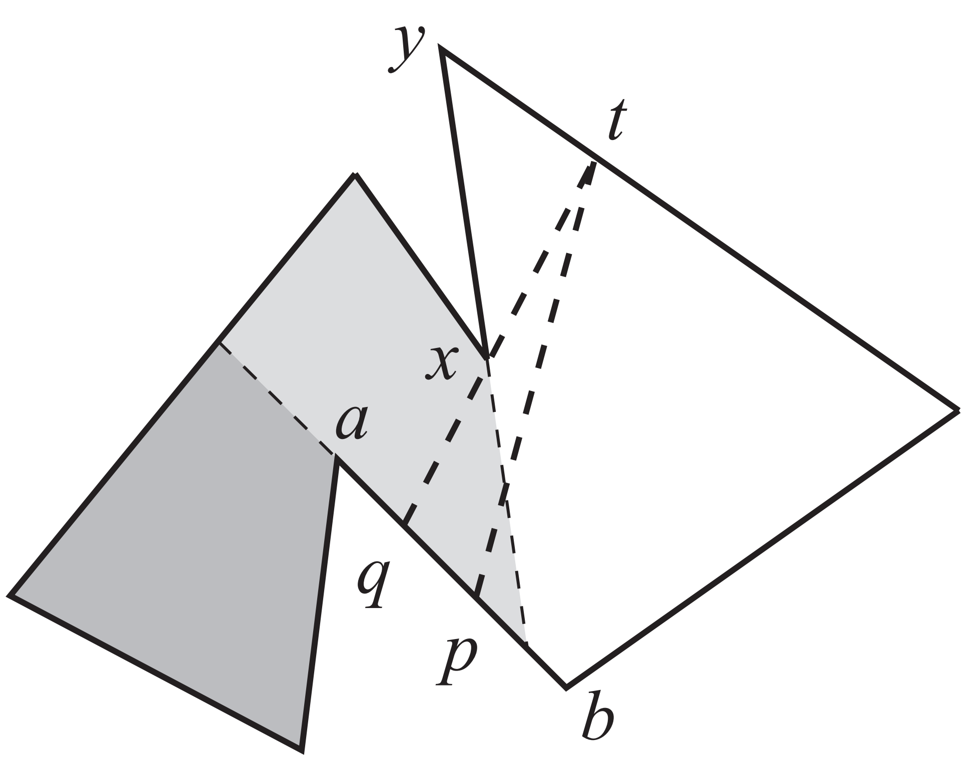
and there is a vertex in $V(v) - V(u)$.

Proof. For any edge (b,a) with a reflex, there is a visibility-increasing edge outside Pocket(b,a). By induction on # vertices outside Pocket(b,a).





induction on (c,b)

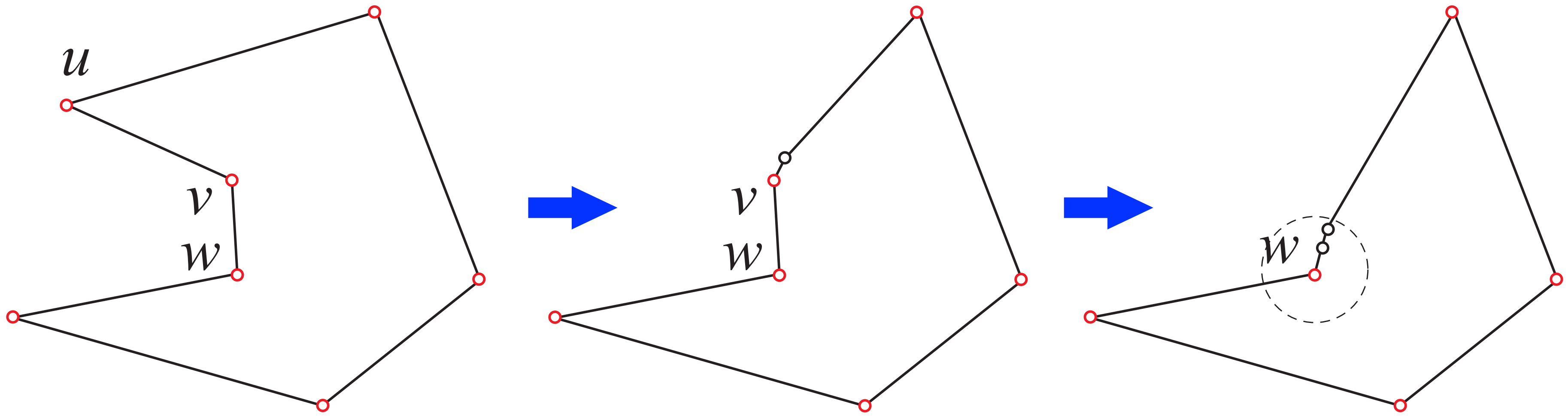


induction on (y,x)

