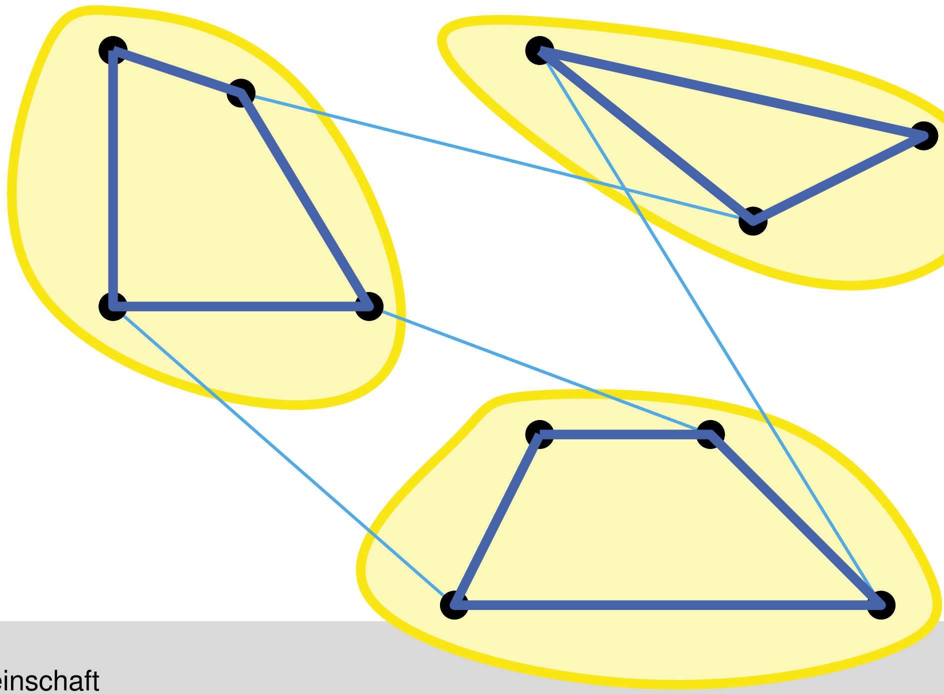
Karlsruhe Institute of Technology

Travelling Salesman Problem für Kantengewichte 1 und 2

Seminar NP-schwere Probleme

Sina Schmitt · **28. Mai 2018**

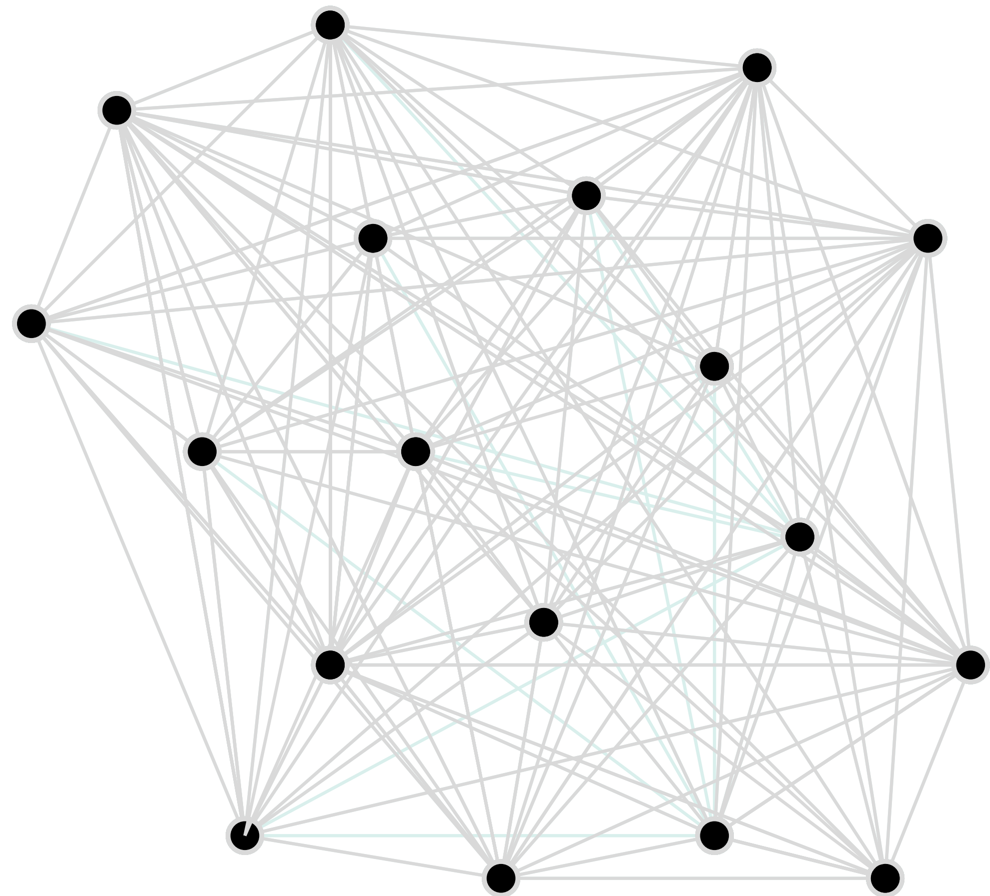
INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP



