Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Lehrstuhl für
INFORMATIK I
Algorithmen & Komplexität

Institut für Informatik

Seminar:
Themen der Algorithmik

Sommersemester 2025

Einführungsveranstaltung am 29. April 2025

Lehrstuhl für Informatik I

Alexander Wolff, Boris Klemz, Diana Sieper, Tim Hegemann, Samuel Wolf

Ziele und Inhalte

In diesem Seminar geht es teils um **aktuelle Forschungsthemen** und **neue Trends**, teils **klassische Resultate** aus dem Gebiet **Algorithmik**.

Ziele und Inhalte

In diesem Seminar geht es teils um **aktuelle Forschungsthemen** und **neue Trends**, teils **klassische Resultate** aus dem Gebiet **Algorithmik**.

JedeR TeilnehmerIn arbeitet sich in ein abgegrenztes Thema ein. Dieses ist didaktisch aufzubereiten und den anderen KursteilnehmerInnen in einem **Vortrag** zu vermitteln, sowie in einer **schriftliche Ausarbeitung** darzustellen.

Ablauf des Seminars

- Di, 29.04.2025: **Einführung**

Ablauf des Seminars

- Di, 29.04.2025: **Einführung**
- Di, 06.05.2025: **Kurzvorträge** zu jedem Thema
  (etwa 5 Min., ca. 3 Folien)

Ablauf des Seminars

- Di, 29.04.2025: **Einführung**

- Di, 06.05.2025: **Kurzvorträge** zu jedem Thema
  (etwa 5 Min., ca. 3 Folien)

Inhalte:

- Ausblick auf den eigentlichen Vortrag geben
- Problemstellung nennen & motivieren
- Wichtigste Resultate nennen & einordnen

Ablauf des Seminars

- Di, 29.04.2025: **Einführung**

- Di, 06.05.2025: **Kurzvorträge** zu jedem Thema
  (etwa 5 Min., ca. 3 Folien)

Inhalte:

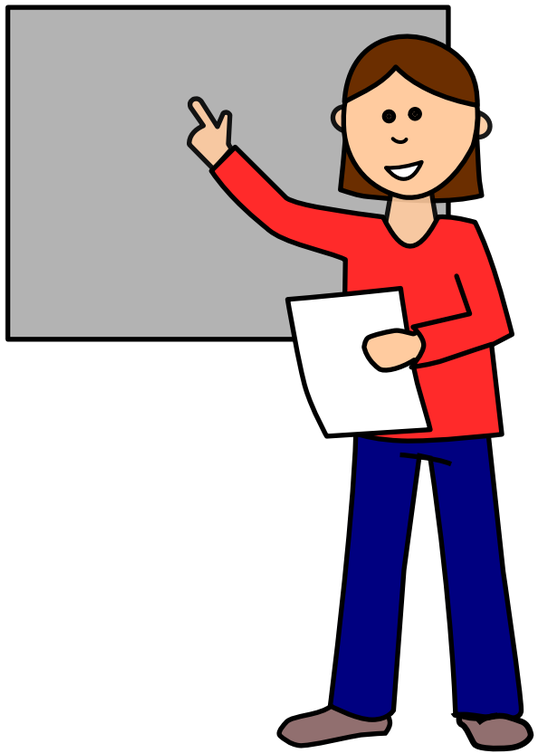
- Ausblick auf den eigentlichen Vortrag geben
- Problemstellung nennen & motivieren
- Wichtigste Resultate nennen & einordnen

Ziele:

- Zeitnah einarbeiten
- Themenauswahl prüfen
- Vortragen üben
- Feedback bekommen ohne Bewertung

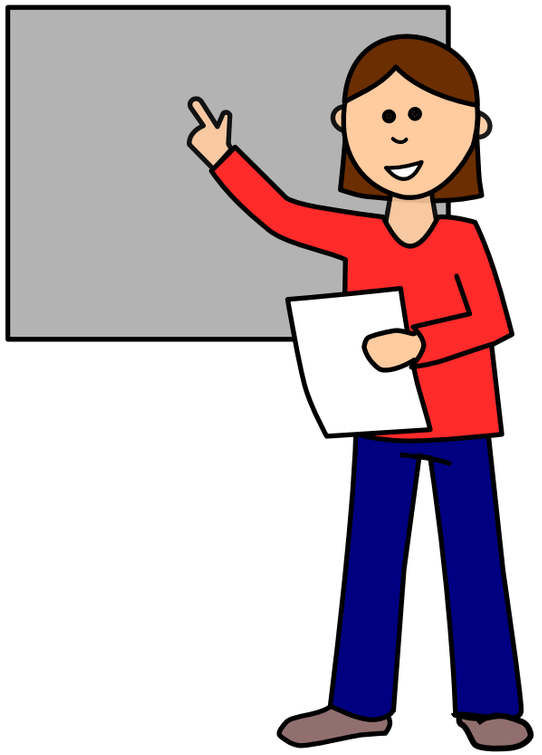
Ablauf des Seminars

- Di, 29.04.2025: **Einführung**
- Di, 06.05.2025: **Kurzvorträge** zu jedem Thema (etwa 5 Min., ca. 3 Folien)
- ab Di, 20.05.2025: **Hauptvorträge** (i.d.R. einer pro Woche)



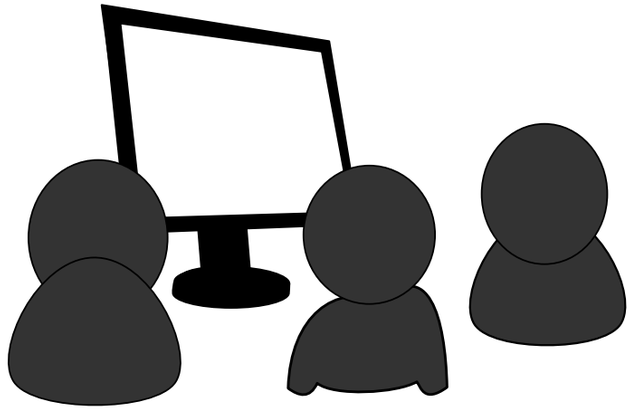
Ablauf des Seminars

- Di, 29.04.2025: **Einführung**
- Di, 06.05.2025: **Kurzvorträge** zu jedem Thema (etwa 5 Min., ca. 3 Folien)
- ab Di, 20.05.2025: **Hauptvorträge** (i.d.R. einer pro Woche)
- Mo, 04.08.2025: **Ausarbeitungen** abgeben



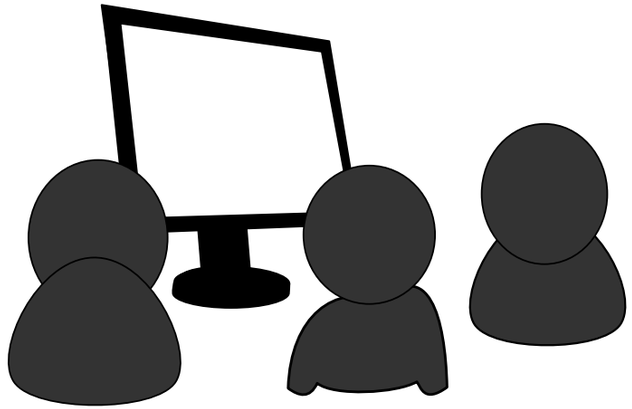
Vorträge

- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)



Vorträge

- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)

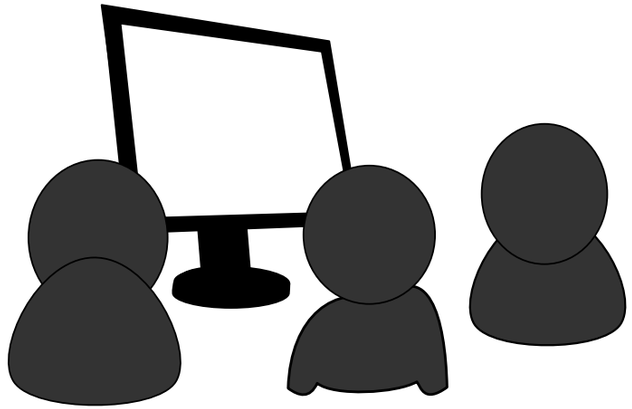


Das reicht i.d.R. nicht um alles im Detail zu besprechen!

$\rightarrow$ wesentliche Teile identifizieren und ausführlich behandeln, unwesentliche Teile skizzieren

Vorträge

- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)



Das reicht i.d.R. nicht um alles im Detail zu besprechen!

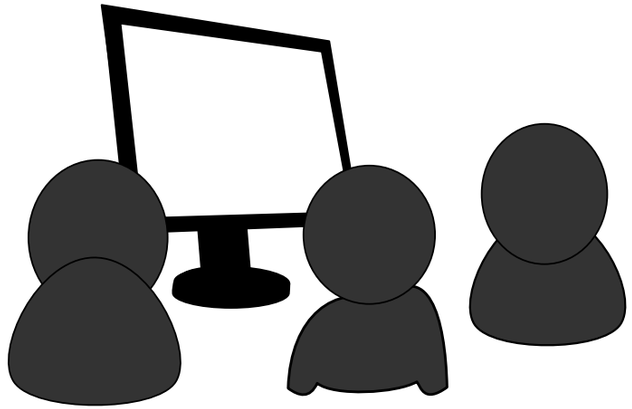
→ wesentliche Teile identifizieren und ausführlich behandeln, unwesentliche Teile skizzieren

Ausnahme: Einige Themen sind weniger umfangreich

→ verbleibende Zeit durch Inhalte angrenzender Literatur füllen (eigene Literaturrecherche!)

Vorträge

- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)



Das reicht i.d.R. nicht um alles im Detail zu besprechen!

→ wesentliche Teile identifizieren und ausführlich behandeln, unwesentliche Teile skizzieren

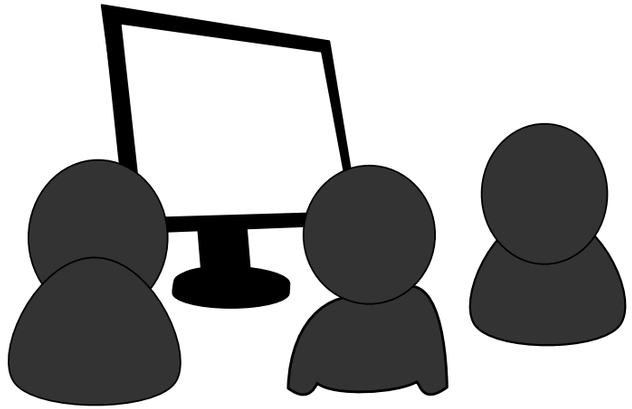
Ausnahme: Einige Themen sind weniger umfangreich

→ verbleibende Zeit durch Inhalte angrenzender Literatur füllen (eigene Literaturrecherche!)

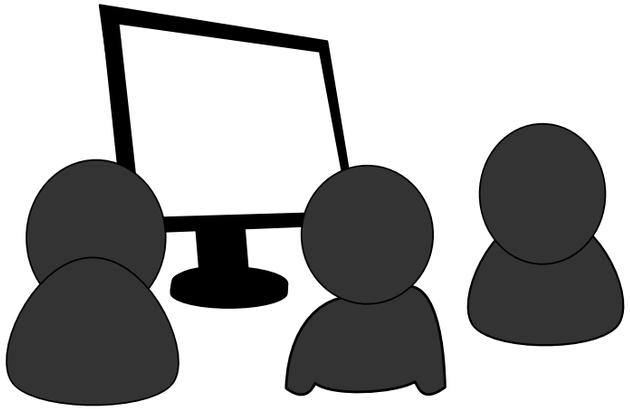
In jedem Fall sollen die 45 / 60 Minuten stimmig ausgefüllt werden.

Vorträge

- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)



Vorträge

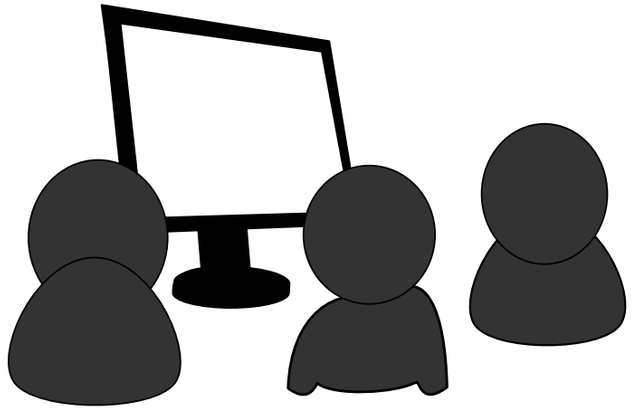


- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)

- anschließend/währenddessen **Diskussion / Interaktion**
  (Übungsaufgaben, interaktive Beispiele, Besprechung offener Probleme, etc.) (geht nicht in die Zeit ein)

  Ideen aus der Diskussion in die Ausarbeitung mitaufnehmen!

Vorträge



- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)

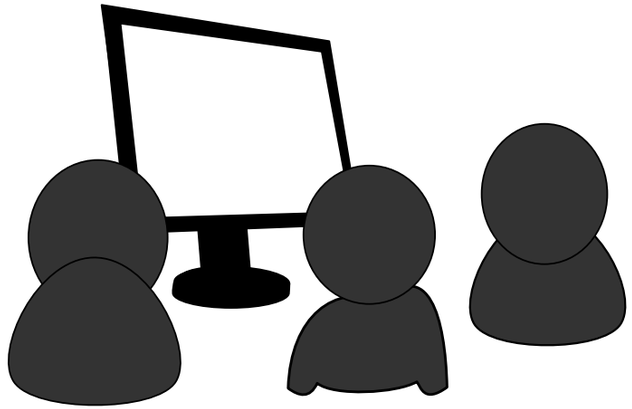
- anschließend/währenddessen **Diskussion / Interaktion**
  (Übungsaufgaben, interaktive Beispiele, Besprechung offener Probleme, etc.) (geht nicht in die Zeit ein)

  Ideen aus der Diskussion in die Ausarbeitung mitaufnehmen!

Vorbesprechungen (verpflichtend):

- **Drei** Wochen vor dem eigenen Vortrag:
  Besprechung der **Inhaltsübersicht** mit eurer BetreuerIn

Vorträge



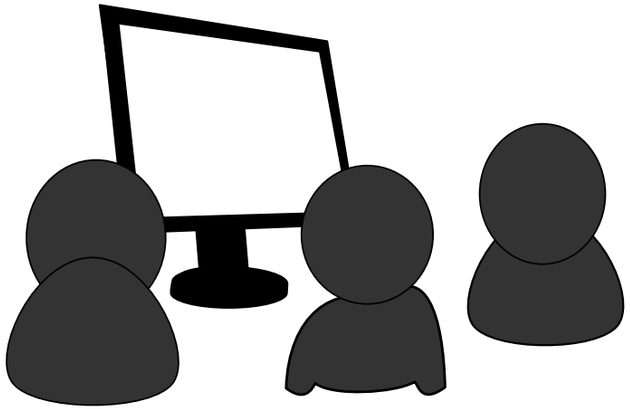
- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)
- anschließend/währenddessen **Diskussion / Interaktion**
  (Übungsaufgaben, interaktive Beispiele, Besprechung offener Probleme, etc.) (geht nicht in die Zeit ein)

  Ideen aus der Diskussion in die Ausarbeitung mitaufnehmen!

Vorbesprechungen (verpflichtend):

- **Drei** Wochen vor dem eigenen Vortrag:
  Besprechung der **Inhaltsübersicht** mit eurer BetreuerIn
- **Zwei** Wochen vor dem Vortrag:
  Besprechung eurer **Folien** mit eurer BetreuerIn

Vorträge



- etwa 45 Minuten **Vortrag**
  (zu zweit etwa 60 Minuten)

- anschließend/währenddessen **Diskussion / Interaktion**
  (Übungsaufgaben, interaktive Beispiele, Besprechung offener Probleme, etc.) (geht nicht in die Zeit ein)

  Ideen aus der Diskussion in die Ausarbeitung mitaufnehmen!

Diese Termine sind **strikt** (außer für den 1. Vortrag)!

Vorbesprechungen (verpflichtend):

- **Drei** Wochen vor dem eigenen Vortrag:
  Besprechung der **Inhaltsübersicht** mit eurer BetreuerIn
- **Zwei** Wochen vor dem Vortrag:
  Besprechung eurer **Folien** mit eurer BetreuerIn

Ausarbeitung

- alleine 7–9, zu zweit 11–13 Seiten;

Ausarbeitung

- alleine 7–9, zu zweit 11–13 Seiten;

Wie schon beim Vortrag gilt auch hier:

Das reicht i.d.R. nicht um alles im Detail zu beschreiben!

$\rightarrow$ wesentliche Teile identifizieren und ausführlich behandeln, unwesentliche Teile skizzieren

Ausarbeitung

- alleine 7–9, zu zweit 11–13 Seiten;

Wie schon beim Vortrag gilt auch hier:

Das reicht i.d.R. nicht um alles im Detail zu beschreiben!

→ wesentliche Teile identifizieren und ausführlich behandeln, unwesentliche Teile skizzieren

Ausnahme: Einige Themen sind weniger umfangreich.

→ durch geeignete eigene Inhalte erweitern

(siehe nächste Folie)

Ausarbeitung

- alleine 7–9, zu zweit 11–13 Seiten;

Wie schon beim Vortrag gilt auch hier:

Das reicht i.d.R. nicht um alles im Detail zu beschreiben!

→ wesentliche Teile identifizieren und ausführlich behandeln, unwesentliche Teile skizzieren

Ausnahme: Einige Themen sind weniger umfangreich.

→ durch geeignete eigene Inhalte erweitern

(siehe nächste Folie)

In jedem Fall sollen die 7–9 / 11–13 Seiten stimmig ausgefüllt werden.

Ausarbeitung

- alleine 7–9, zu zweit 11–13 Seiten;
  Abbildungen sind hilfreich!

Ausarbeitung

- alleine 7–9, zu zweit 11–13 Seiten;
  Abbildungen sind hilfreich! (und gehen nicht in das Seitenlimit ein)

Ausarbeitung

Bitte Vektorgrafiken, keine Bitmaps!

- alleine 7–9, zu zweit 11–13 Seiten;
  Abbildungen sind hilfreich! (und gehen nicht in das Seitenlimit ein)

Ausarbeitung

Bitte Vektorgrafiken, keine Bitmaps!

- alleine 7–9, zu zweit 11–13 Seiten;
  Abbildungen sind hilfreich! (und gehen nicht in das Seitenlimit ein)
- **keine reine Zusammenfassung** des Artikels; wir erwarten einen **eigenen Beitrag**. Z.B. manche Resultate weglassen, andere Beweise ausführlicher, offene Probleme diskutieren, eigene Literaturrecherche & Material aus angrenzender Literatur, Verbindungen zu anderen Vortragsthemen etc.

Ausarbeitung

Bitte Vektorgrafiken, keine Bitmaps!

- alleine 7–9, zu zweit 11–13 Seiten;
  Abbildungen sind hilfreich! (und gehen nicht in das Seitenlimit ein)
- **keine reine Zusammenfassung** des Artikels; wir erwarten einen **eigenen Beitrag**. Z.B. manche Resultate weglassen, andere Beweise ausführlicher, offene Probleme diskutieren, eigene Literaturrecherche & Material aus angrenzender Literatur, Verbindungen zu anderen Vortragsthemen etc.
- LaTeX-Vorlage auf der WueCampus Seite!

Ausarbeitung

Bitte Vektorgrafiken, keine Bitmaps!

- alleine 7–9, zu zweit 11–13 Seiten;
  Abbildungen sind hilfreich! (und gehen nicht in das Seitenlimit ein)
- **keine reine Zusammenfassung** des Artikels; wir erwarten einen **eigenen Beitrag**. Z.B. manche Resultate weglassen, andere Beweise ausführlicher, offene Probleme diskutieren, eigene Literaturrecherche & Material aus angrenzender Literatur, Verbindungen zu anderen Vortragsthemen etc.
- LaTeX-Vorlage auf der WueCampus Seite!
- **Vorabversion** der Ausarbeitung bis spätestens 2 Wochen nach dem eigenen Vortrag abgeben, um Feedback zu erhalten (freiwillig).