How to convexify without coincident vertices

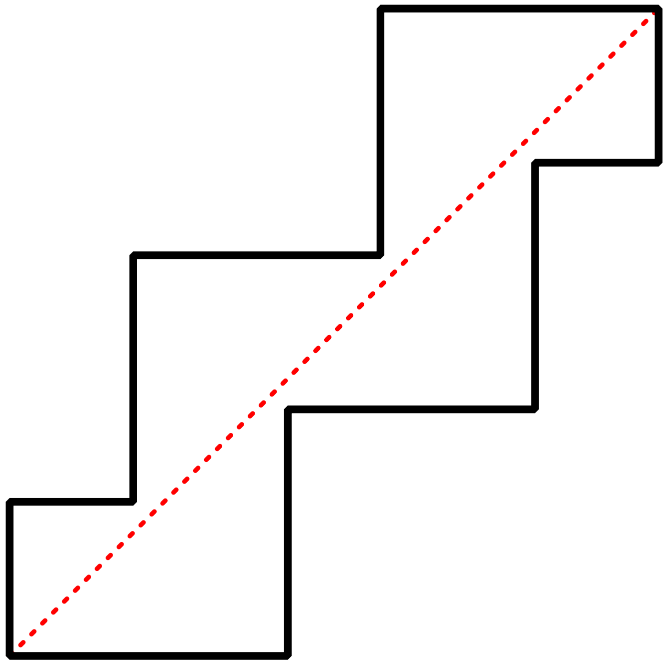
Theorem. Can convexify any polygon in $O(n^2)$ moves where

- no move decreases visibility
- a move translates one vertex in a straight line
- vertices are never coincident



Open Questions

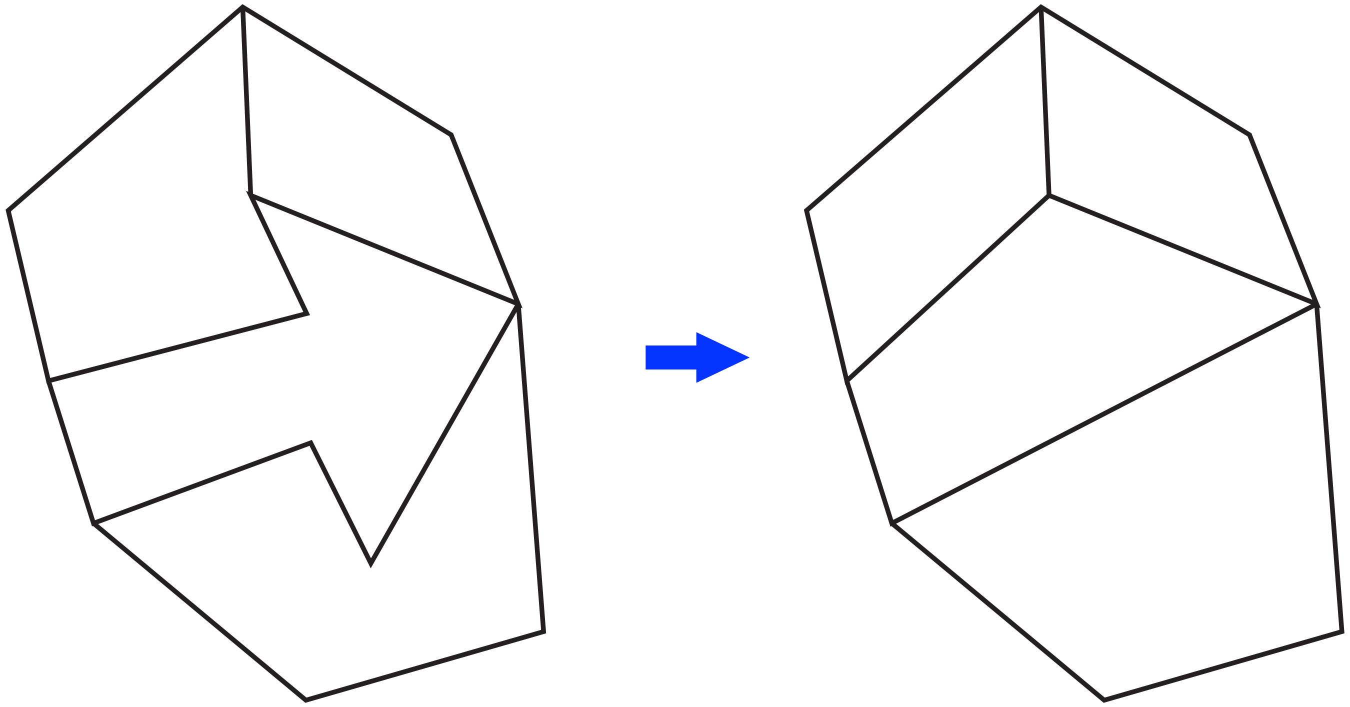
Convexify an orthogonal polygon without losing visibility, maintaining orthogonality.