TRAVELLING SALESMAN PROBLEM

- **Gegeben:**

  Vollständiger, ungerichteter Graph mit Kantengewichten



Institute of Theoretical Informatics
Algorithmics Group

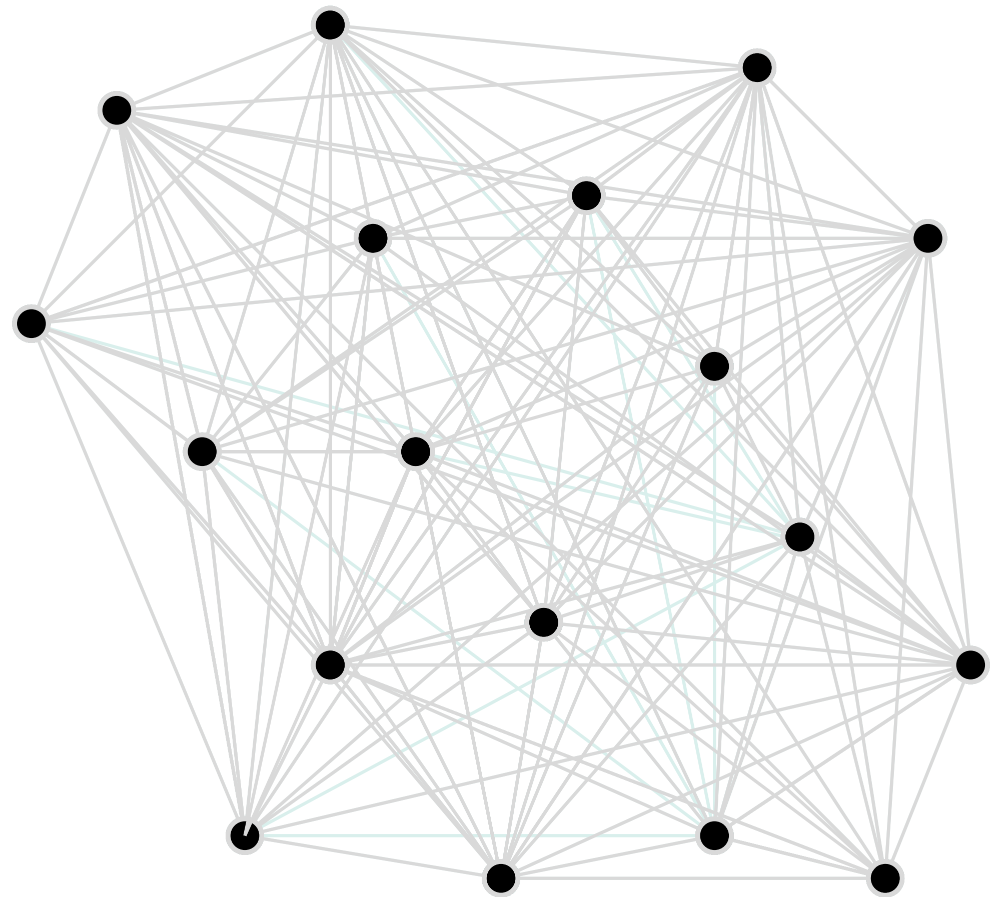
TRAVELLING SALESMAN PROBLEM

- **Gegeben:**

  Vollständiger, ungerichteter Graph mit Kantengewichten

- **Gesucht:**