Bestehen & Bewertung

Voraussetzungen für das Bestehen des Seminars

- Halten einer Präsentation zum gewählten Thema
- Anfertigen einer Ausarbeitung
- Anwesenheit bei den anderen Vorträgen
- Einmaliges Fehlen ist erlaubt
- Teilnahme an den Diskussionen

Bestehen & Bewertung

Voraussetzungen für das Bestehen des Seminars

- Halten einer Präsentation zum gewählten Thema
- Anfertigen einer Ausarbeitung
- Anwesenheit bei den anderen Vorträgen
- Einmaliges Fehlen ist erlaubt
- Teilnahme an den Diskussionen

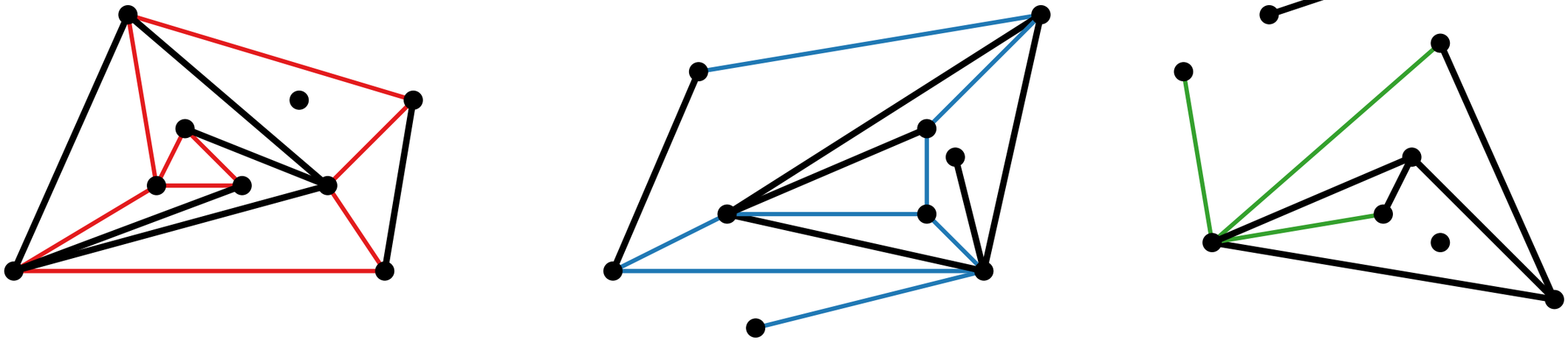
Bewertung

- Vortrag (Inhalte, Gestaltung der Folien, Verständlichkeit, Interaktivität)
- Ausarbeitung (Inhalte, roter Faden, sprachliche Darstellung, Rechtschreibung, eigener Beitrag)
- 50 : 50

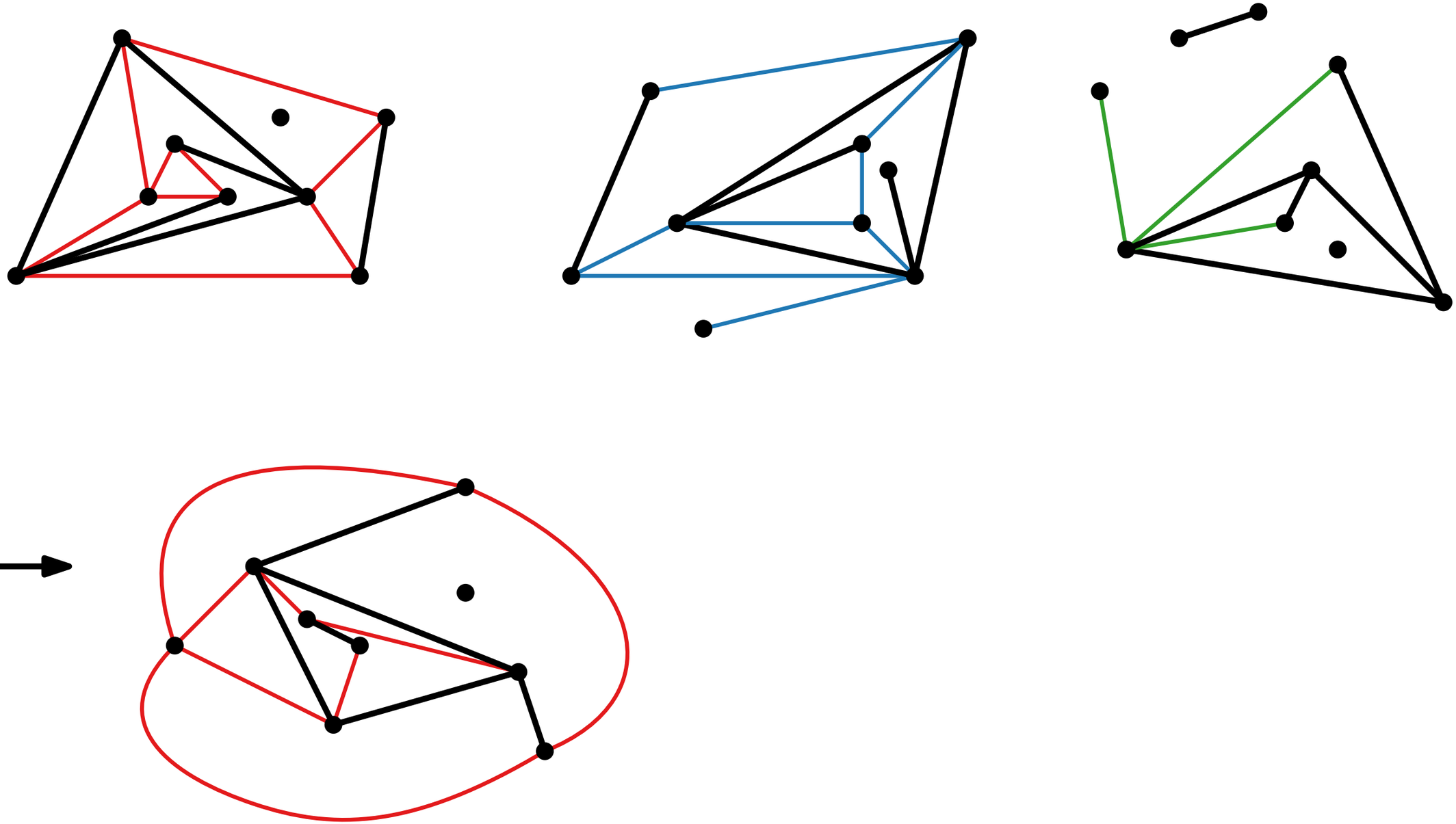
Themenübersicht

1. Parameterized Complexity of Simultaneous Planarity
2. Lower bounds based on the Exponential-Time Hypothesis
3. Constraint Logic: A Uniform Framework for Modeling Computation as Games
4. XALP-completeness of Parameterized Problems on Planar Graphs
5. Approximation Algorithms for NP-complete Problems on Planar Graphs
6. How to Morph Planar Graph Drawings
7. Sliding Squares in Parallel
8. Crossing Number of 3-Plane Drawings
9. Geometric Spanners of Bounded Tree-width
10. Efficient Parameterized Approximation
11. Kuratowski's Theorem

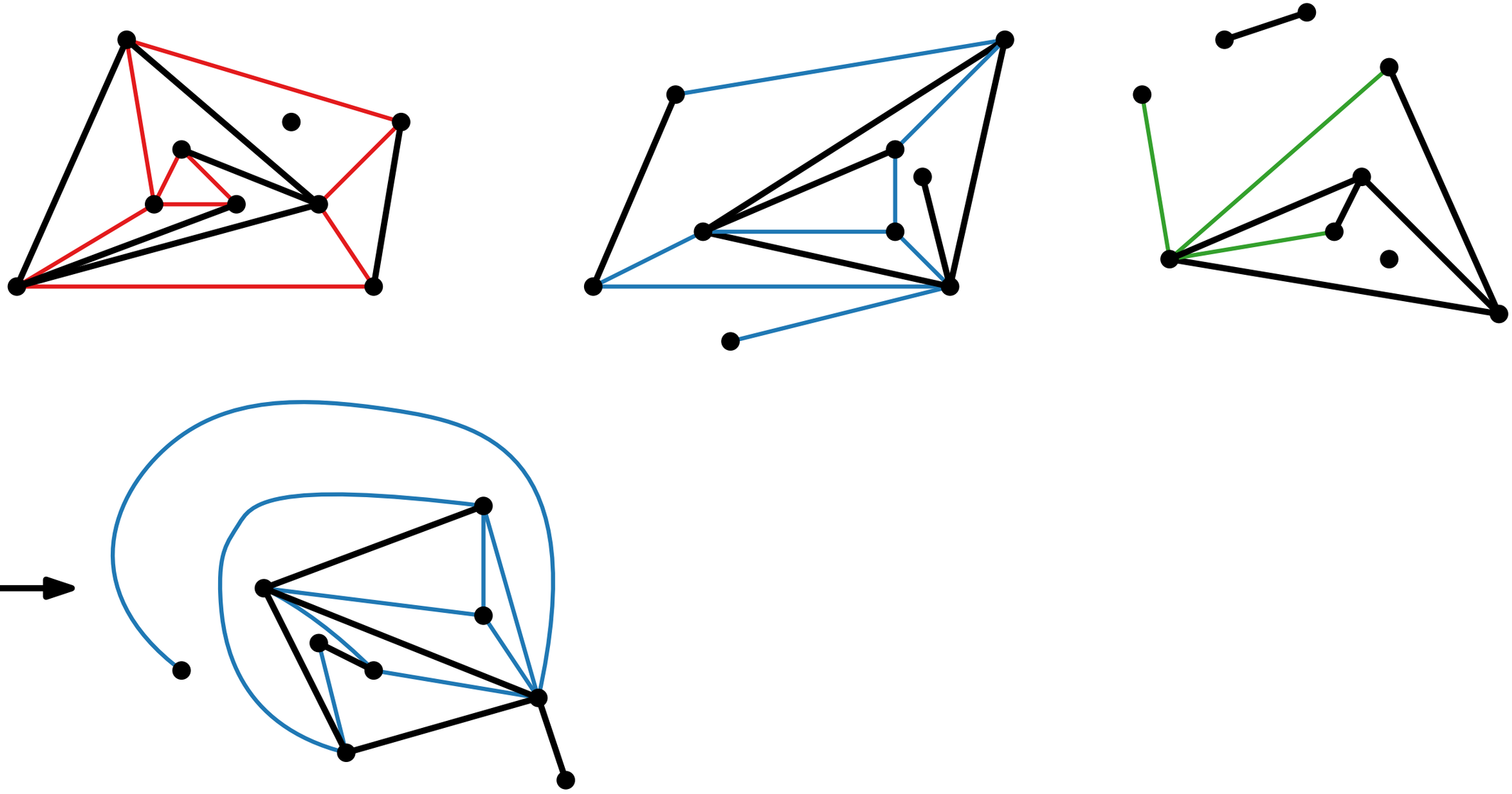
1. Parameterized Complexity of Simultaneous Planarity



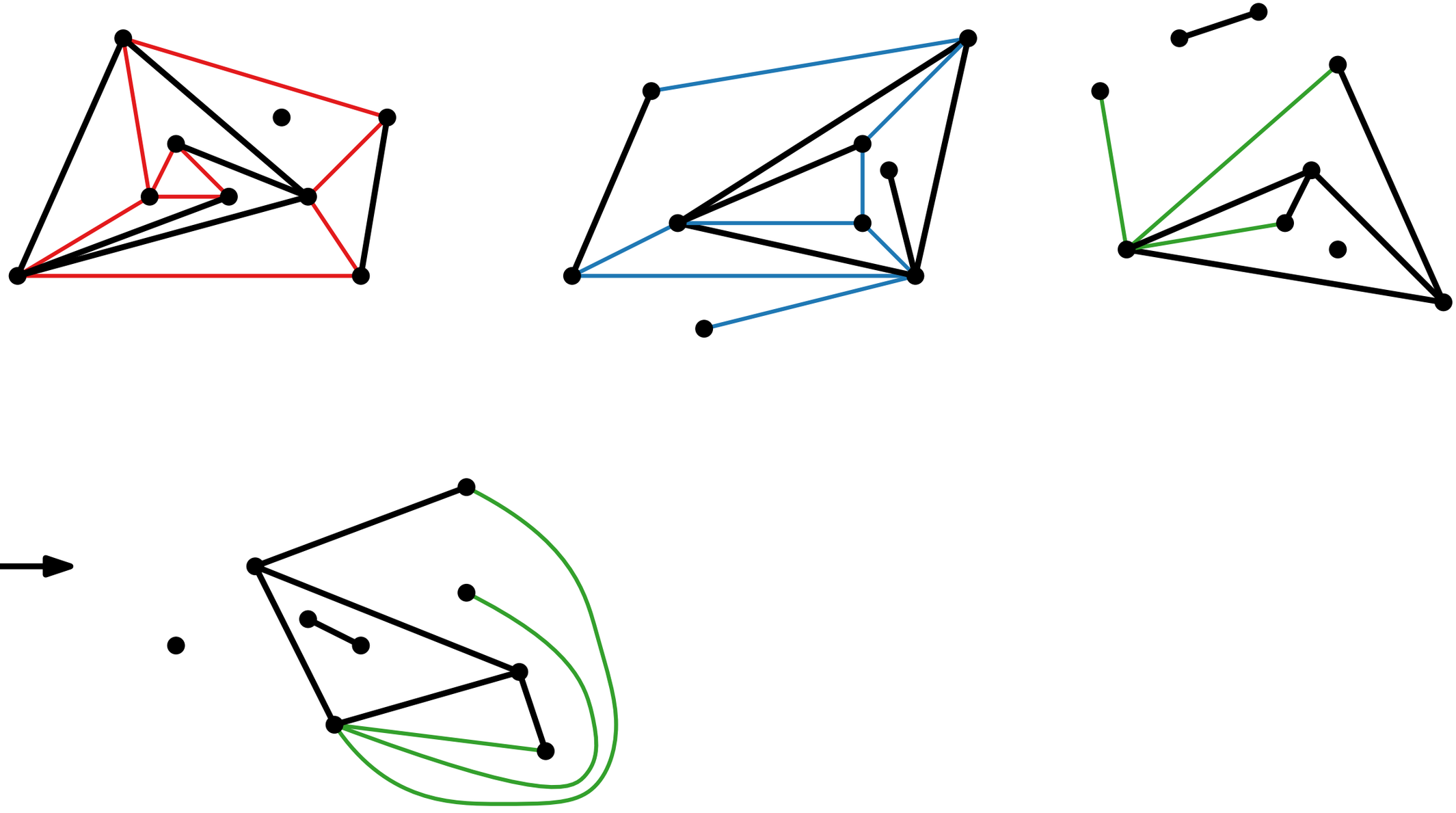
1. Parameterized Complexity of Simultaneous Planarity



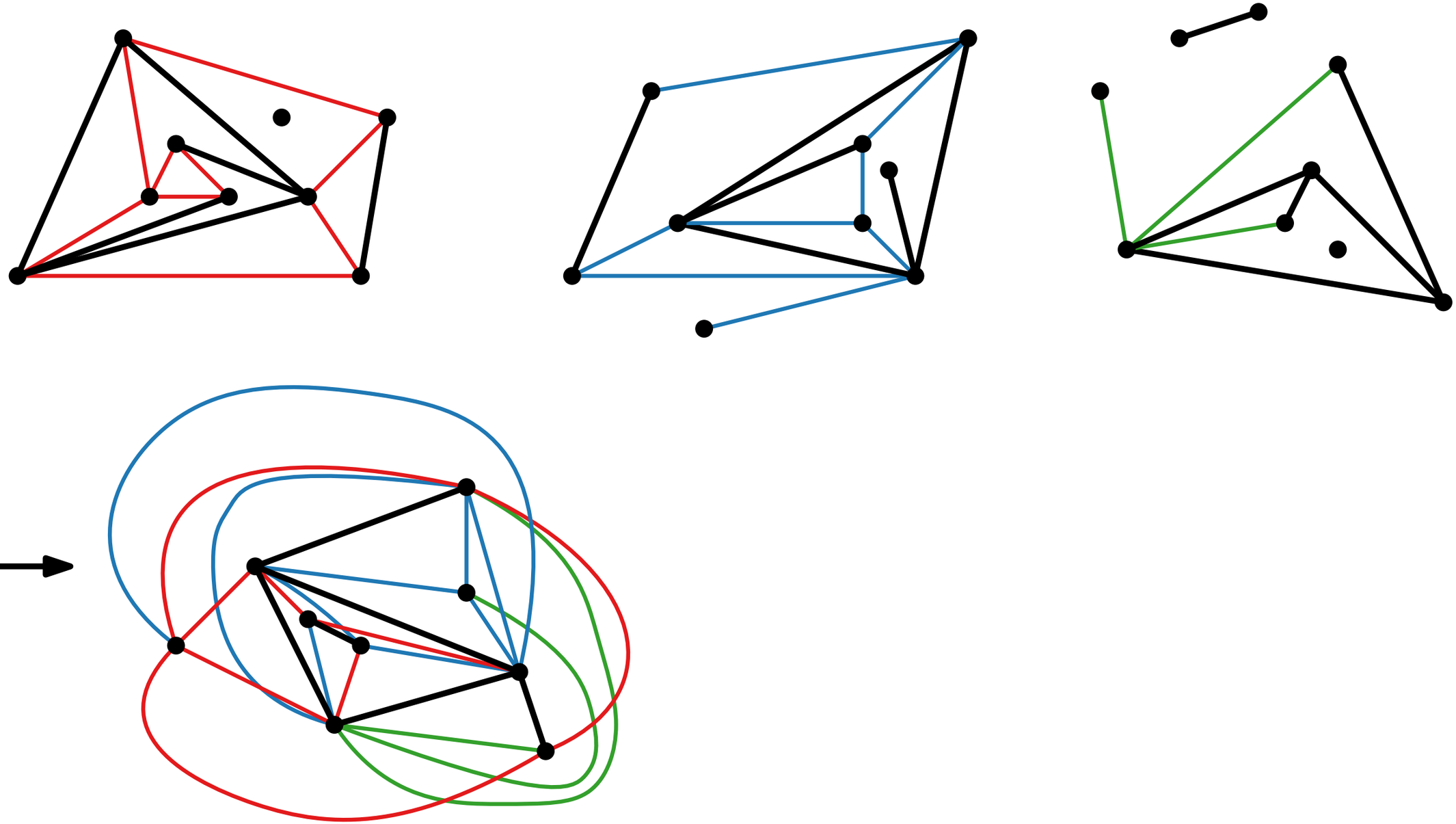
1. Parameterized Complexity of Simultaneous Planarity



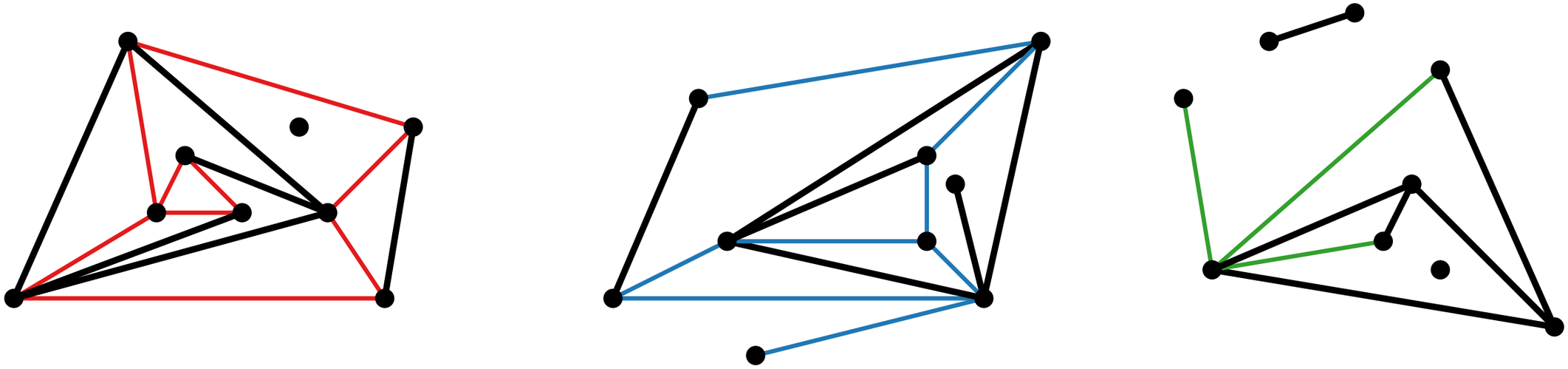
1. Parameterized Complexity of Simultaneous Planarity

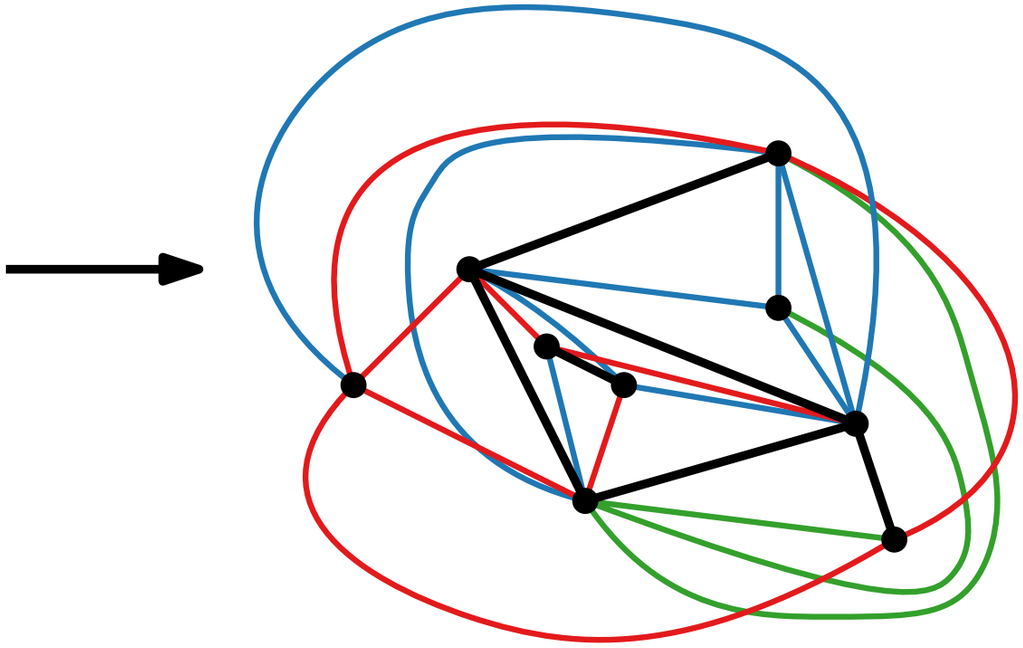


1. Parameterized Complexity of Simultaneous Planarity



1. Parameterized Complexity of Simultaneous Planarity

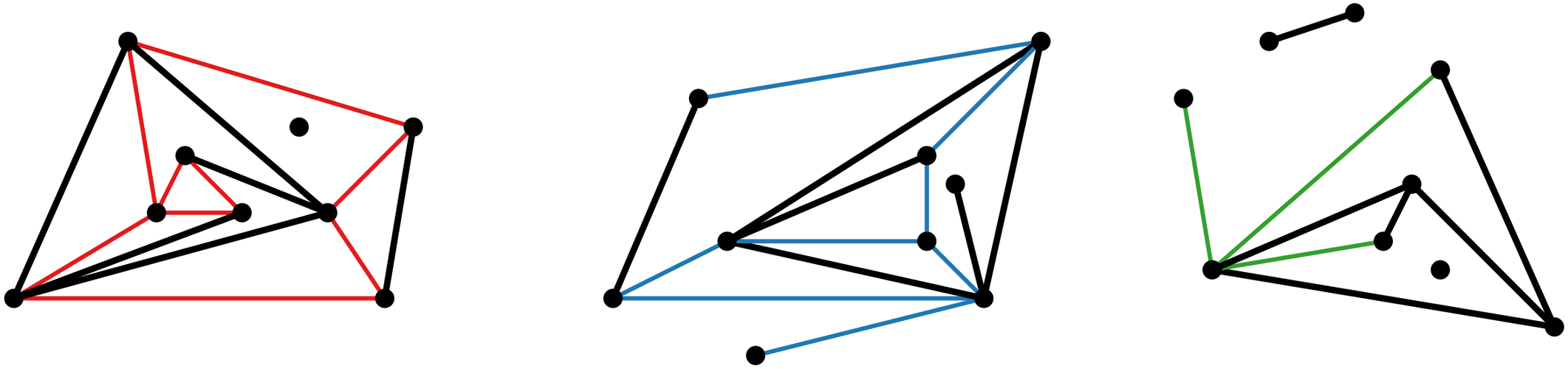


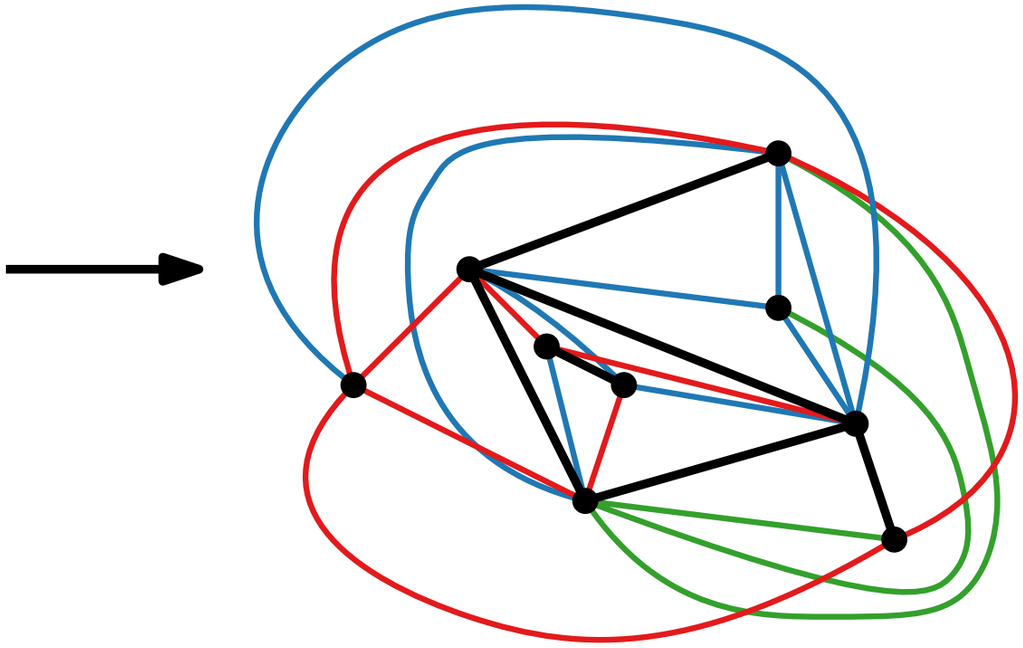


Simultaneous Embedding with Fixed Edges:

Eingabe: k planar graphs that have some edges in common.
Frage: Can the graphs be drawn in a planar way simultaneously?