in this example, no single edge can move

Open Questions

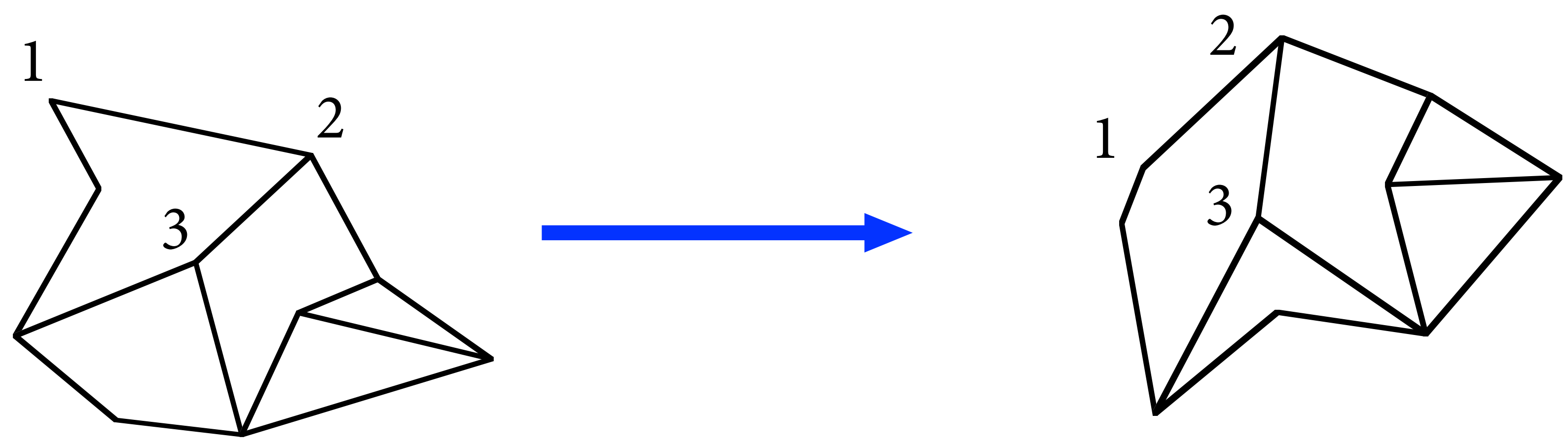
Transform a planar straight-line graph drawing to a convex one, without losing visibilities.



Ignoring visibility constraints, this can be done [Thomassen 1983, Cairns 1944].
But can it be done efficiently?

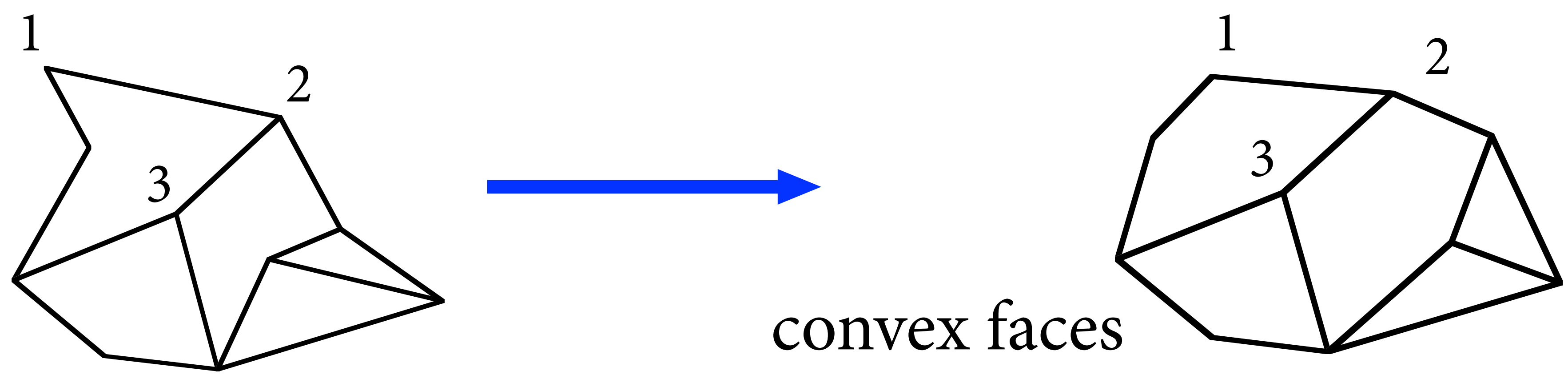
Open Questions

Given two straight line planar drawings of a graph, find a polynomial size planar morph between them.



or at least:

Given a straight line planar drawing of a graph, find a polynomial size planar morph to a drawing with convex internal faces.



The End