  Hamiltonkreis, für den die Summe der Kantengewichte minimal ist



Institute of Theoretical Informatics
Algorithmics Group

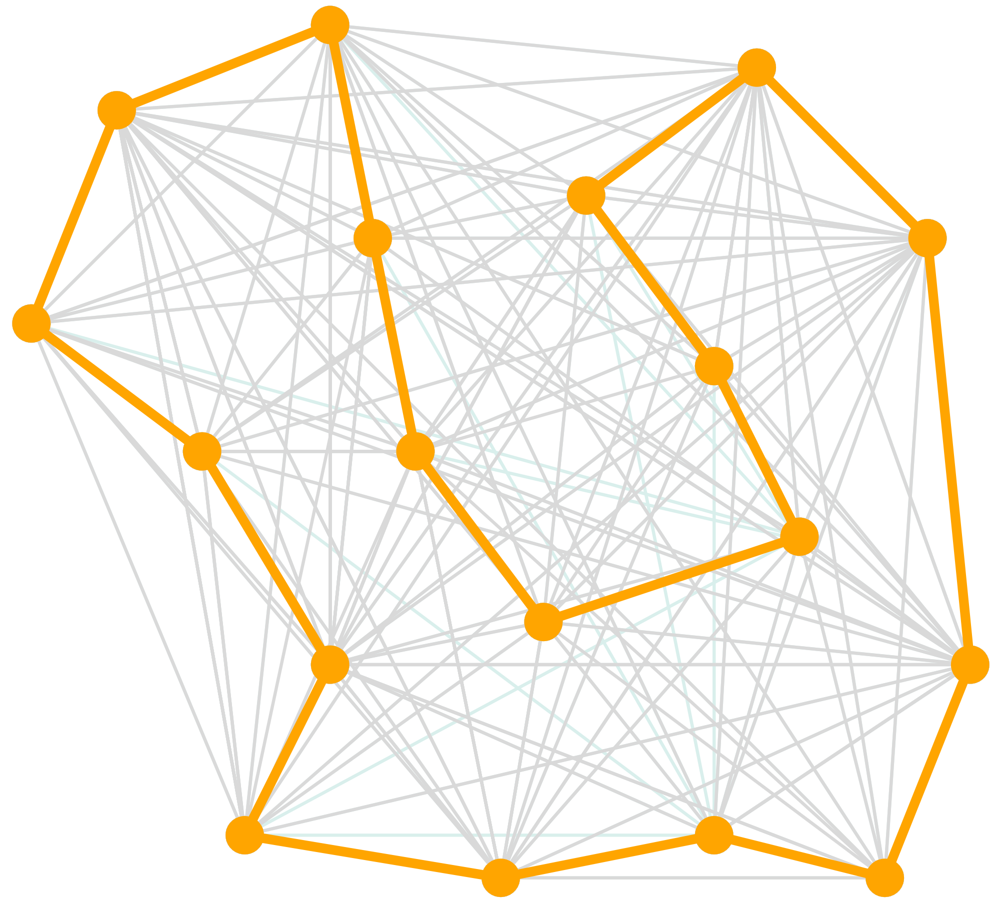
TRAVELLING SALESMAN PROBLEM

- **Gegeben:**

  Vollständiger, ungerichteter Graph mit Kantengewichten

- **Gesucht:**

  Hamiltonkreis, für den die Summe der Kantengewichte minimal ist