1. Parameterized Complexity of Simultaneous Planarity



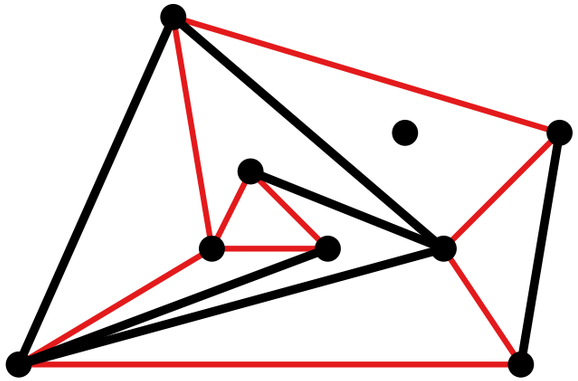


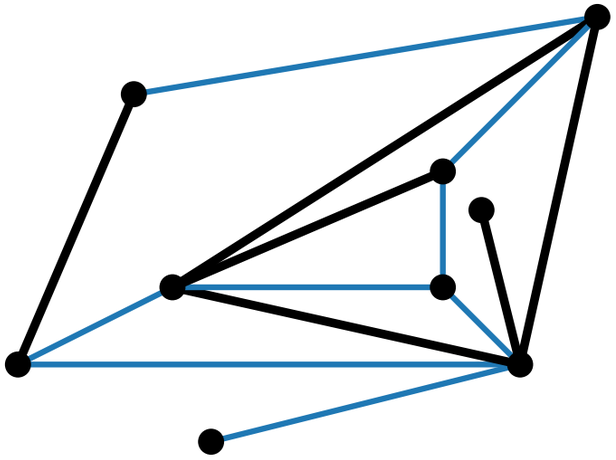
Simultaneous Embedding with Fixed Edges:

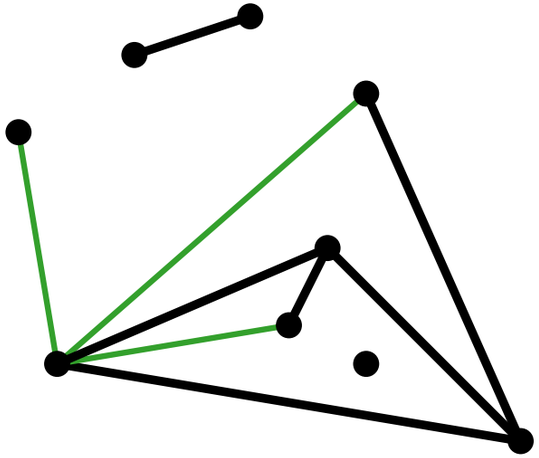
Eingabe: k planar graphs that have some edges in common.
Frage: Can the graphs be drawn in a planar way simultaneously?

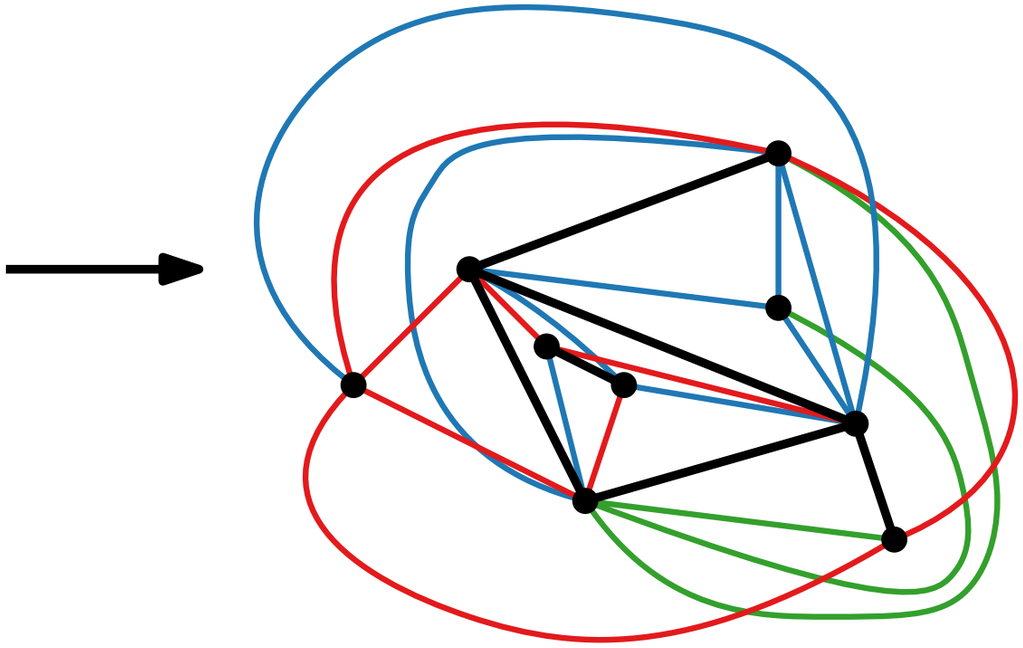
NP-schwer für $k \geq 3$, nicht bekannt für $k = 2$.

1. Parameterized Complexity of Simultaneous Planarity









Simultaneous Embedding with Fixed Edges:

Eingabe: k planar graphs that have some edges in common.
Frage: Can the graphs be drawn in a planar way simultaneously?

In this seminar: **What is the parameterized complexity of SEFE?**

2. Lower bounds based on (S)ETH

Let A be an NP-hard problem.

2. Lower bounds based on (S)ETH

Let A be an NP-hard problem.

If $P \neq NP$, there is no polynomial time algorithm for A.

2. Lower bounds based on (S)ETH

Let A be an NP-hard problem.

If $P \neq NP$, there is no polynomial time algorithm for A.

With a polynomial time reduction from A to some other problem B, we can transfer this **conditional lower bound** to B.

2. Lower bounds based on (S)ETH

Let A be an NP-hard problem.

If $P \neq NP$, there is no polynomial time algorithm for A.

With a polynomial time reduction from A to some other problem B, we can transfer this **conditional lower bound** to B.

Here: More fine-grained lower bounds based on other conditions:

2. Lower bounds based on (S)ETH

Let A be an NP-hard problem.

If $P \neq NP$, there is no polynomial time algorithm for A.

With a polynomial time reduction from A to some other problem B, we can transfer this **conditional lower bound** to B.

Here: More fine-grained lower bounds based on other conditions:

The **Strong Exponential Time Hypothesis (SETH)** implies that there is no algorithm for Satisfiability with runtime $(2-\varepsilon)^n \cdot \text{poly}(n)$, where n is the number of variables and $\varepsilon > 0$.

2. Lower bounds based on (S)ETH

Let A be an NP-hard problem.

If $P \neq NP$, there is no polynomial time algorithm for A.

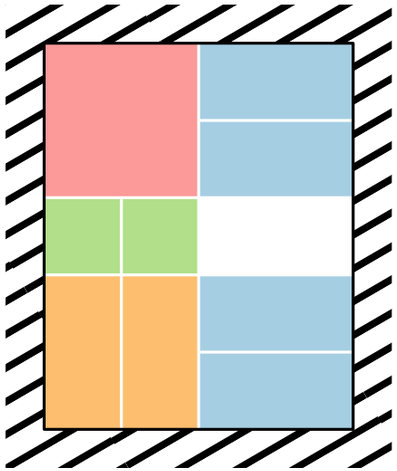
With a polynomial time reduction from A to some other problem B, we can transfer this **conditional lower bound** to B.

Here: More fine-grained lower bounds based on other conditions:

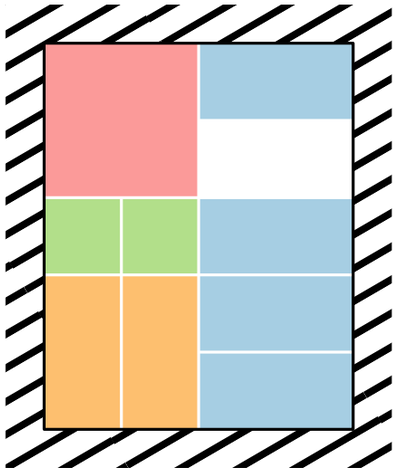
The **Strong Exponential Time Hypothesis (SETH)** implies that there is no algorithm for Satisfiability with runtime $(2-\varepsilon)^n \cdot \text{poly}(n)$, where n is the number of variables and $\varepsilon > 0$.

With a suitable reduction from Satisfiability to some other problem B, we can obtain a similiar lower bound for the runtime of B, assuming SETH.

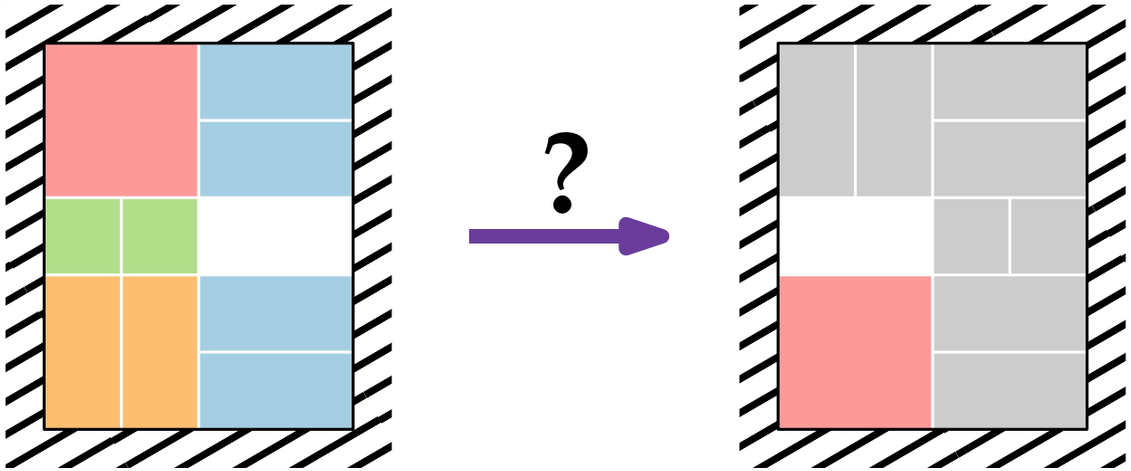
3. Constraint Logic



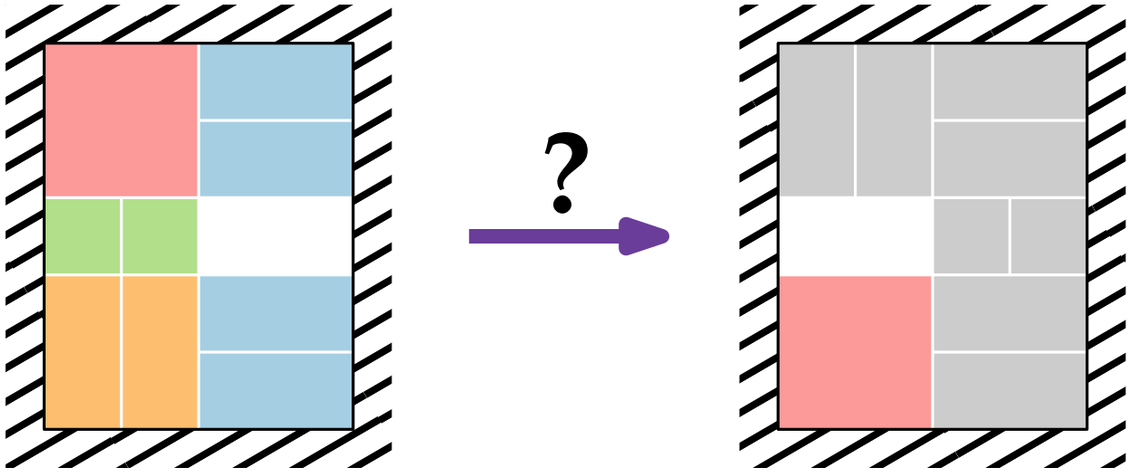
3. Constraint Logic



3. Constraint Logic

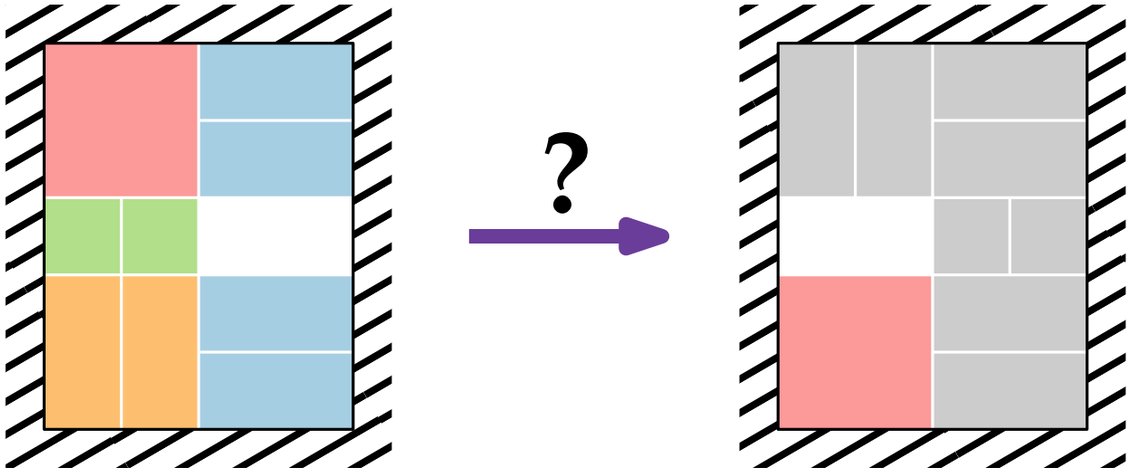


3. Constraint Logic

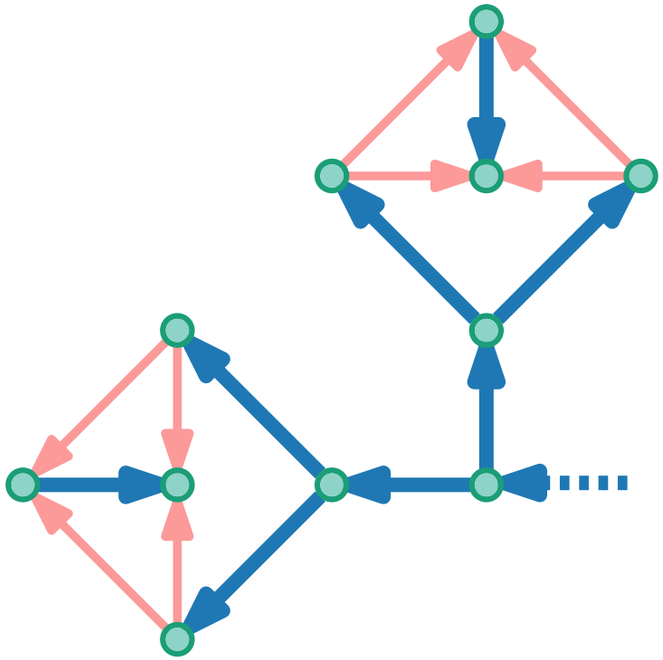


[Hearn, Demaine. TCS, 2005]

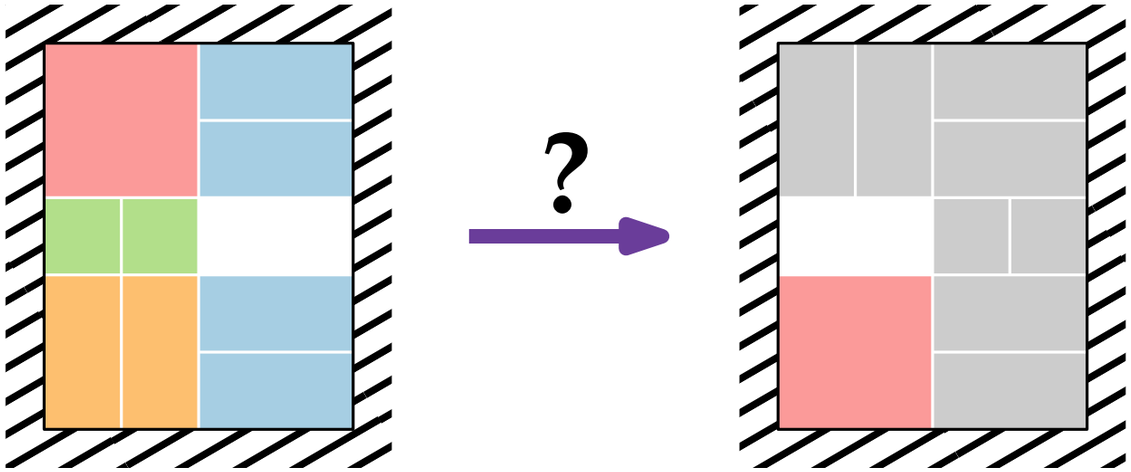
3. Constraint Logic



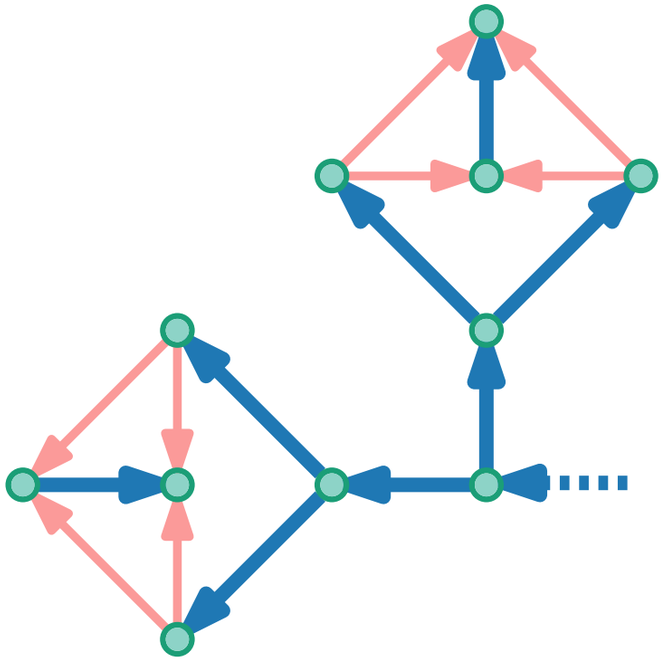
[Hearn, Demaine. TCS, 2005]



3. Constraint Logic



[Hearn, Demaine. TCS, 2005]



4. XNLP/XALP-complete Problems for Planar Graphs

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

4. XNLP/XALP-complete Problems for Planar Graphs

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time;

This is an advanced topic intended for students who have completed the Exact Algorithms course!

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time; and

XP if it can be solved in $f(k) \cdot n^{g(k)}$ time,

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time; and

XP if it can be solved in $f(k) \cdot n^{g(k)}$ time,

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

Recently, the complexity class **XNLP** has been introduced and shown to be the "correct" class for many problems whose solutions have a *linear* structure and that are XP, but likely not FPT.

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time; and

XP if it can be solved in $f(k) \cdot n^{g(k)}$ time,

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

Recently, the complexity class **XNLP** has been introduced and shown to be the "correct" class for many problems whose solutions have a *linear* structure and that are XP, but likely not FPT.

small path-width

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time; and

XP if it can be solved in $f(k) \cdot n^{g(k)}$ time,

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

Recently, the complexity class **XNLP** has been introduced and shown to be the "correct" class for many problems whose solutions have a *linear* structure and that are XP, but likely not FPT.

small path-width

XALP is the analogue for problems whose solutions have a *tree-like* structure.

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time; and

XP if it can be solved in $f(k) \cdot n^{g(k)}$ time,

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

Recently, the complexity class **XNLP** has been introduced and shown to be the "correct" class for many problems whose solutions have a *linear* structure and that are XP, but likely not FPT.

small path-width

XALP is the analogue for problems whose solutions have a *tree-like* structure.

small tree-width

4. XNLP/XALP-complete Problems for Planar Graphs

This is an advanced topic intended for students who have completed the Exact Algorithms course!

Recall: A parameterized problem is called

FPT if it can be solved in $f(k) \cdot n^c$ time; and

XP if it can be solved in $f(k) \cdot n^{g(k)}$ time,

where n is the input size, k is the parameter, c is a constant, and f, g are some computable functions.

Recently, the complexity class **XNLP** has been introduced and shown to be the "correct" class for many problems whose solutions have a *linear* structure and that are XP, but likely not FPT.

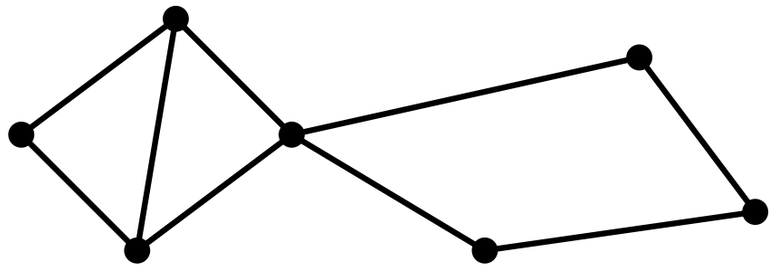
small path-width

XALP is the analogue for problems whose solutions have a *tree-like* structure.

small tree-width

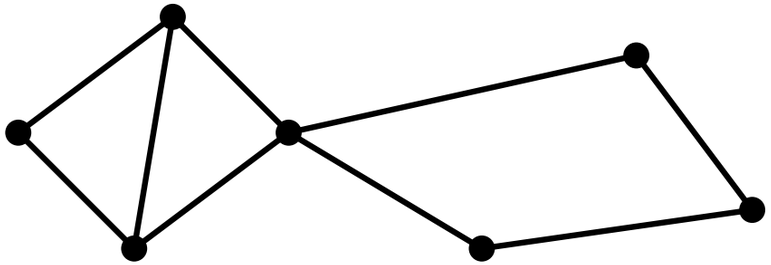
Here: XNLP/XALP-completeness of several problems defined for planar graphs.

5. Baker's Technique

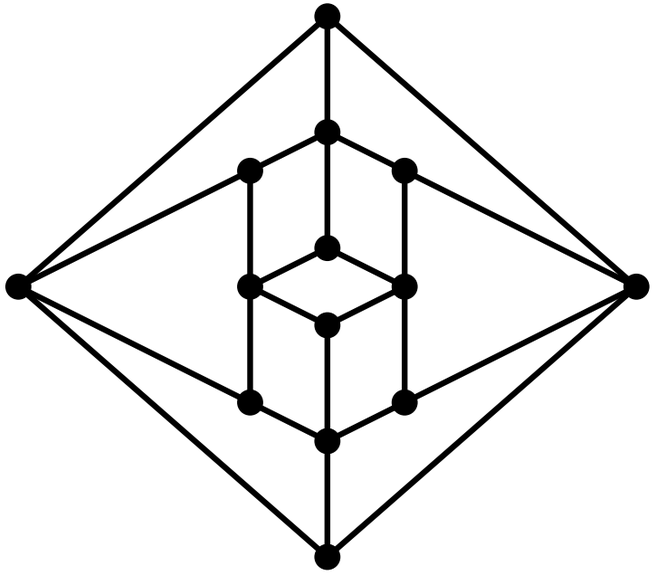


(1-)outerplaner graph

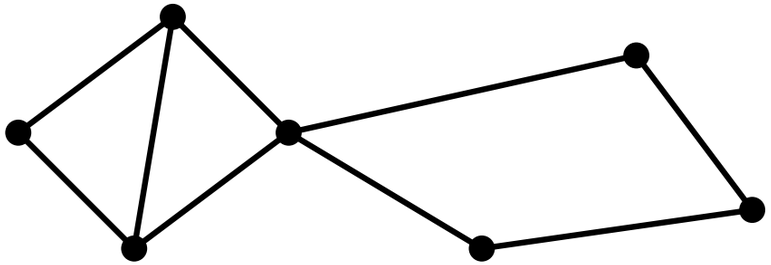
5. Baker's Technique



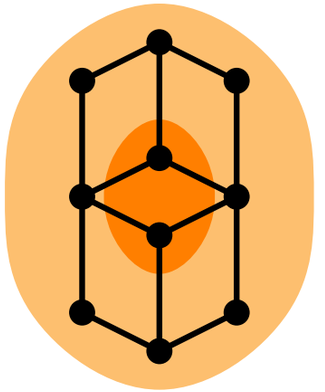
(1-)outerplaner graph



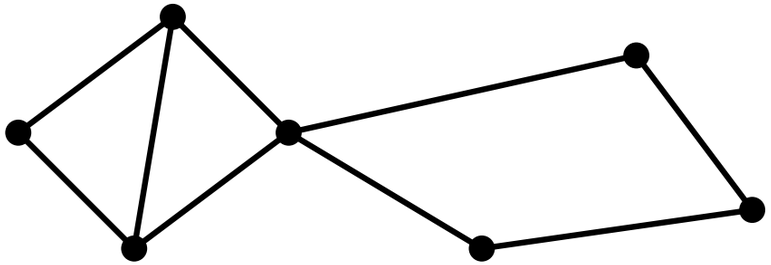
5. Baker's Technique



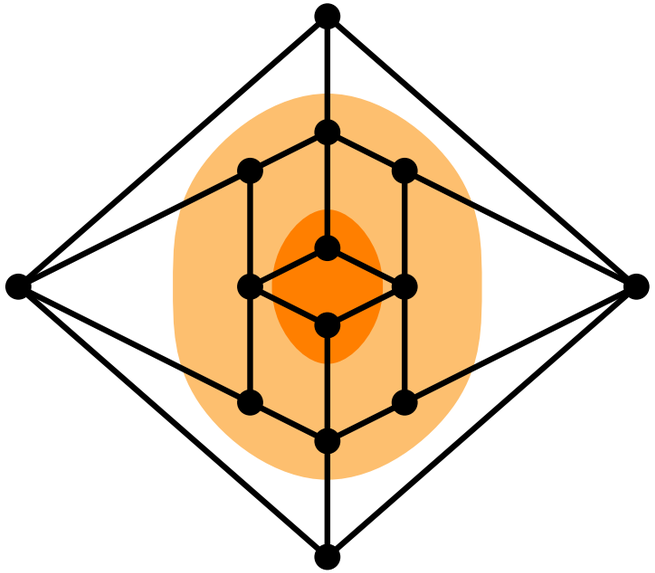
(1-)outerplaner graph



5. Baker's Technique

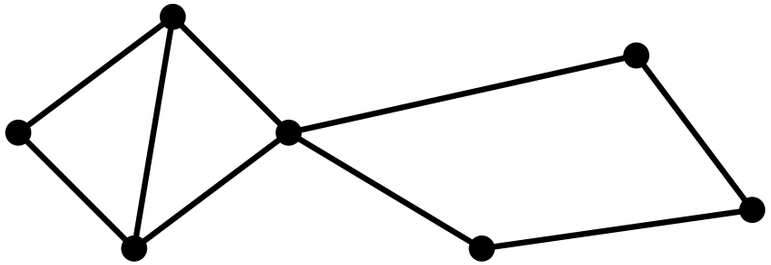


(1-)outerplaner graph

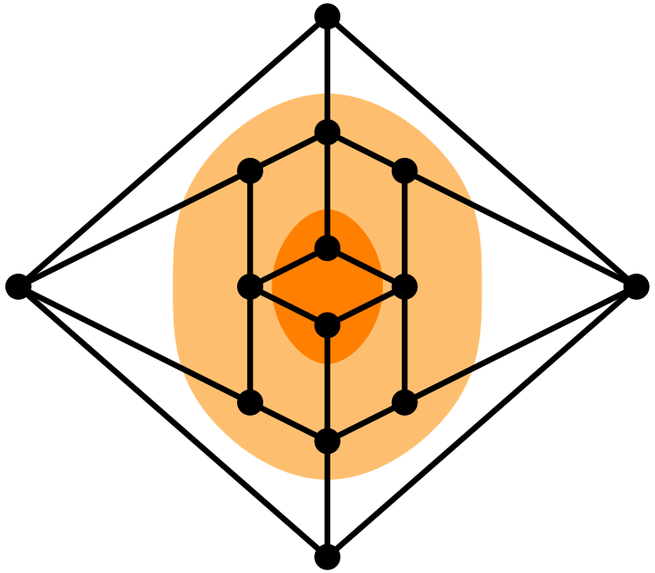


3-outerplanar

5. Baker's Technique



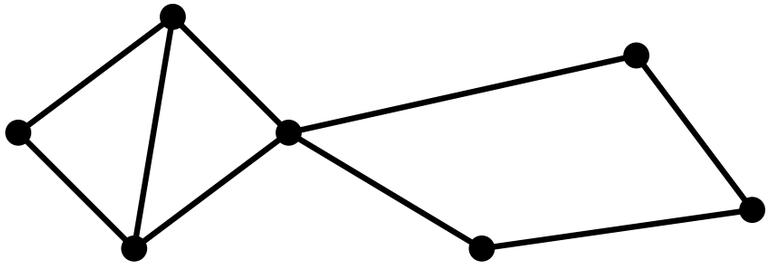
(1-)outerplaner graph



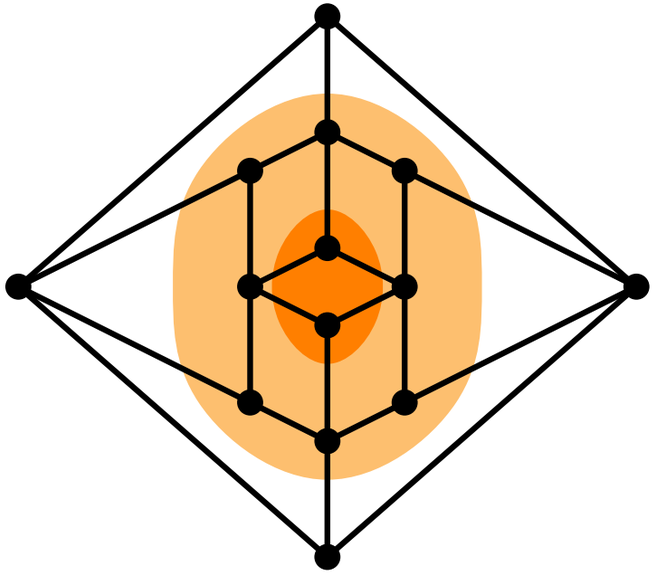
3-outerplanar

Many NP-hard problems can be solved efficiently on k-outerplanar graphs for a fixed k

5. Baker's Technique



(1-)outerplaner graph

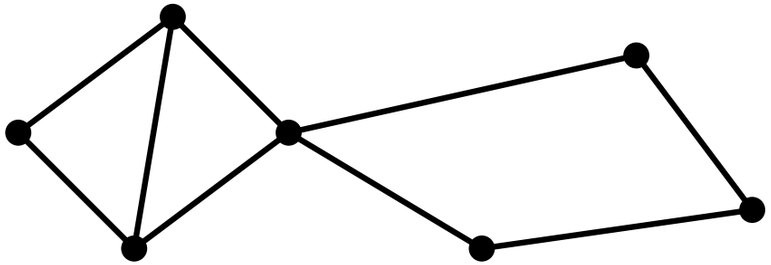


3-outerplanar

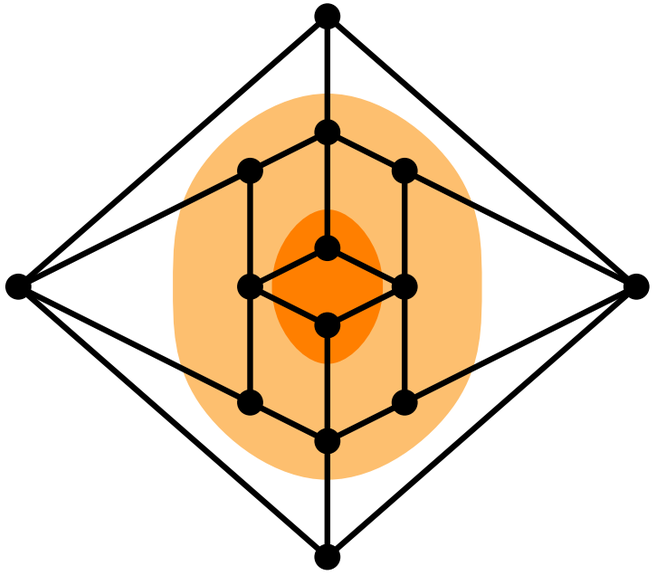
Many NP-hard problems can be solved efficiently on k-outerplanar graphs for a fixed k

Baker's Technique: A general approach to use these algorithms to find approximation algorithms for planar graphs

5. Baker's Technique



(1-)outerplaner graph



3-outerplanar

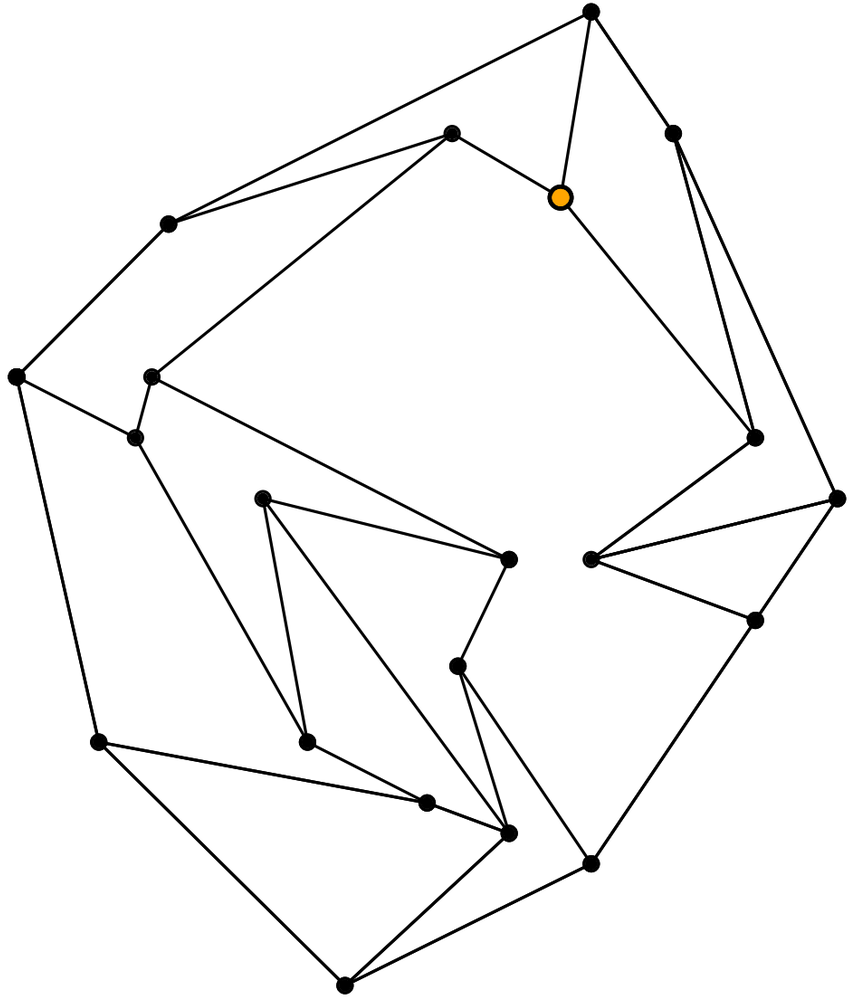
Many NP-hard problems can be solved efficiently on k-outerplanar graphs for a fixed k

Baker's Technique: A general approach to use these algorithms to find approximation algorithms for planar graphs

Task: Explain Baker's Technique with an illustrative example

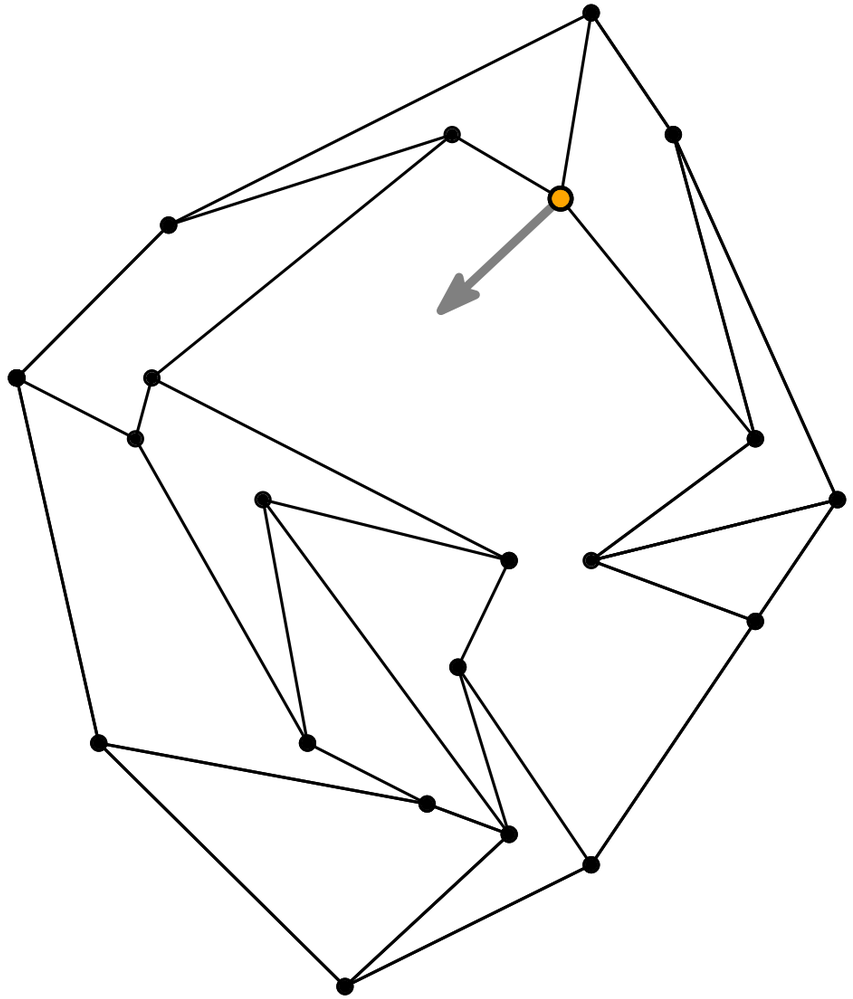
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



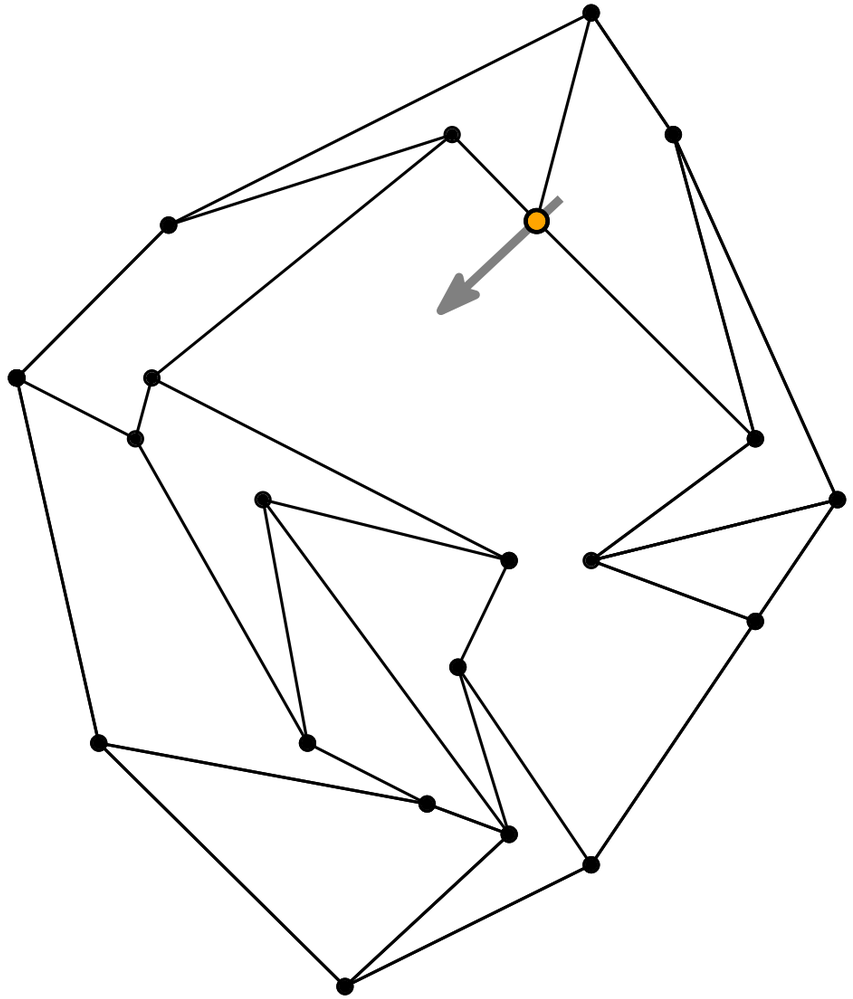
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



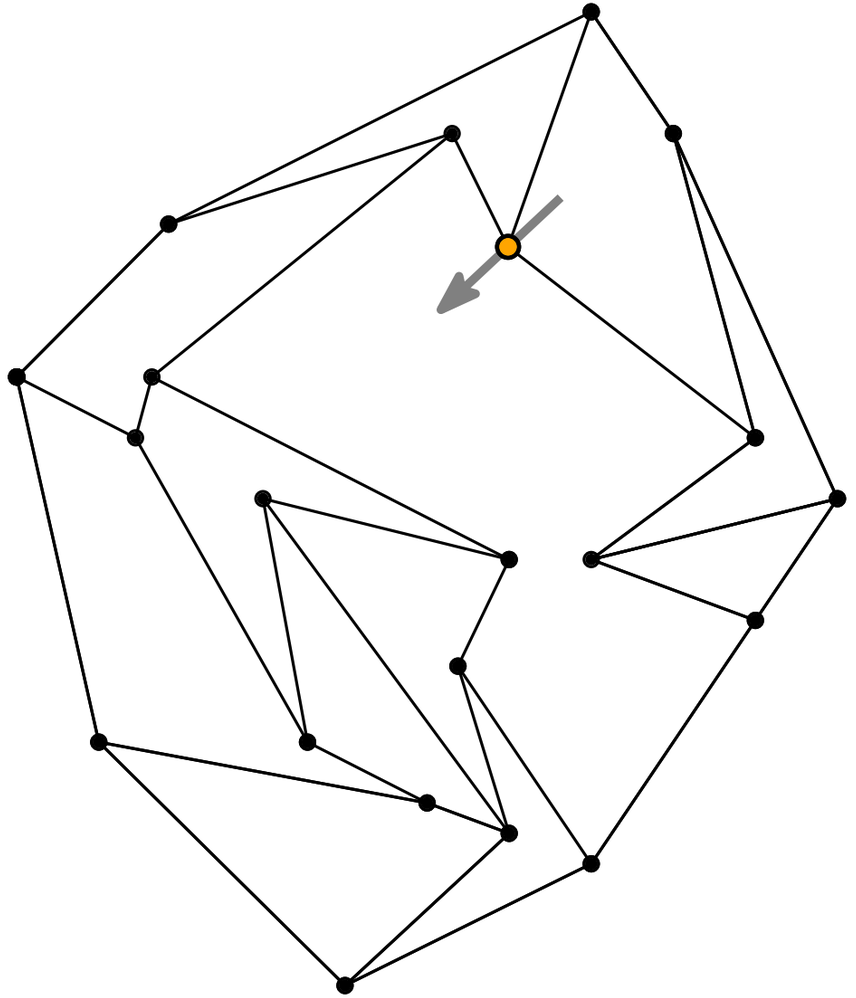
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