Institute of Theoretical Informatics
Algorithmics Group

TRAVELLING SALESMAN PROBLEM

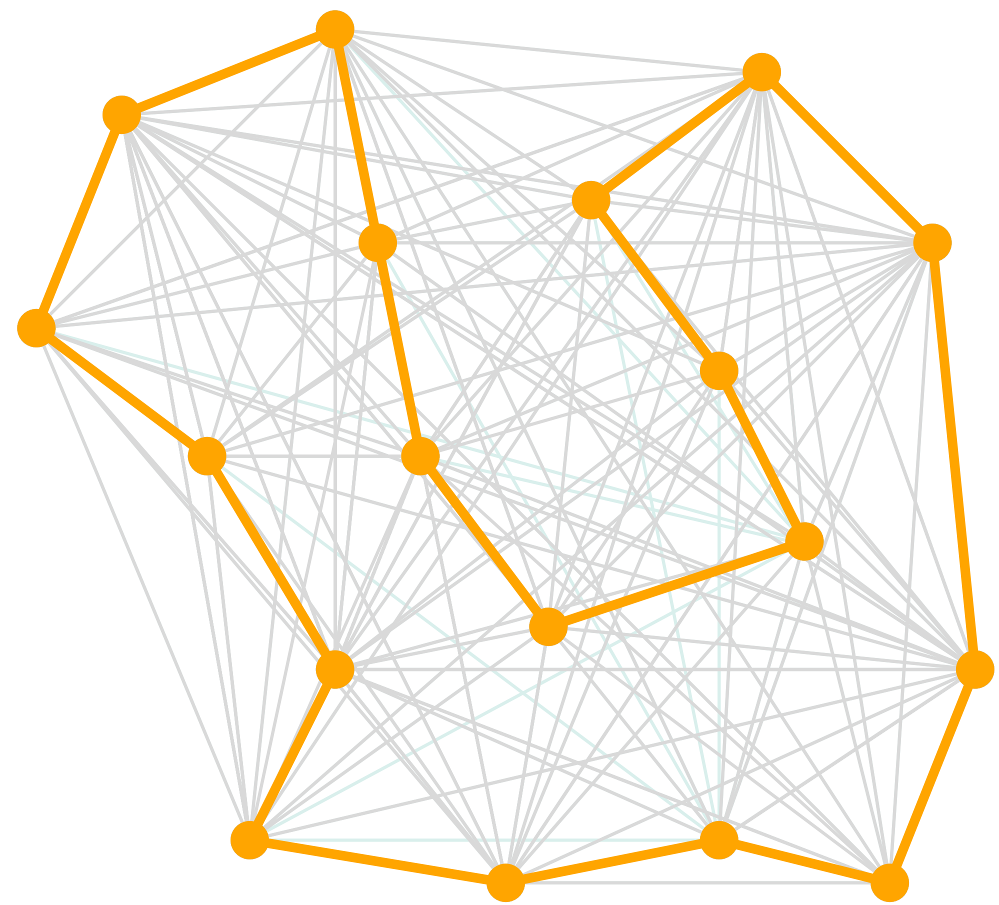
- **Gegeben:**

  Vollständiger, ungerichteter Graph mit Kantengewichten

- **Gesucht:**

  Hamiltonkreis, für den die Summe der Kantengewichte minimal ist

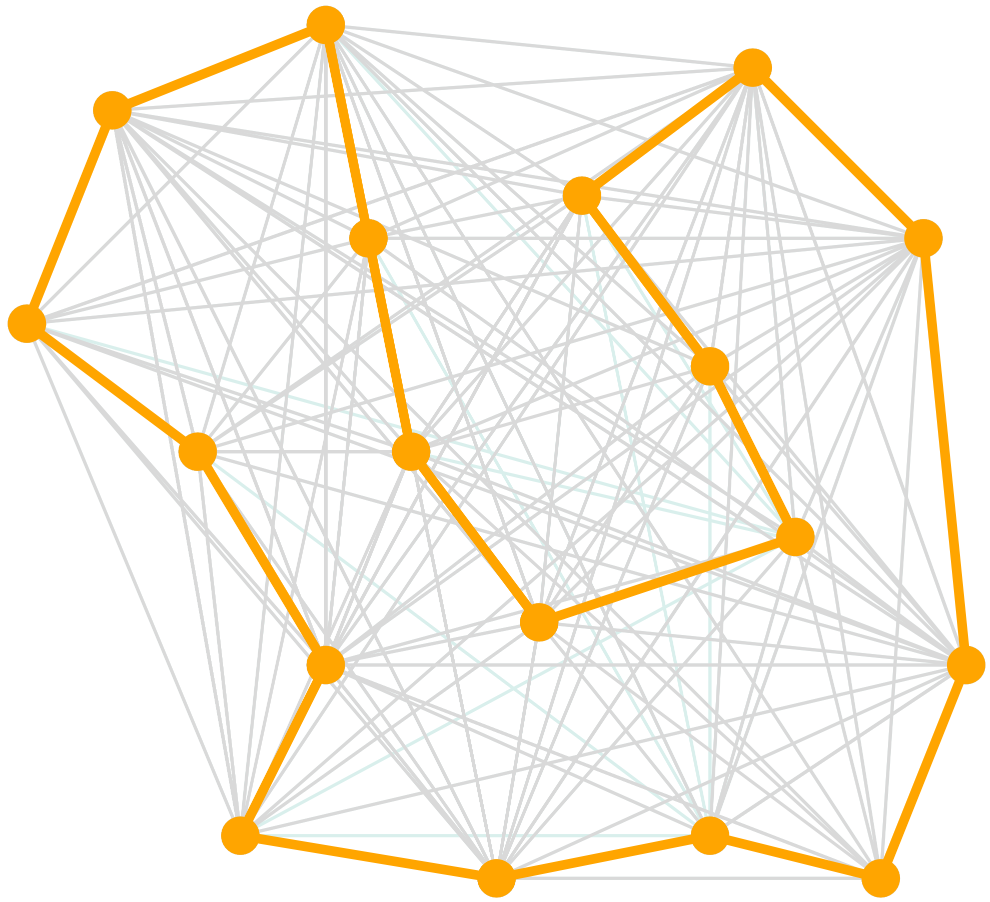