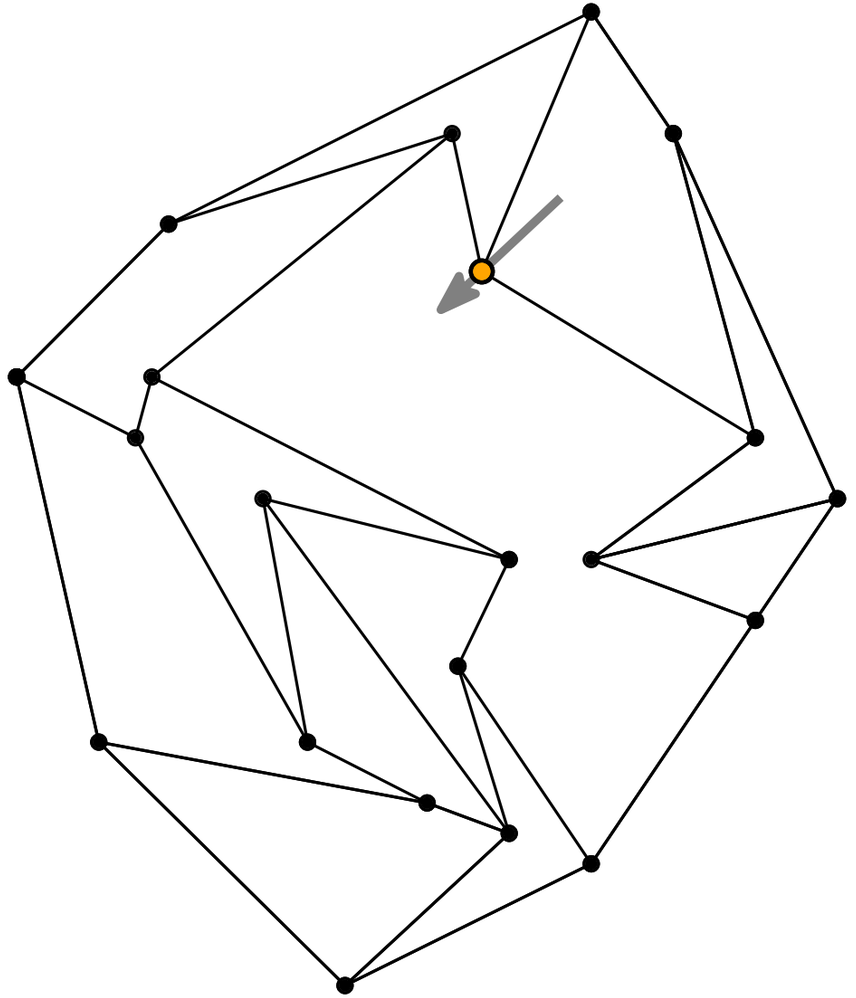
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



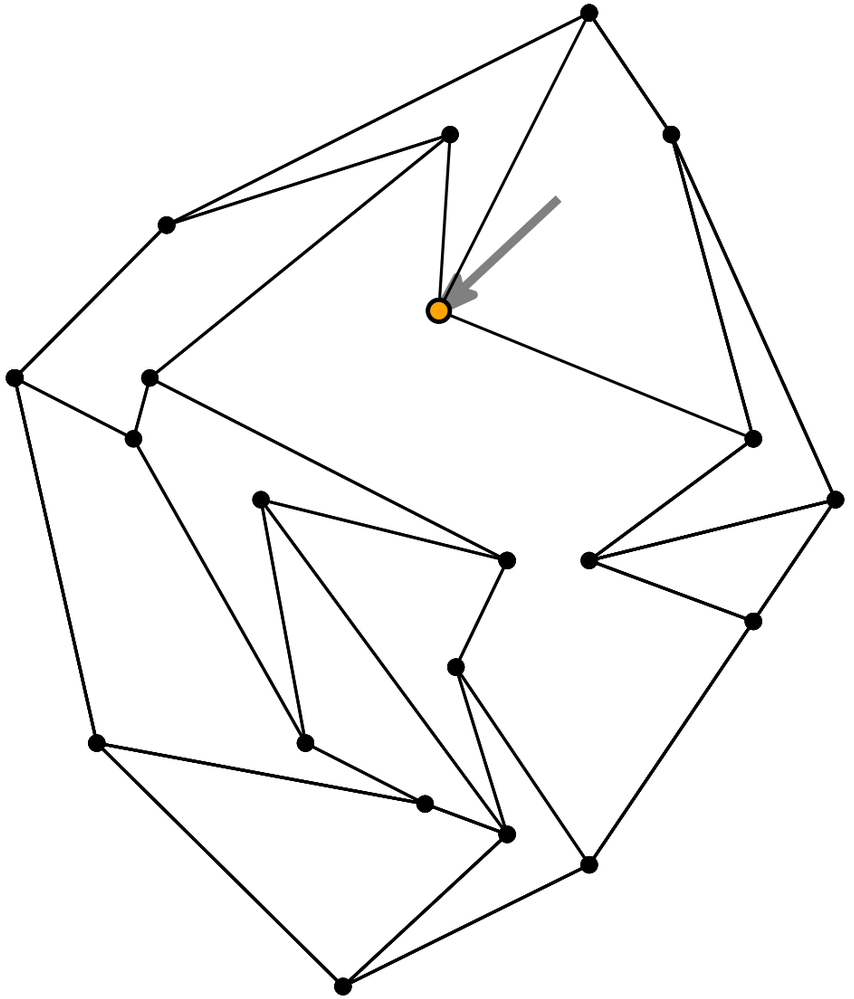
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



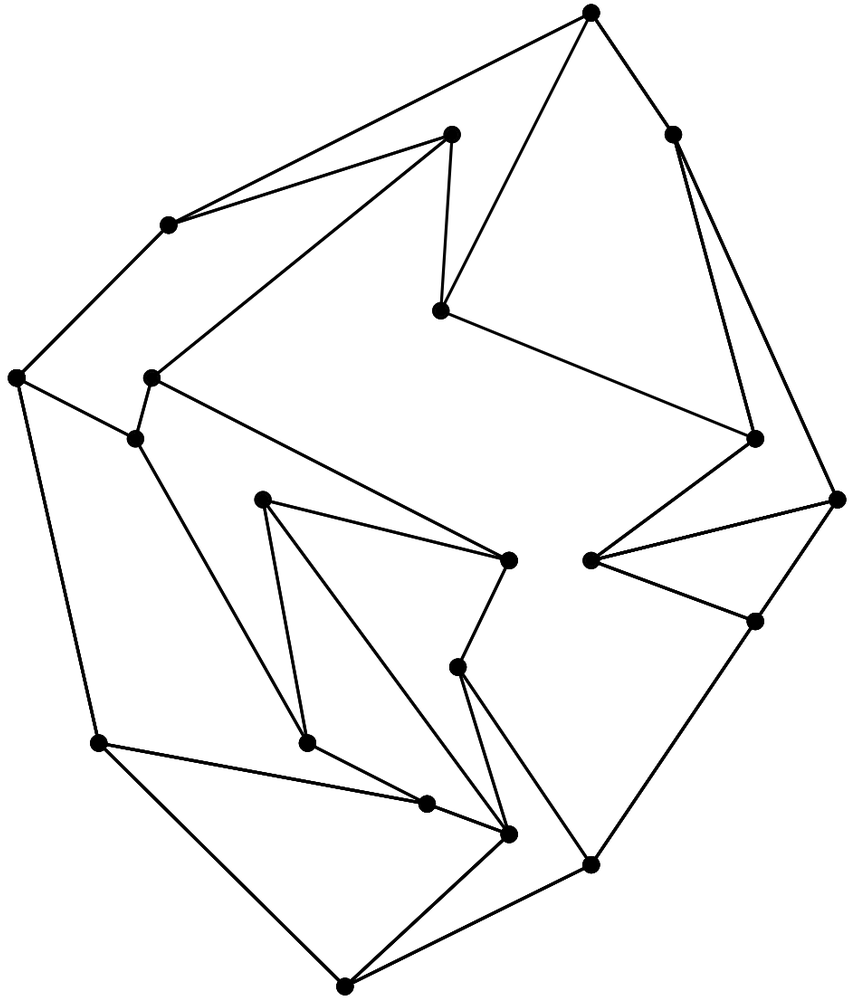
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



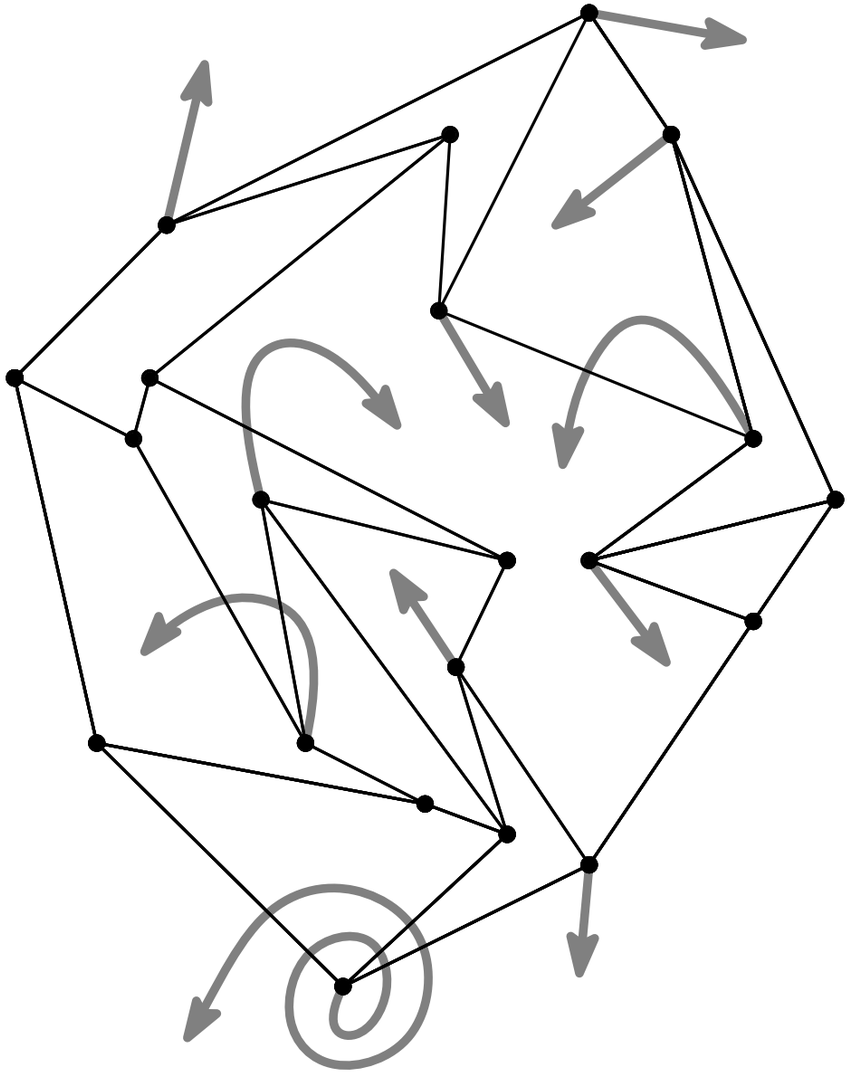
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



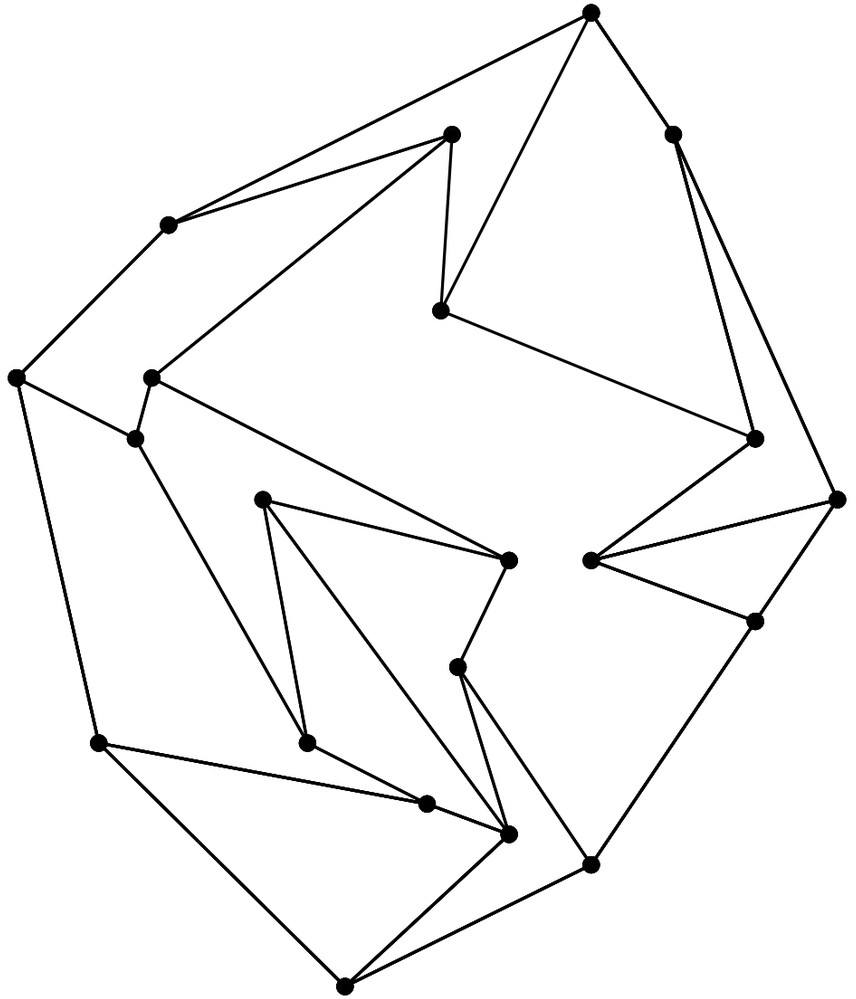
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



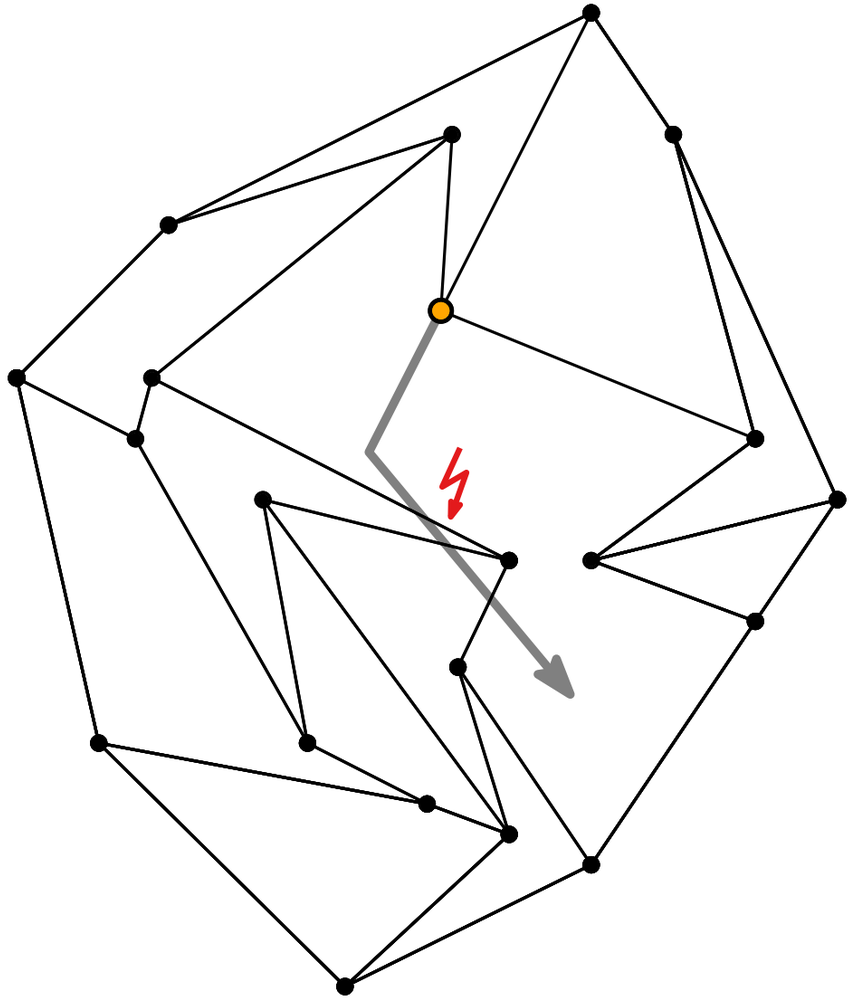
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



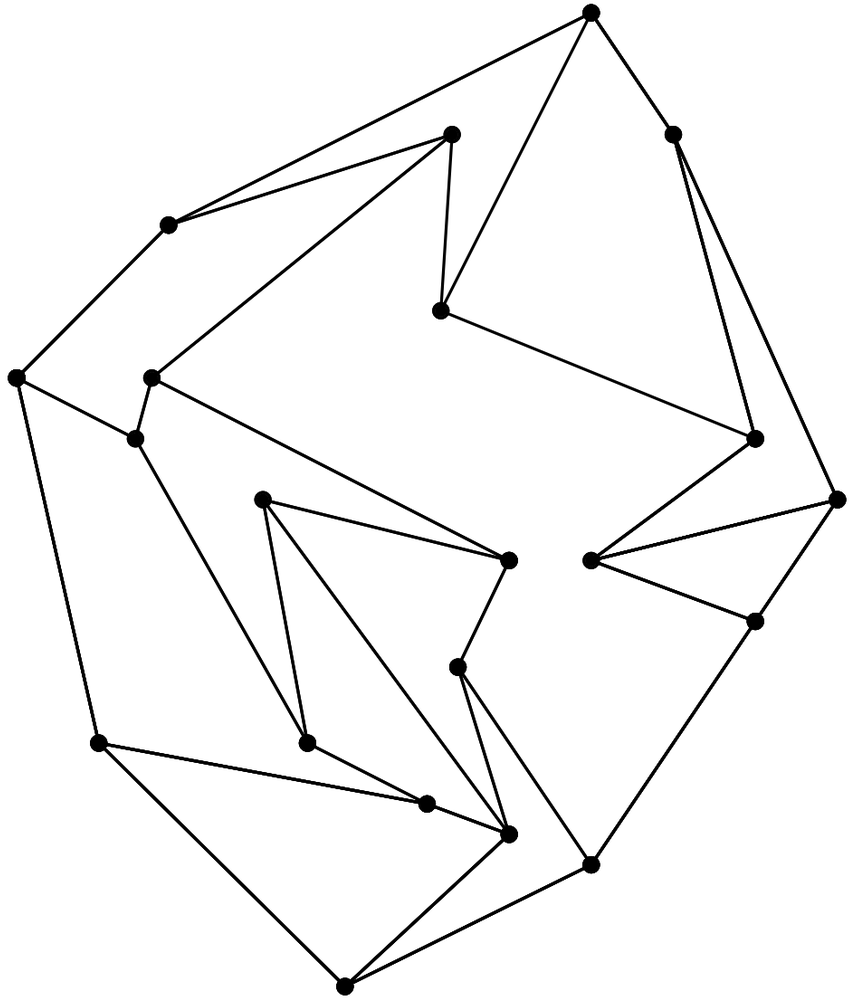
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



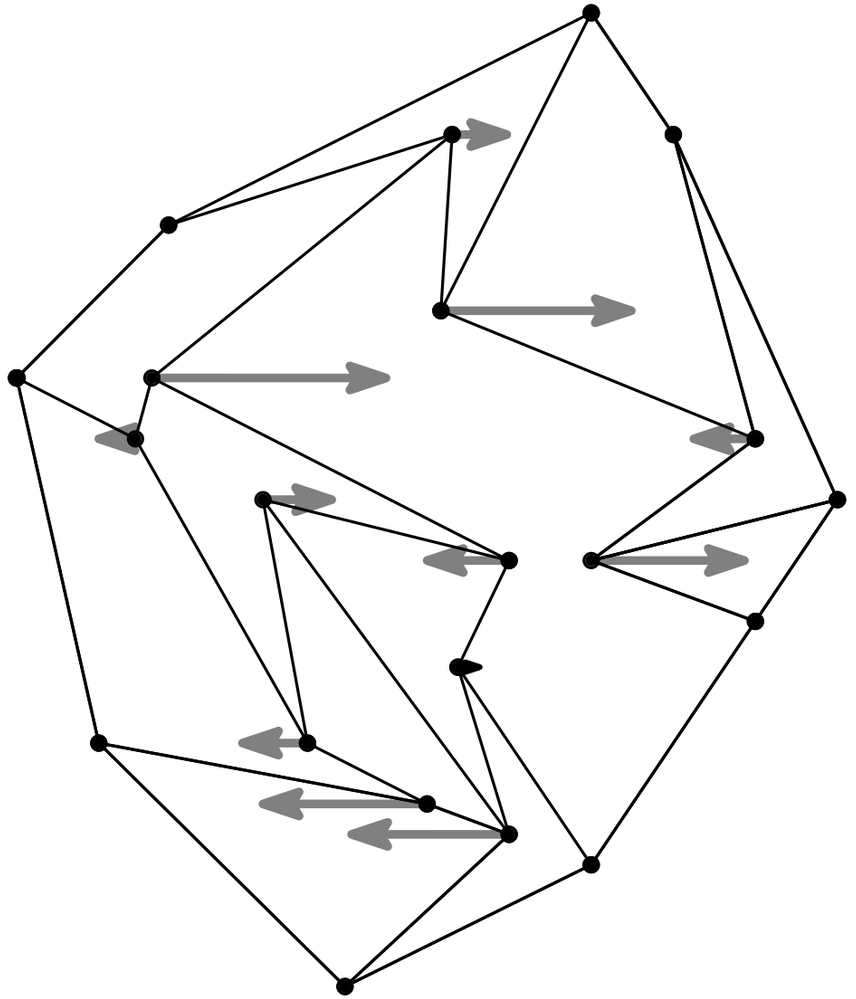
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



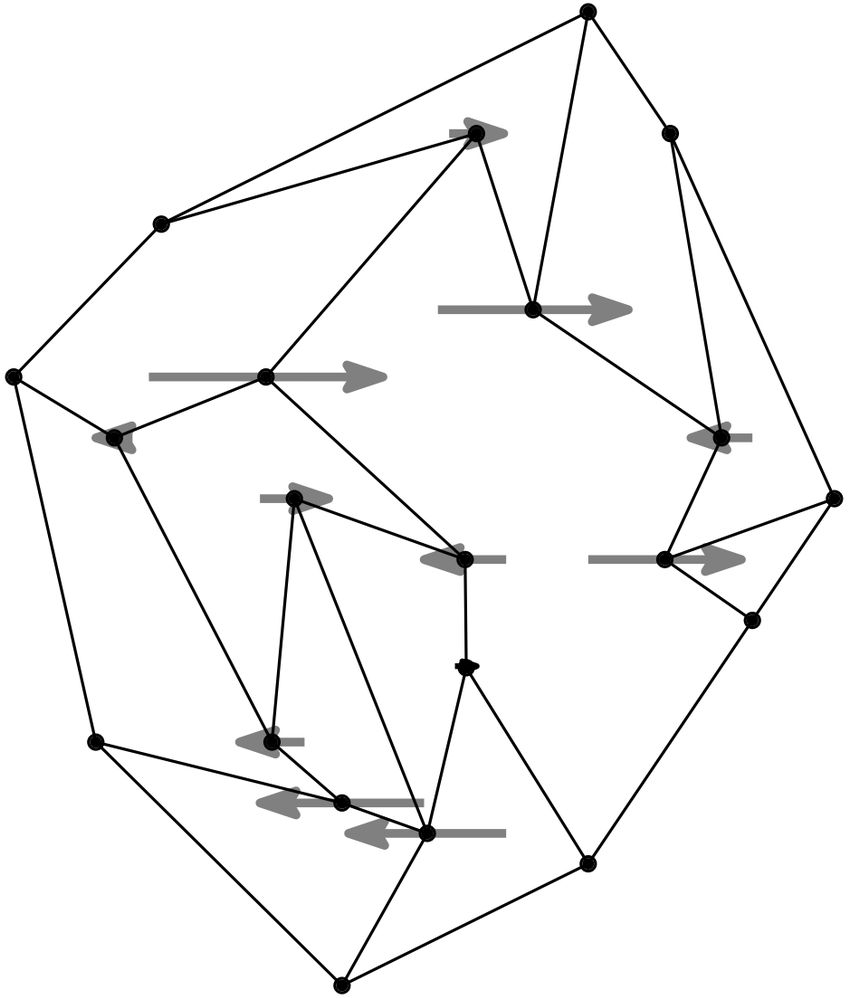
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