→ **NP-vollständig!**



Institute of Theoretical Informatics
Algorithmics Group

TRAVELLING SALESMAN PROBLEM

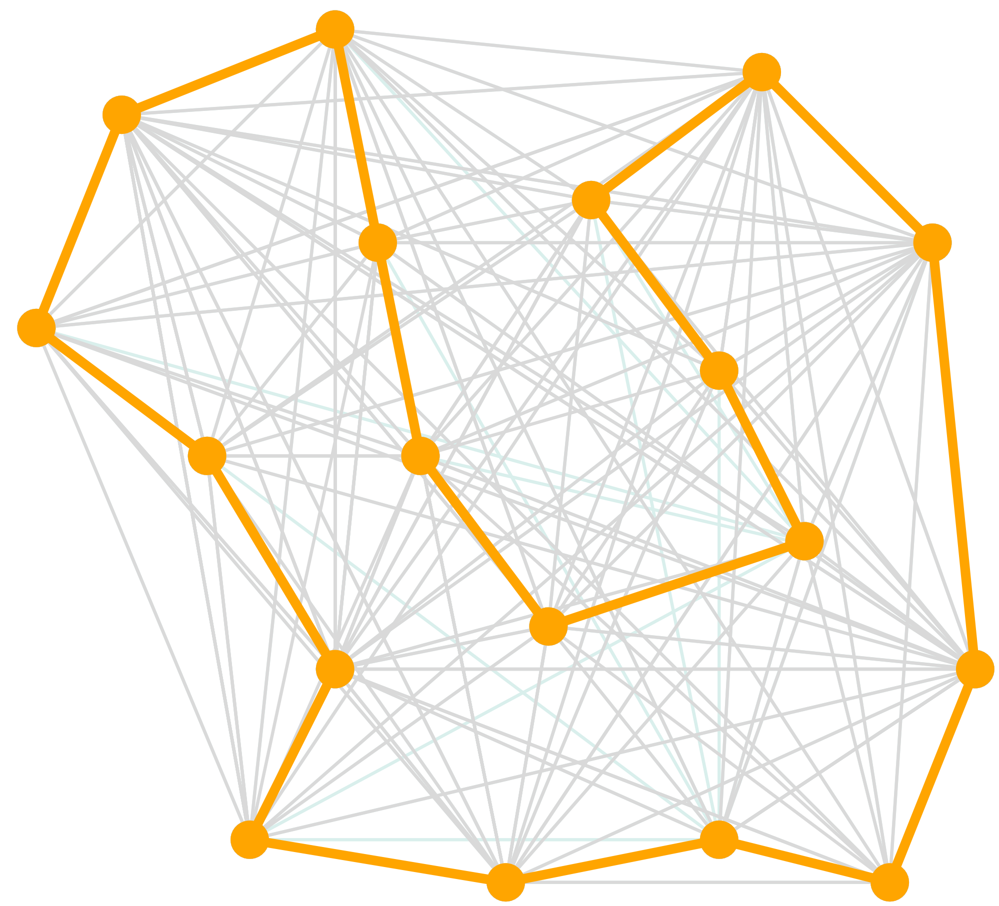
Approximationsalgorithmus



TRAVELLING SALESMAN PROBLEM

Approximationsalgorithmus

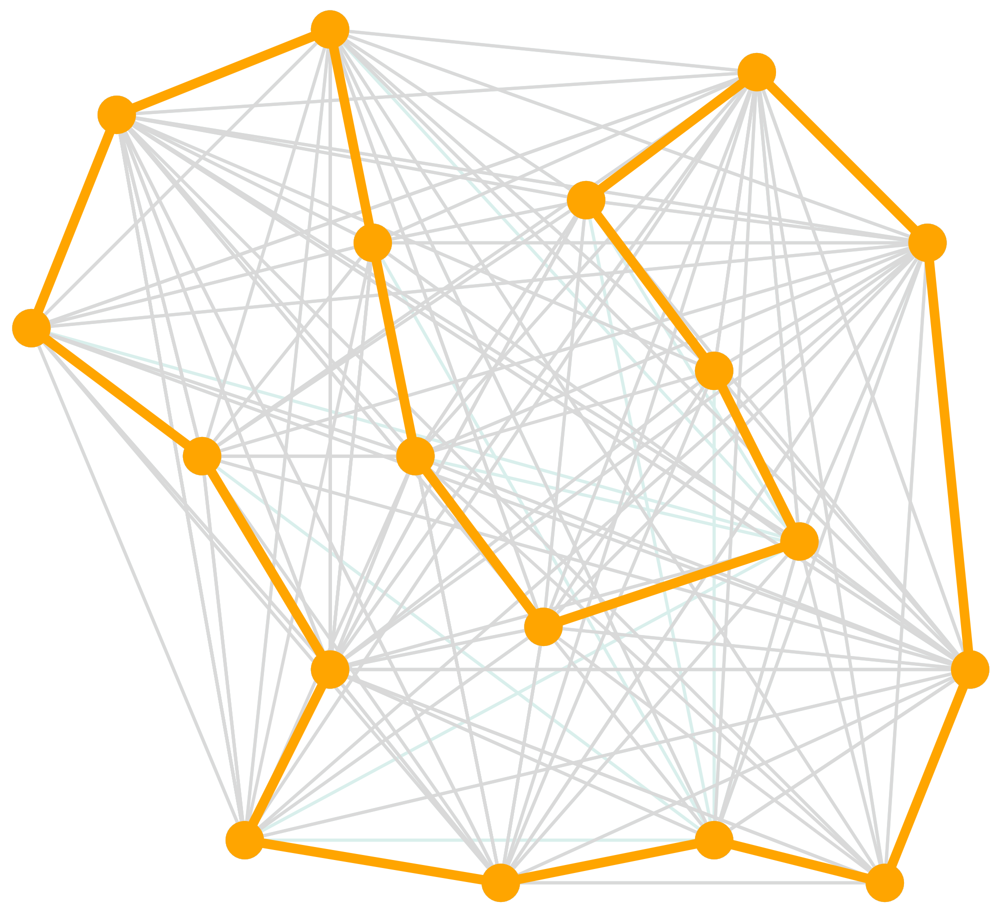
→ Einschränkung auf Kantengewichte 1 und 2



TRAVELLING SALESMAN PROBLEM

Approximationsalgorithmus

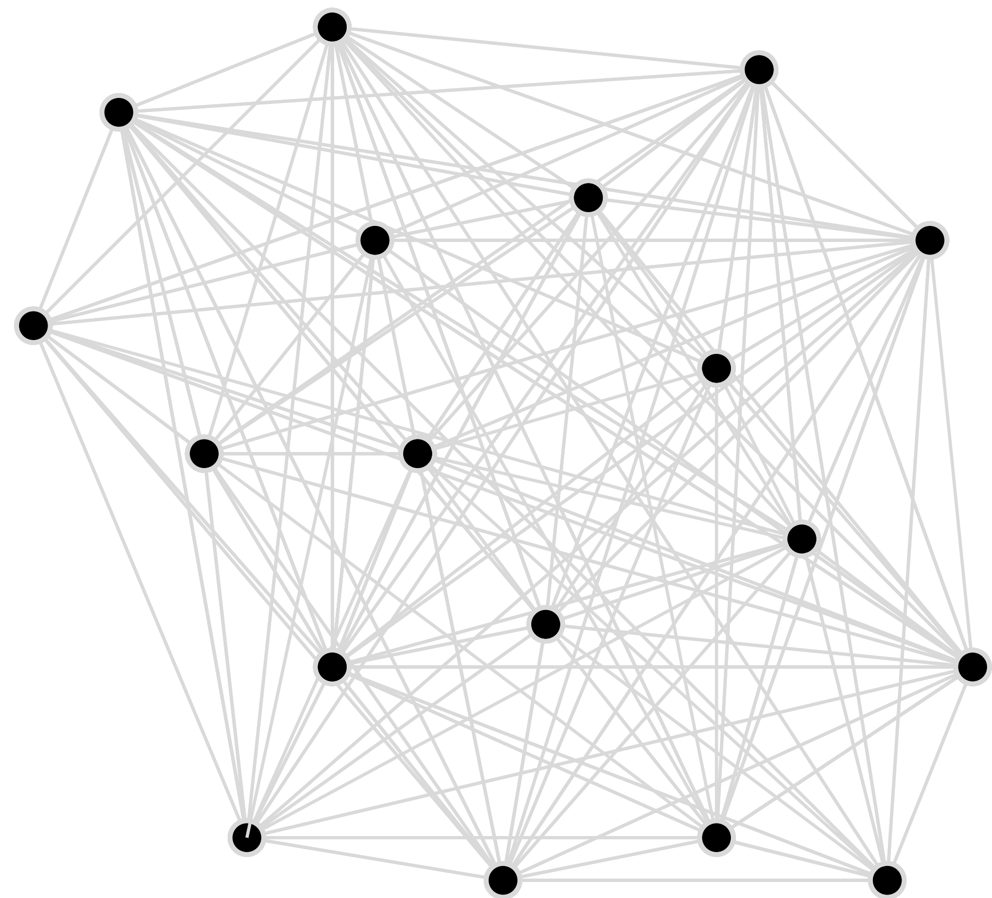
- Einschränkung auf Kantengewichte 1 und 2
- Methode: ¨Subtour Patching¨