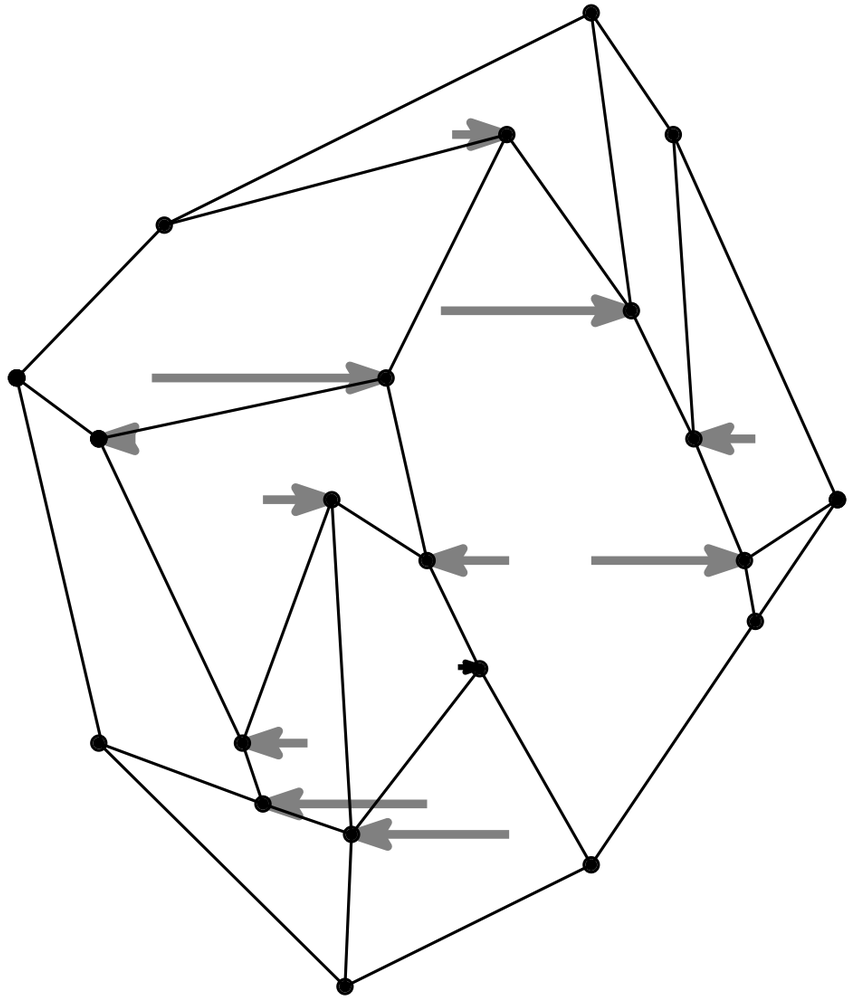
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



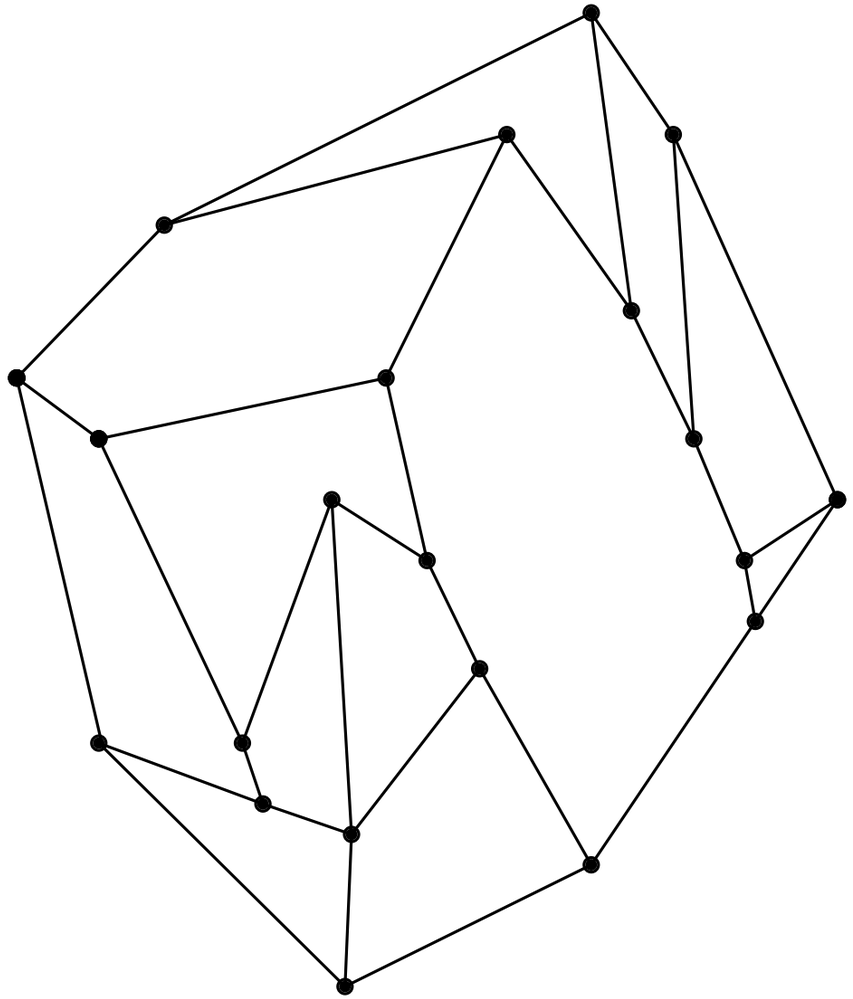
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



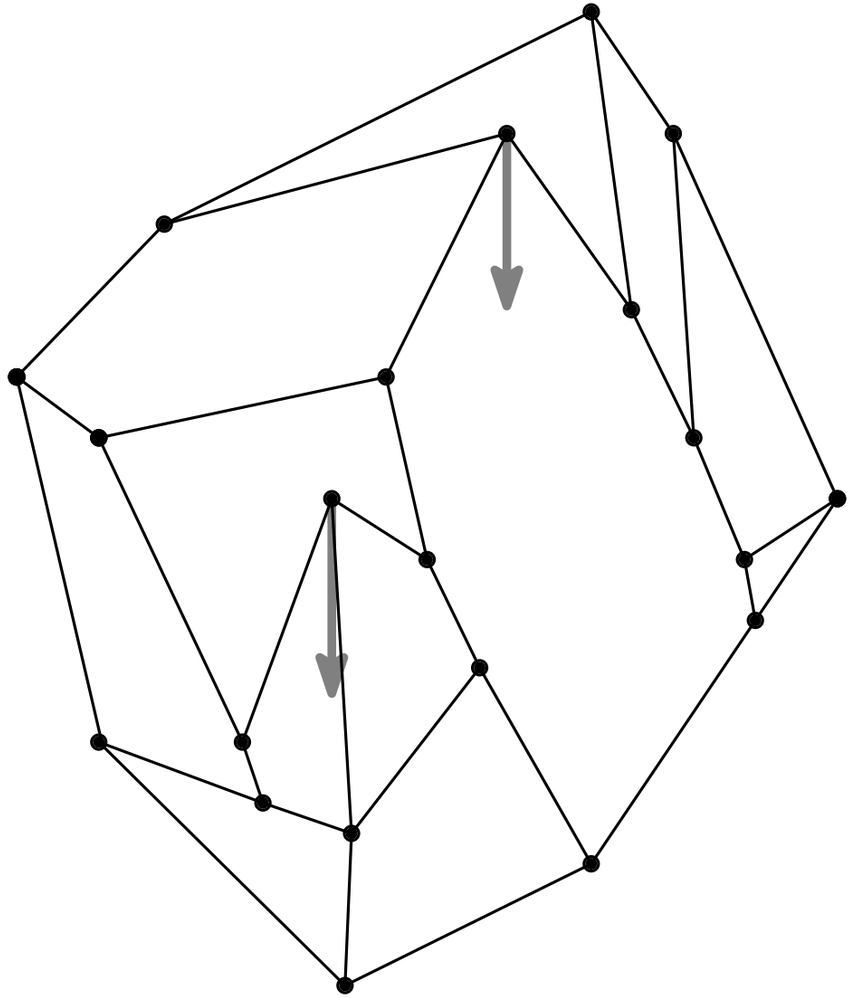
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



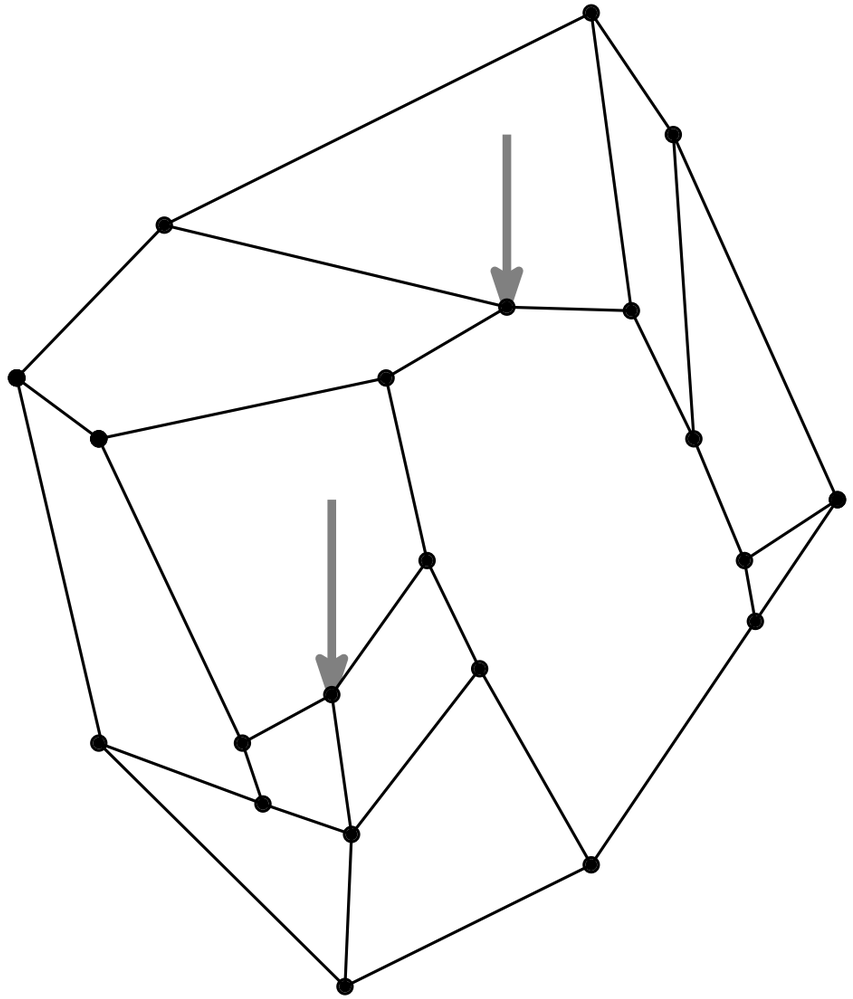
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



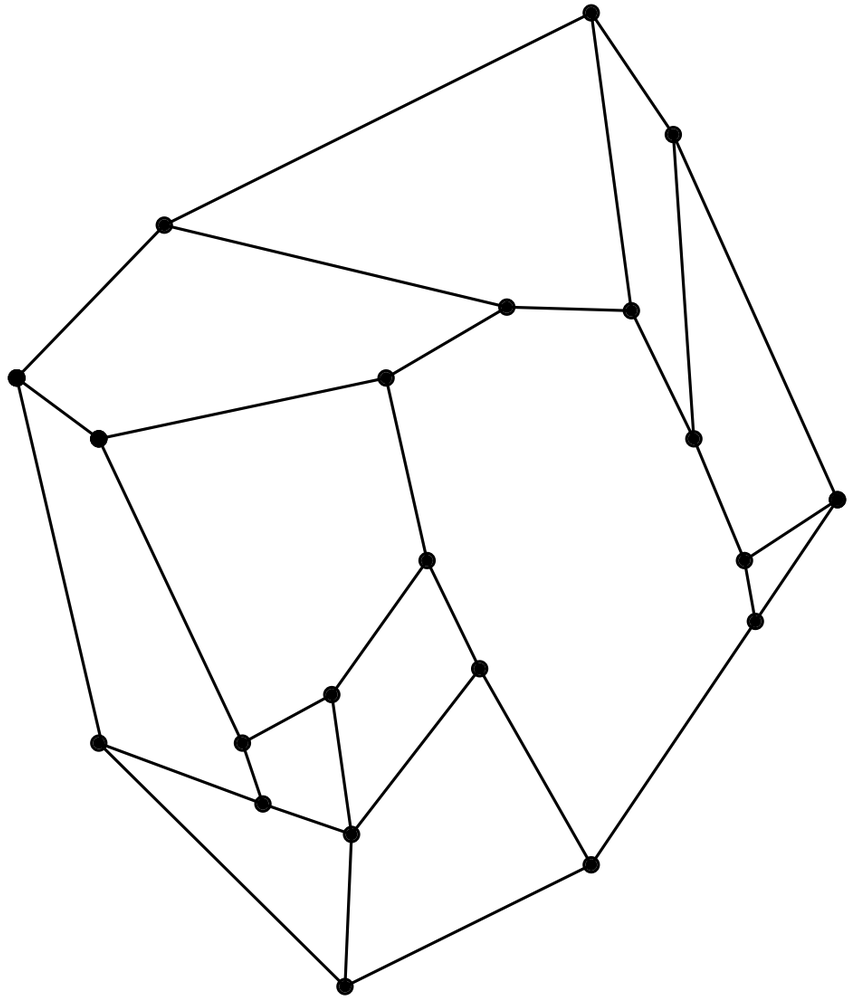
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



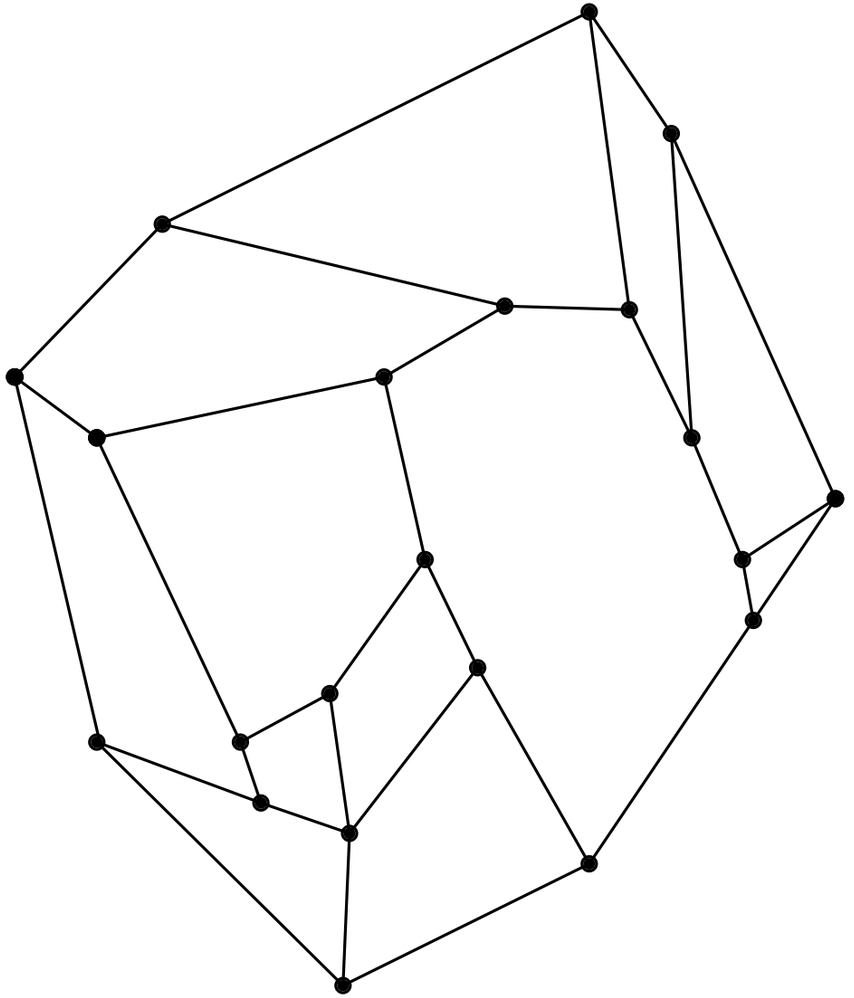
6. How to Morph Planar Graph Drawings

Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



6. How to Morph Planar Graph Drawings

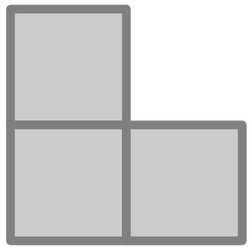
Morph: continuous deformation of a graph drawing that preserves straight-line crossing-free edges.



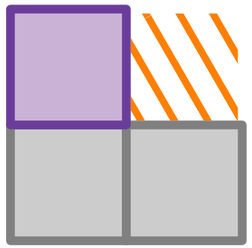
Questions:

- Does there always exist a morph between two crossing-free drawings of the same graph?
- How can it be computed and encoded?

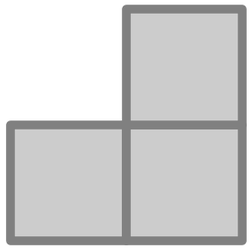
7. Sliding Squares in Parallel



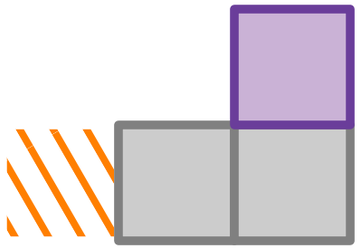
7. Sliding Squares in Parallel



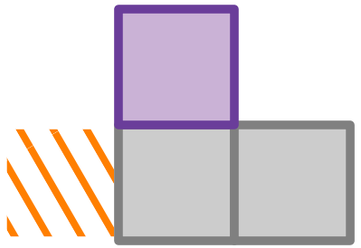
7. Sliding Squares in Parallel



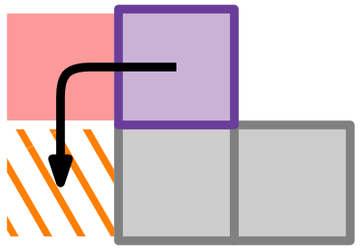
7. Sliding Squares in Parallel



7. Sliding Squares in Parallel

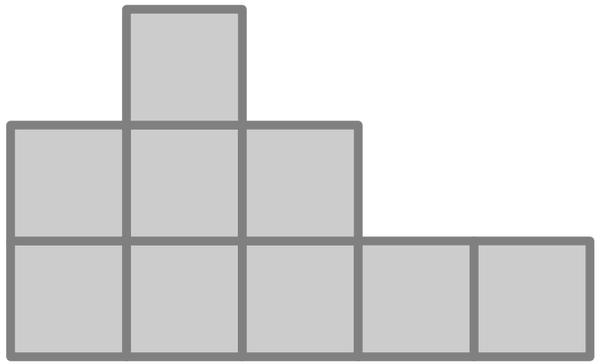


7. Sliding Squares in Parallel

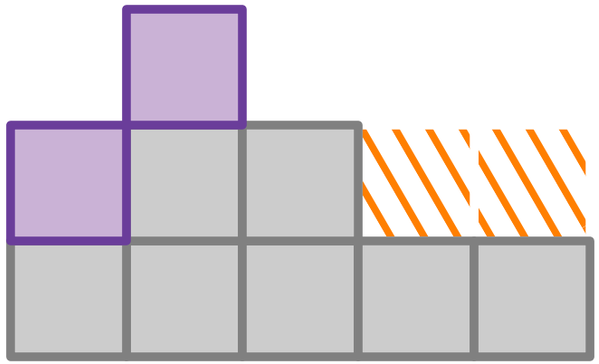


7. Sliding Squares in Parallel

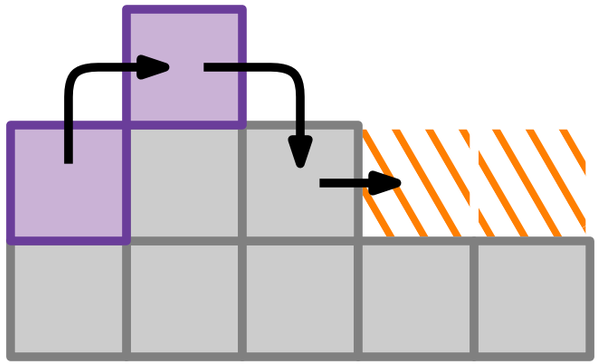
7. Sliding Squares in Parallel



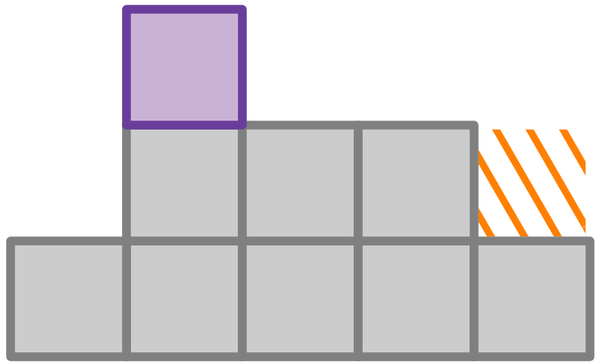
7. Sliding Squares in Parallel



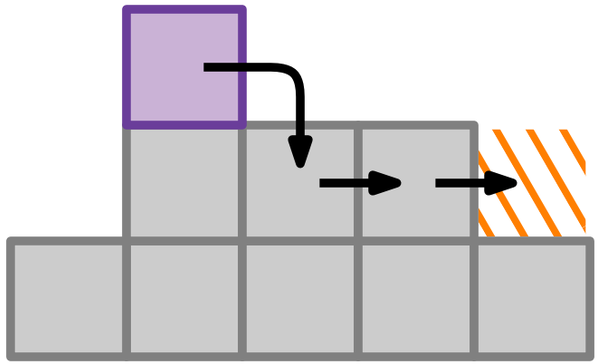
7. Sliding Squares in Parallel



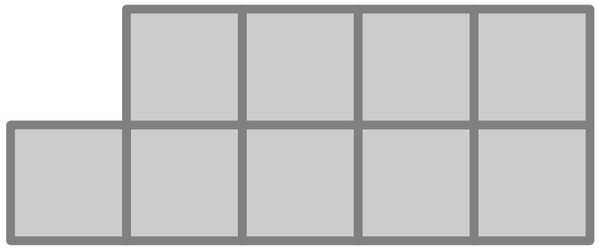
7. Sliding Squares in Parallel



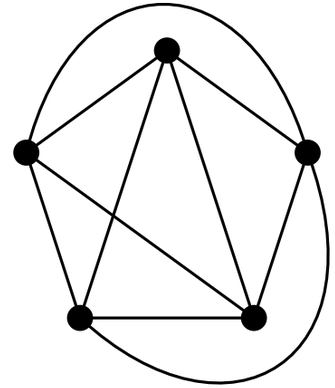
7. Sliding Squares in Parallel



7. Sliding Squares in Parallel

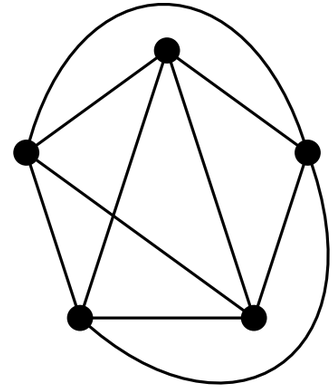


8. Crossing Number of 3-Plane Drawings



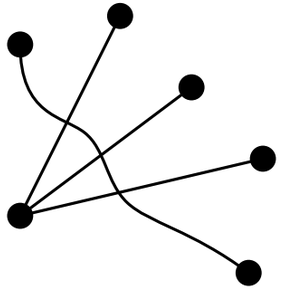
How many edges can a 1-planar graph have?