Institute of Theoretical Informatics
Algorithmics Group

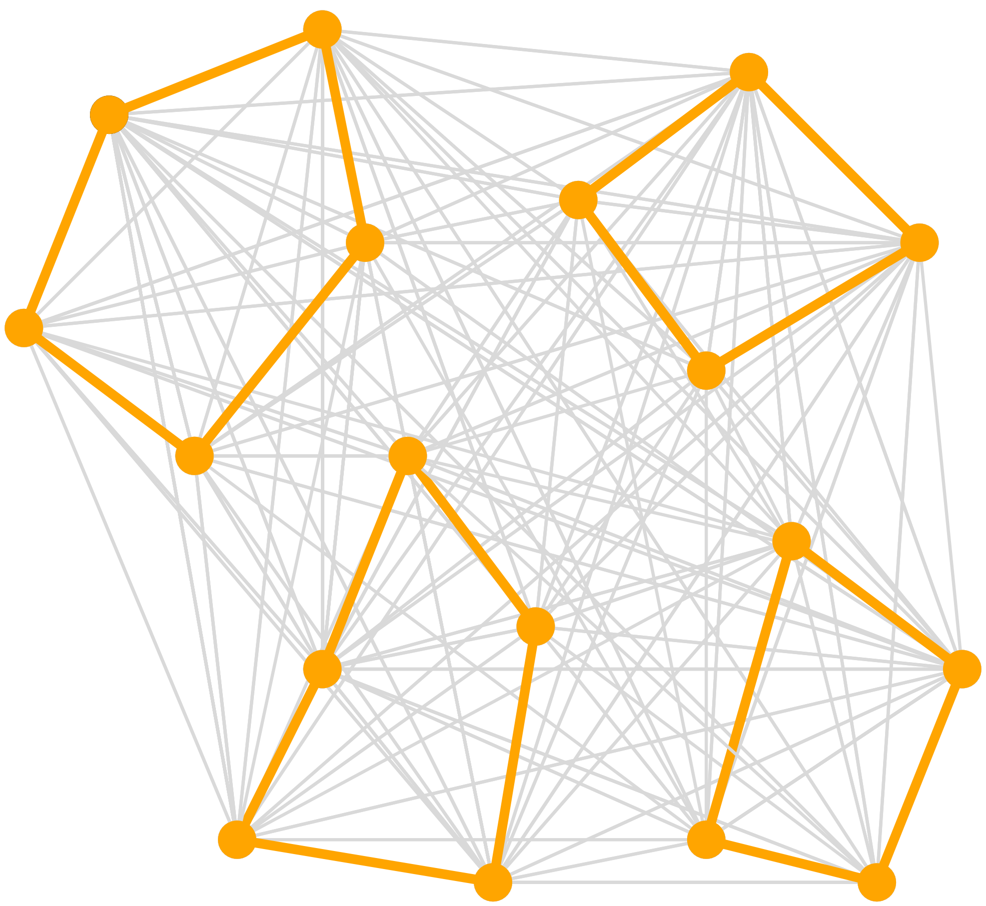
"SUBTOUR PATCHING"

- Start:
  Optimales 2-Matching



"SUBTOUR PATCHING"