8. Crossing Number of 3-Plane Drawings

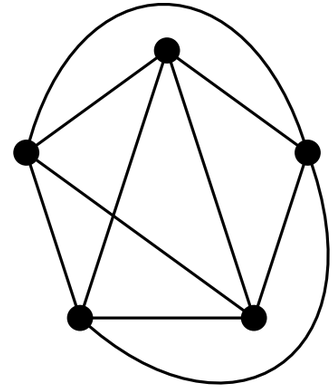


How many edges can a 1-planar graph have?

How many edges can a fan-planar graph have?

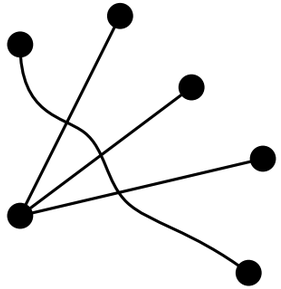


8. Crossing Number of 3-Plane Drawings



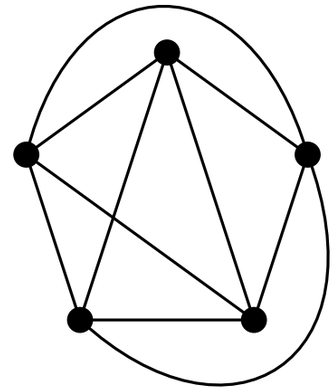
How many edges can a 1-planar graph have?

How many edges can a fan-planar graph have?



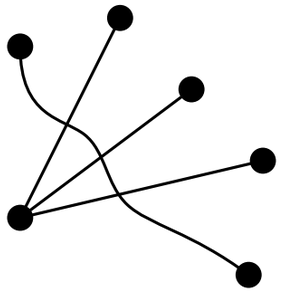
General: How many edges can . . . have?

8. Crossing Number of 3-Plane Drawings



How many edges can a 1-planar graph have?

How many edges can a fan-planar graph have?



General: How many edges can . . . have?

The density formula:

$$|E| = t(|V| - 2) - \sum_{c \in \mathcal{C}} \left(\frac{t-1}{4} \|c\| - t \right) - |\mathcal{X}|$$

9. Geometric Spanners of Bounded Tree-width

- Tree-width: Measure for how tree-like a graph is.

9. Geometric Spanners of Bounded Tree-width

- Tree-width: Measure for how tree-like a graph is.
  Trees have tree-width 1; the complete graph K_n has tree-width $n - 1$.

9. Geometric Spanners of Bounded Tree-width

- Tree-width: Measure for how tree-like a graph is.
  Trees have tree-width 1; the complete graph K_n has tree-width $n-1$.

- Given a set P of points in $\mathbb{R}^d$, a geometric t-spanner graph for P is a graph G with vertex set P and, for every two points u and v in P,
  $d_G(u,v) \leq t \cdot \|u-v\|$.

9. Geometric Spanners of Bounded Tree-width

- Tree-width: Measure for how tree-like a graph is.
  Trees have tree-width 1; the complete graph K_n has tree-width $n-1$.

- Given a set P of points in $\mathbb{R}^d$, a geometric t-spanner graph for P is a graph G with vertex set P and, for every two points u and v in P,
  $d_G(u,v) \leq t \cdot \|u-v\|$. The number t is called the *dilation* of G.

9. Geometric Spanners of Bounded Tree-width

- Tree-width: Measure for how tree-like a graph is.
  Trees have tree-width 1; the complete graph K_n has tree-width $n-1$.

- Given a set P of points in $\mathbb{R}^d$, a geometric t-spanner graph for P is a graph G with vertex set P and, for every two points u and v in P,
  $d_G(u,v) \leq t \cdot \|u-v\|$. The number t is called the *dilation* of G.

- The authors show how to compute, for any fixed dimension d, a t-spanner with tree-width k and $t \in O(n/k^{d(d-1)})$.

9. Geometric Spanners of Bounded Tree-width

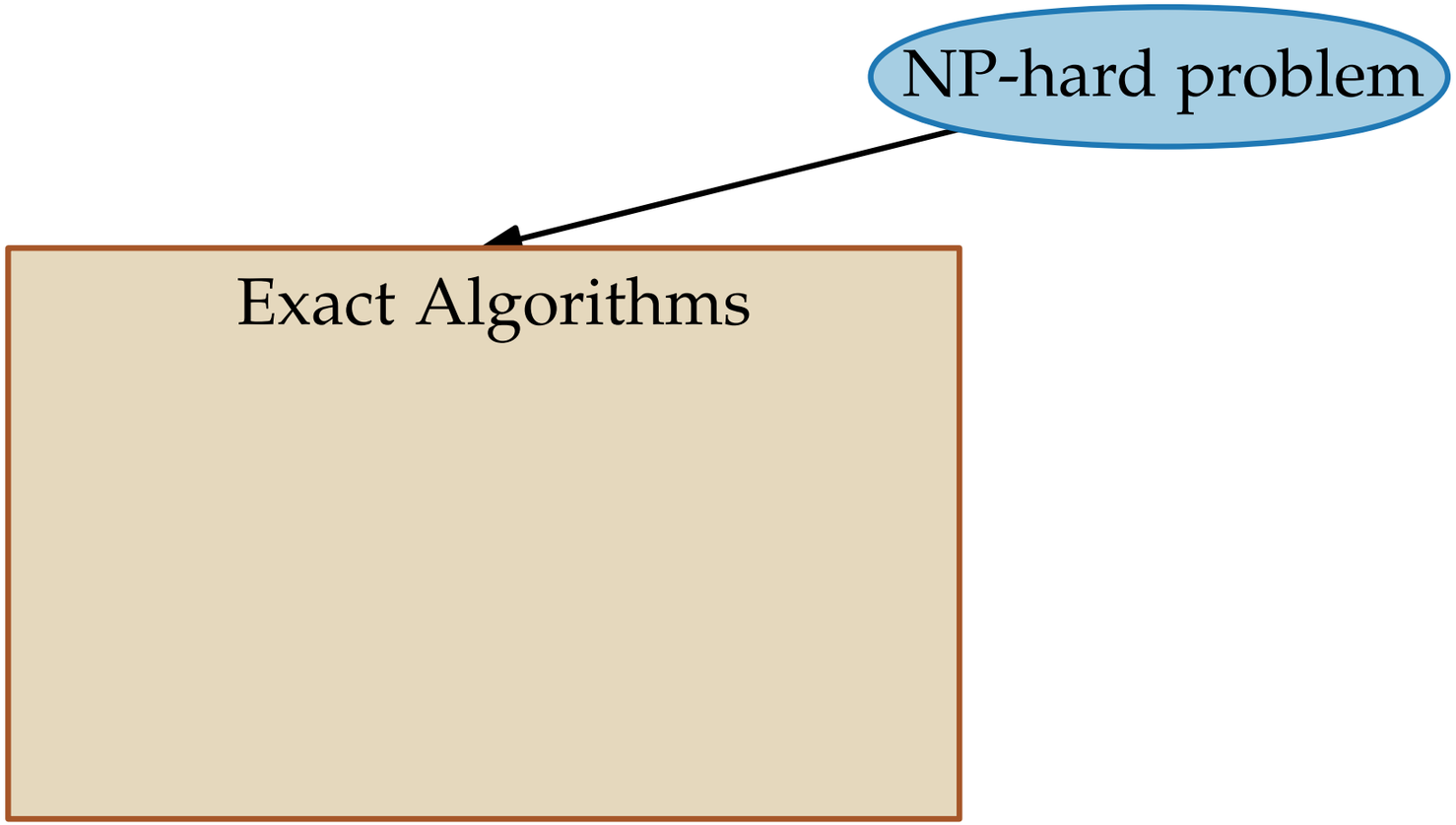
- Tree-width: Measure for how tree-like a graph is.
  Trees have tree-width 1; the complete graph K_n has tree-width $n-1$.

- Given a set P of points in $\mathbb{R}^d$, a geometric t-spanner graph for P is a graph G with vertex set P and, for every two points u and v in P,
  $d_G(u,v) \leq t \cdot \|u-v\|$. The number t is called the *dilation* of G.

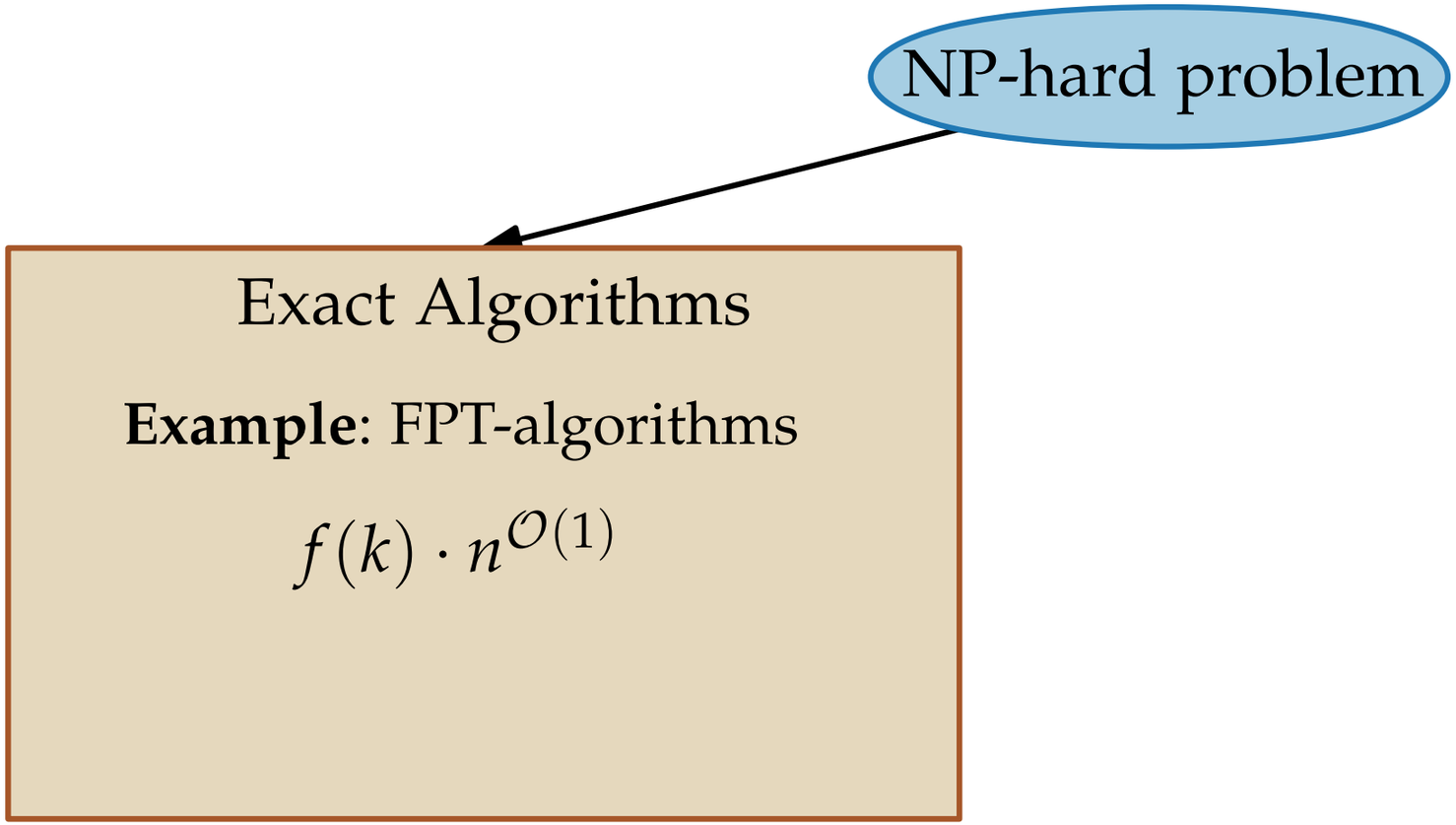
- The authors show how to compute, for any fixed dimension d, a t-spanner with tree-width k and $t \in O(n/k^{d(d-1)})$.
  The result is asymptotically worst-case optimal.

10. Efficient Parameterized Approximation

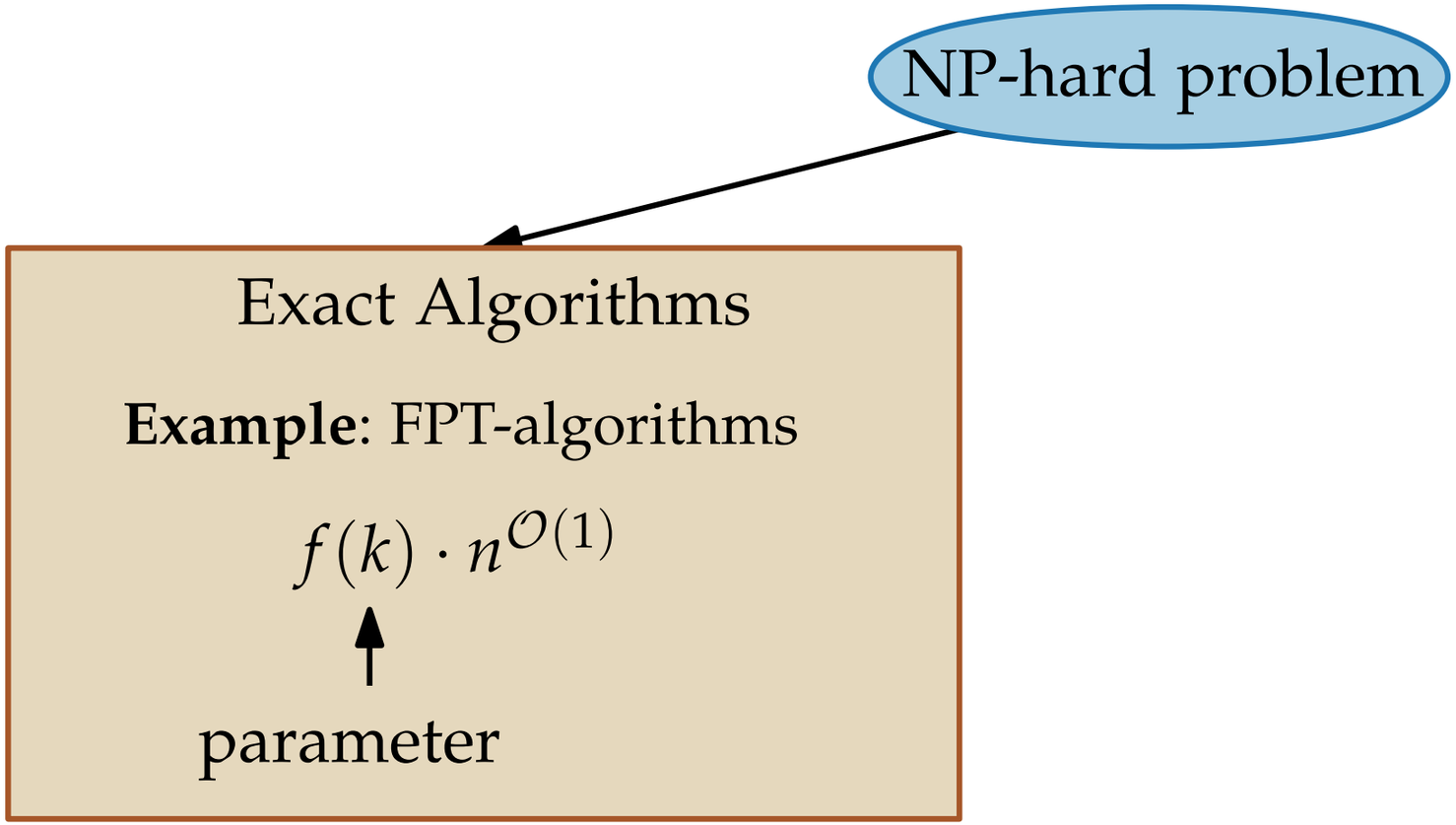
10. Efficient Parameterized Approximation



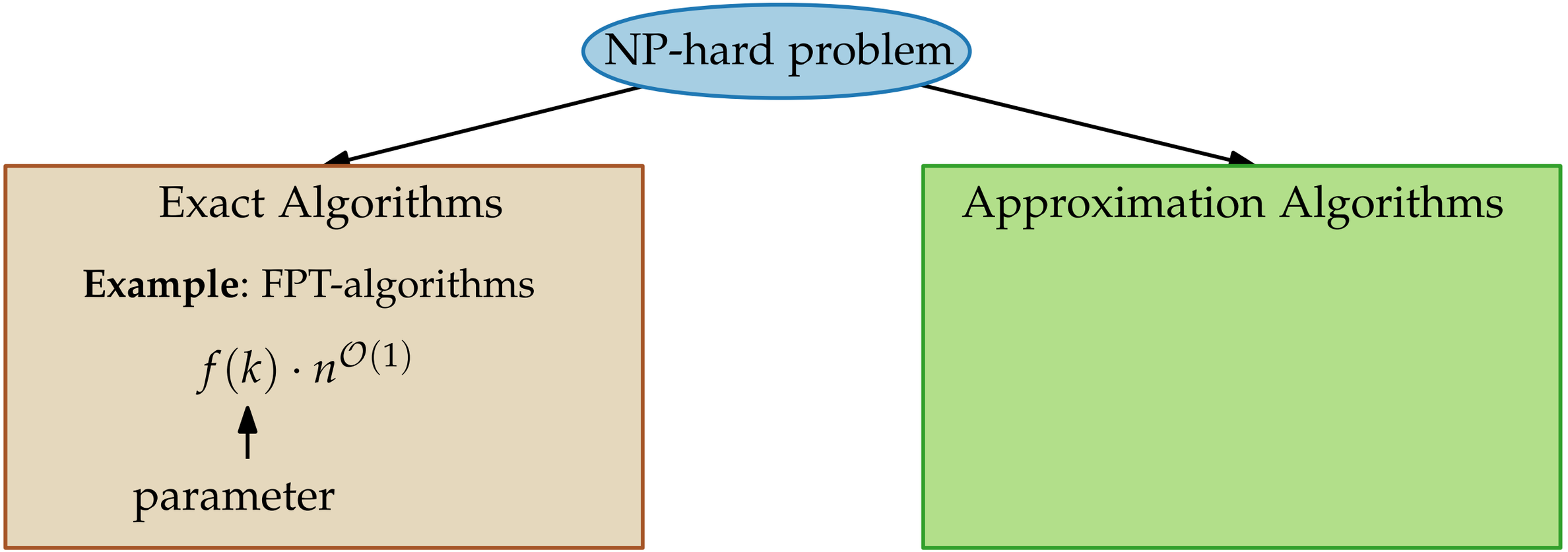
10. Efficient Parameterized Approximation



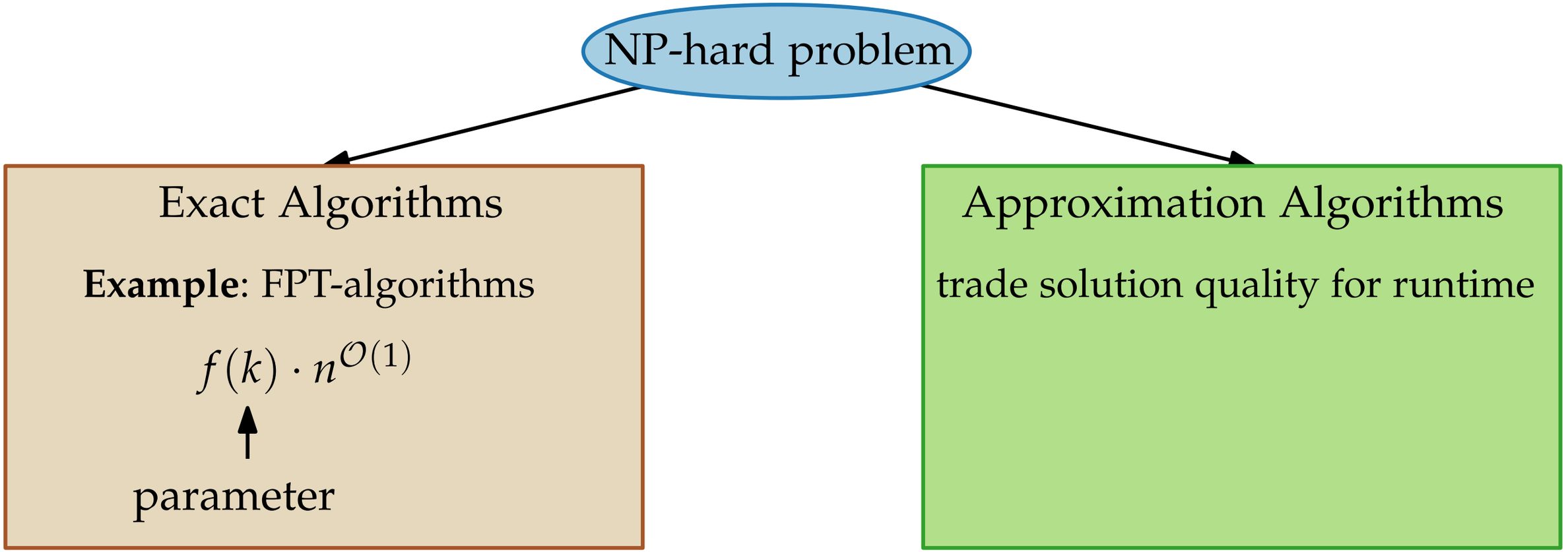
10. Efficient Parameterized Approximation



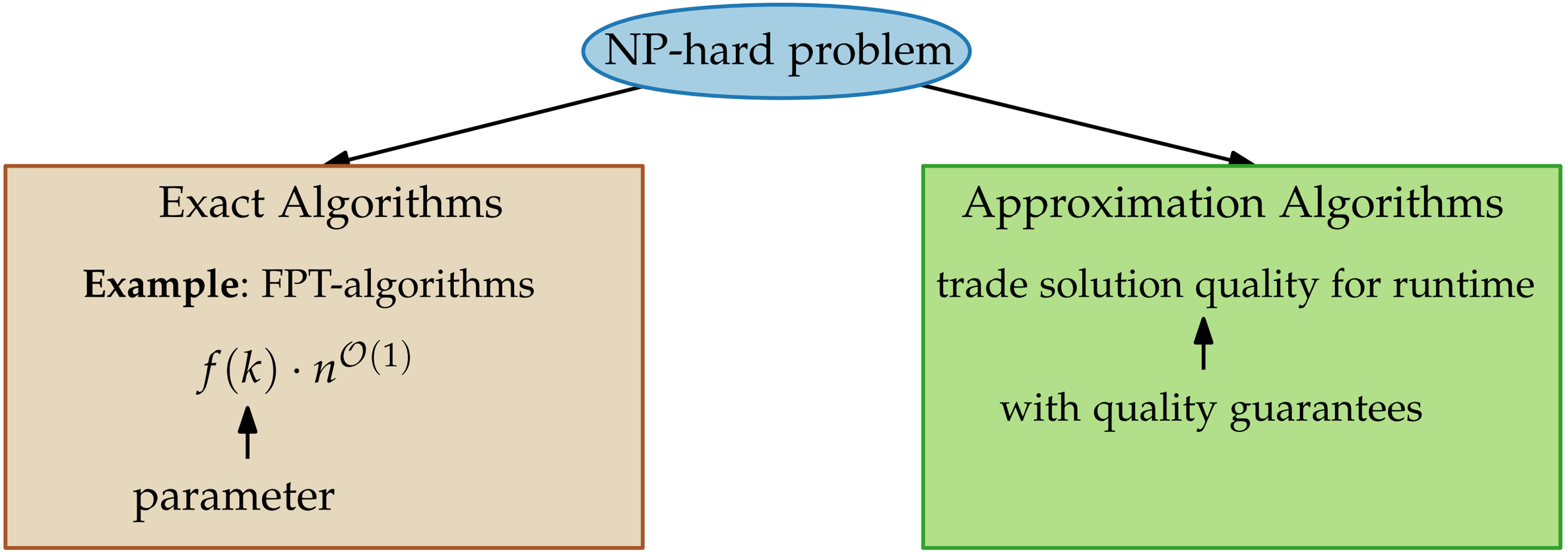
10. Efficient Parameterized Approximation



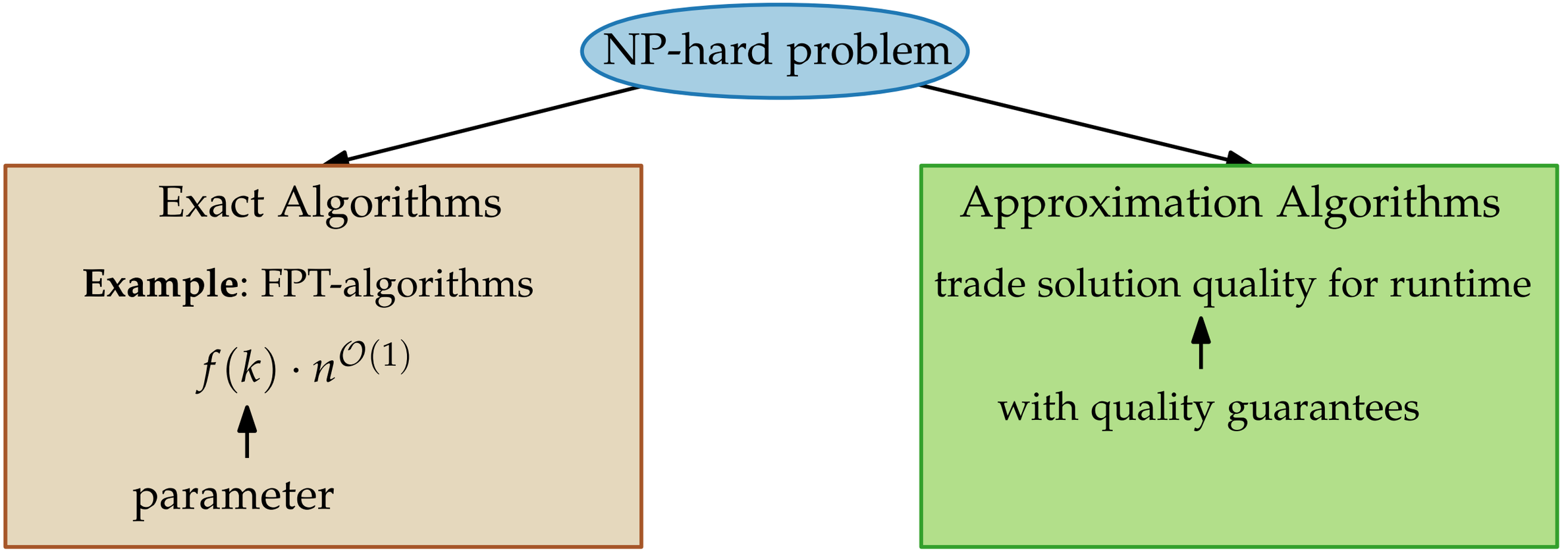
10. Efficient Parameterized Approximation



10. Efficient Parameterized Approximation



10. Efficient Parameterized Approximation



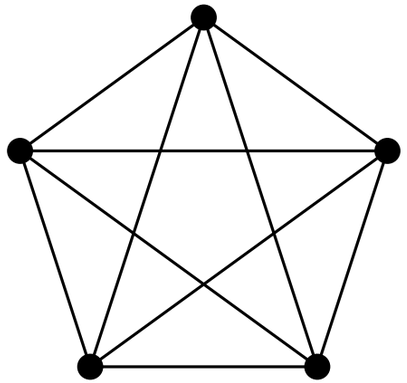
Here: Polynomial-time algorithms that produce a solution of value at most/least $c\mathrm{OPT} \pm f(k)$

11. Kuratowski's Theorem

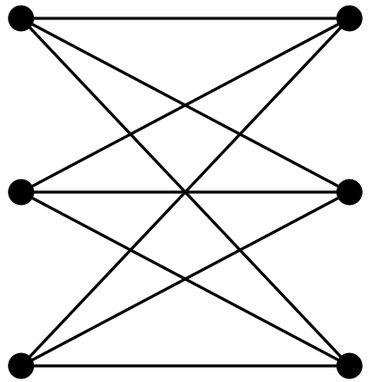
When is a graph planar?

11. Kuratowski's Theorem

When is a graph planar?



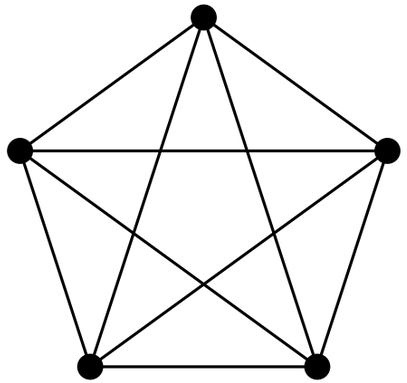
K_5



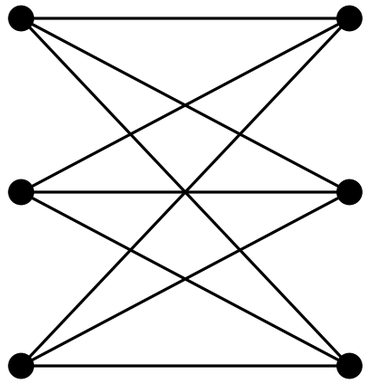
$K_{3,3}$

11. Kuratowski's Theorem

When is a graph planar?



K_5



$K_{3,3}$

Kuratowski's Theorem:
A graph is planar if and only if it does not contain K_5 or $K_{3,3}$ as a minor.

Themenverteilung

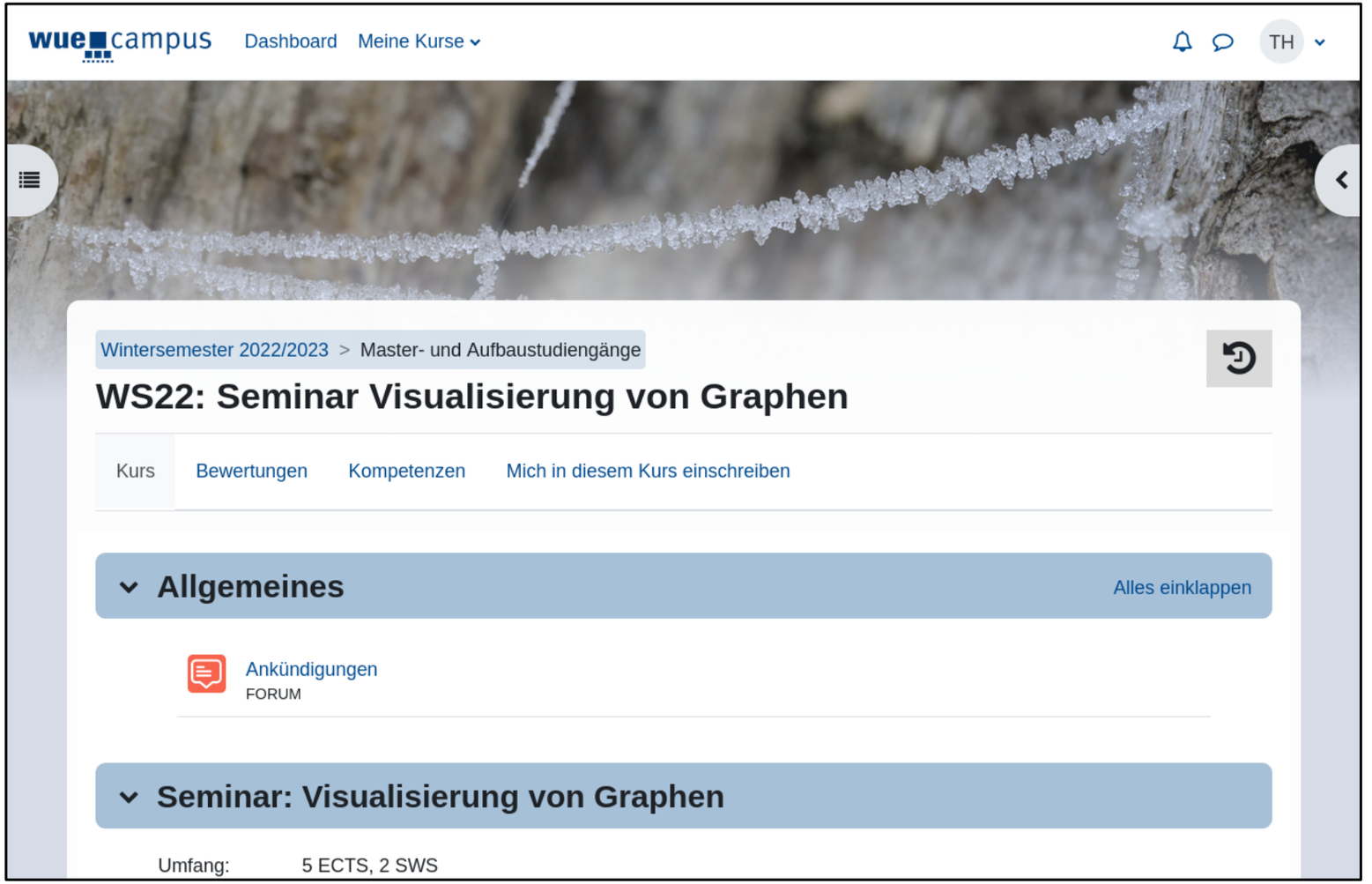
1. Parameterized Complexity of Simultaneous Planarity
2. Lower bounds based on the Exponential-Time Hypothesis
3. Constraint Logic: A Uniform Framework for Modeling Computation as Games
4. XALP-completeness of Parameterized Problems on Planar Graphs
5. Approximation Algorithms for NP-complete Problems on Planar Graphs
6. How to Morph Planar Graph Drawings
7. Sliding Squares in Parallel
8. Crossing Number of 3-Plane Drawings
9. Geometric Spanners of Bounded Tree-width
10. Efficient Parameterized Approximation
11. Kuratowski's Theorem

Nächste Schritte

- In WueCampus anmelden

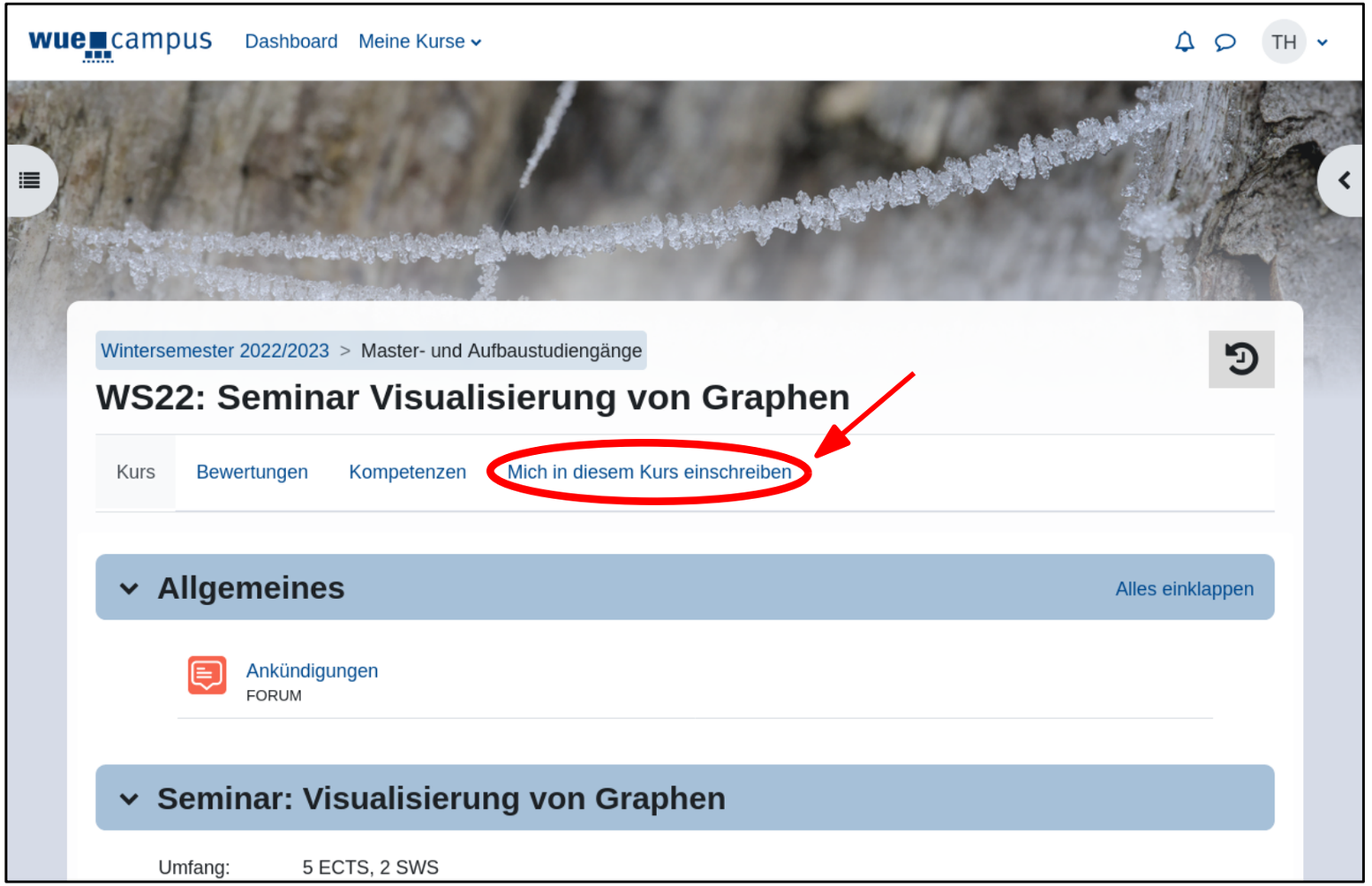
Nächste Schritte

- In WueCampus anmelden



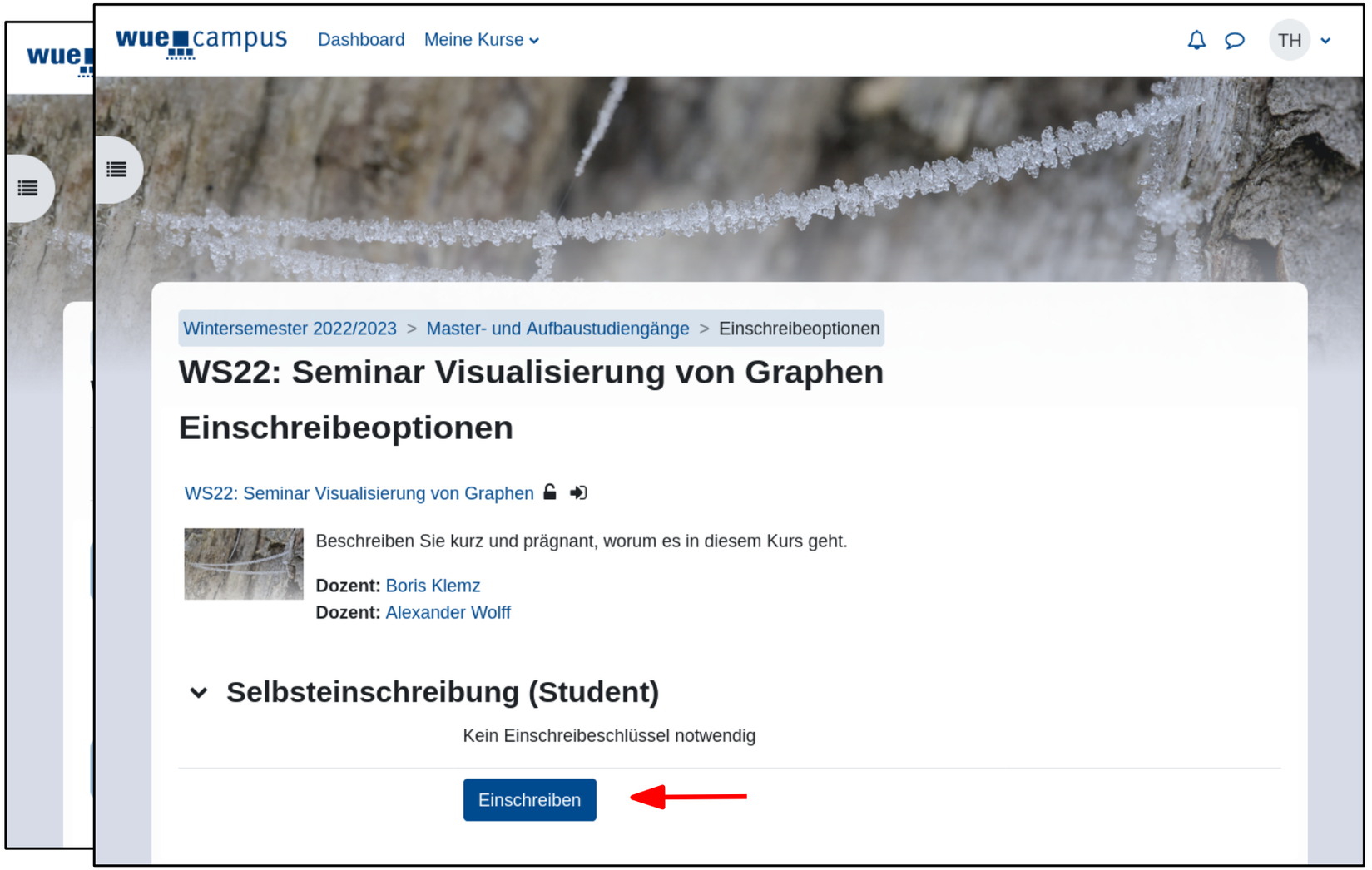
Nächste Schritte

- In WueCampus anmelden



Nächste Schritte

- In WueCampus anmelden



Nächste Schritte

- In WueCampus anmelden

Nächste Schritte

- In WueCampus anmelden
- In WueStudy anmelden

Nächste Schritte

- In WueCampus anmelden
- In WueStudy anmelden
- Überblick verschaffen und Kurzvortrag vorbereiten

Nächste Schritte

- In WueCampus anmelden
- In WueStudy anmelden
- Überblick verschaffen und Kurzvortrag vorbereiten
- Bei Fragen (oder *spätestens drei Wochen vor dem eigenen Vortrag*) an die BetreuerIn wenden

Nächste Schritte

Bei allgemeinen Fragen kann gerne das **Diskussionsforum** im WueCampus genutzt werden!

- In WueCampus anmelden
- In WueStudy anmelden
- Überblick verschaffen und Kurzvortrag vorbereiten
- Bei Fragen (oder *spätestens drei Wochen vor dem eigenen Vortrag*) an die BetreuerIn wenden

Nächste Schritte

Bei allgemeinen Fragen kann gerne das **Diskussionsforum** im WueCampus genutzt werden!

- In WueCampus anmelden
- In WueStudy anmelden
- Überblick verschaffen und Kurzvortrag vorbereiten
- Bei Fragen (oder *spätestens drei Wochen vor dem eigenen Vortrag*) an die BetreuerIn wenden

Zum Abschluss:

Demonstration des Programms IPE
zum Erstellen von Abbildungen und Folien

http://ipe.otfried.org/

Nächste Schritte

Bei allgemeinen Fragen kann gerne das **Diskussionsforum** im WueCampus genutzt werden!

- In WueCampus anmelden
- In WueStudy anmelden
- Überblick verschaffen und Kurzvortrag vorbereiten
- Bei Fragen (oder *spätestens drei Wochen vor dem eigenen Vortrag*) an die BetreuerIn wenden

Zum Abschluss: Demonstration des Programms IPE zum Erstellen von Abbildungen und Folien

http://ipe.otfried.org/

Übrigens: ein gemeinsames git-Verzeichnis eignet sich hervorragend zum gemeinsamen Bearbeiten von .tex, aber auch .ipe Dateien!

Nächste Schritte

Bei allgemeinen Fragen kann gerne das **Diskussionsforum** im WueCampus genutzt werden!

- In WueCampus anmelden
- In WueStudy anmelden
- Überblick verschaffen und Kurzvortrag vorbereiten
- Bei Fragen (oder *spätestens drei Wochen vor dem eigenen Vortrag*) an die BetreuerIn wenden

Zum Abschluss: Demonstration des Programms IPE zum Erstellen von Abbildungen und Folien

http://ipe.otfried.org/

https://gitlab2.informatik.uni-wuerzburg.de/

Übrigens: ein gemeinsames git-Verzeichnis eignet sich hervorragend zum gemeinsamen Bearbeiten von .tex, aber auch .ipe Dateien!