- Start:
  Optimales 2-Matching

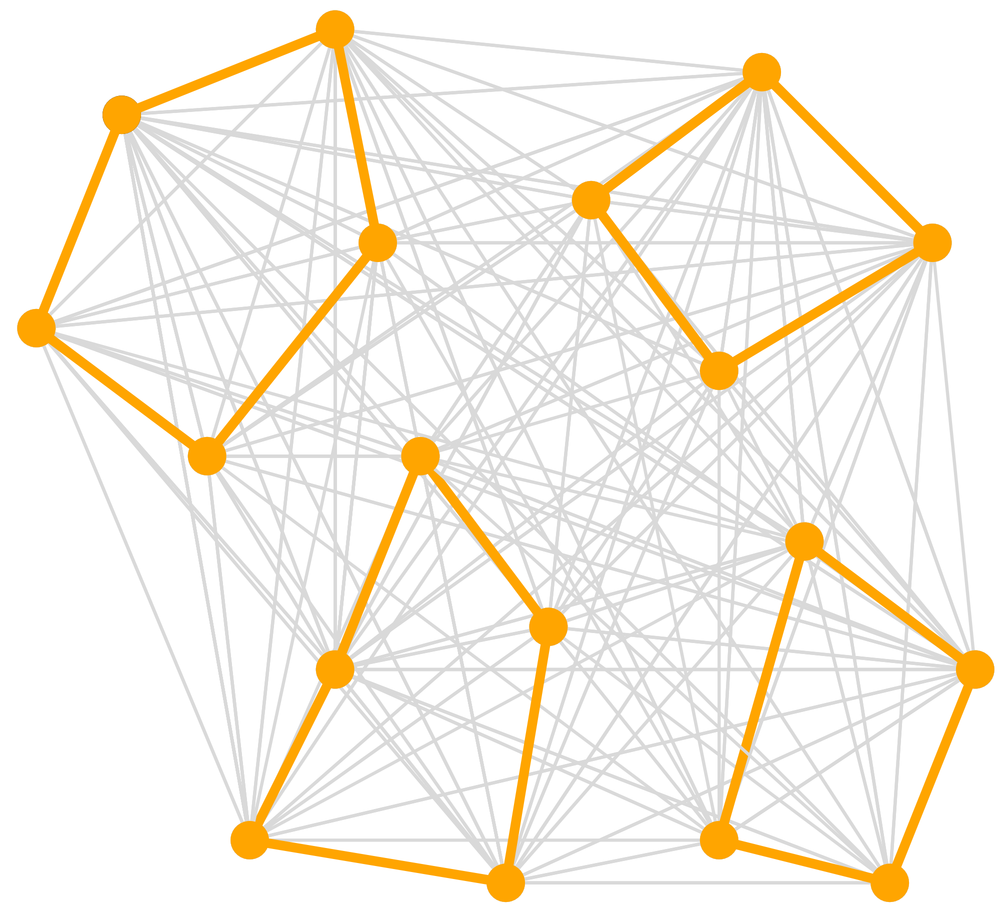


"SUBTOUR PATCHING"

Karlsruhe Institute of Technology

- Start:
  Optimales 2-Matching

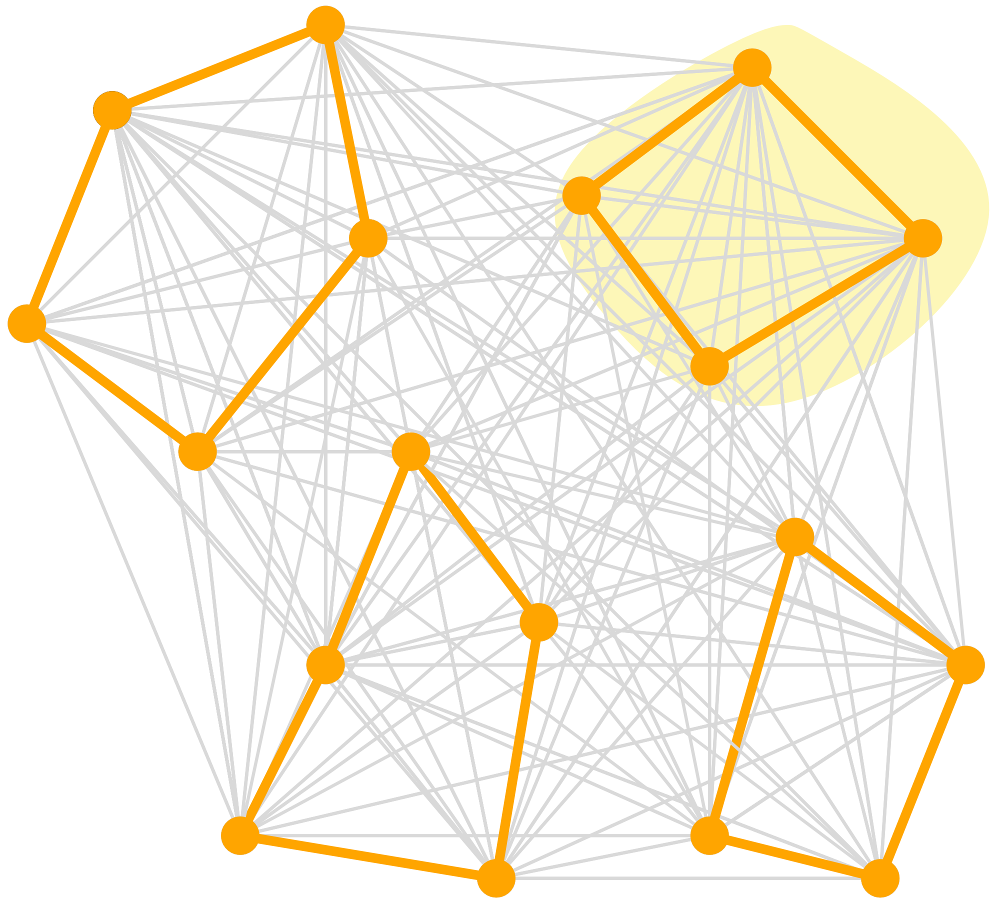
- Zwei Kreise wählen



Institute of Theoretical Informatics
Algorithmics Group

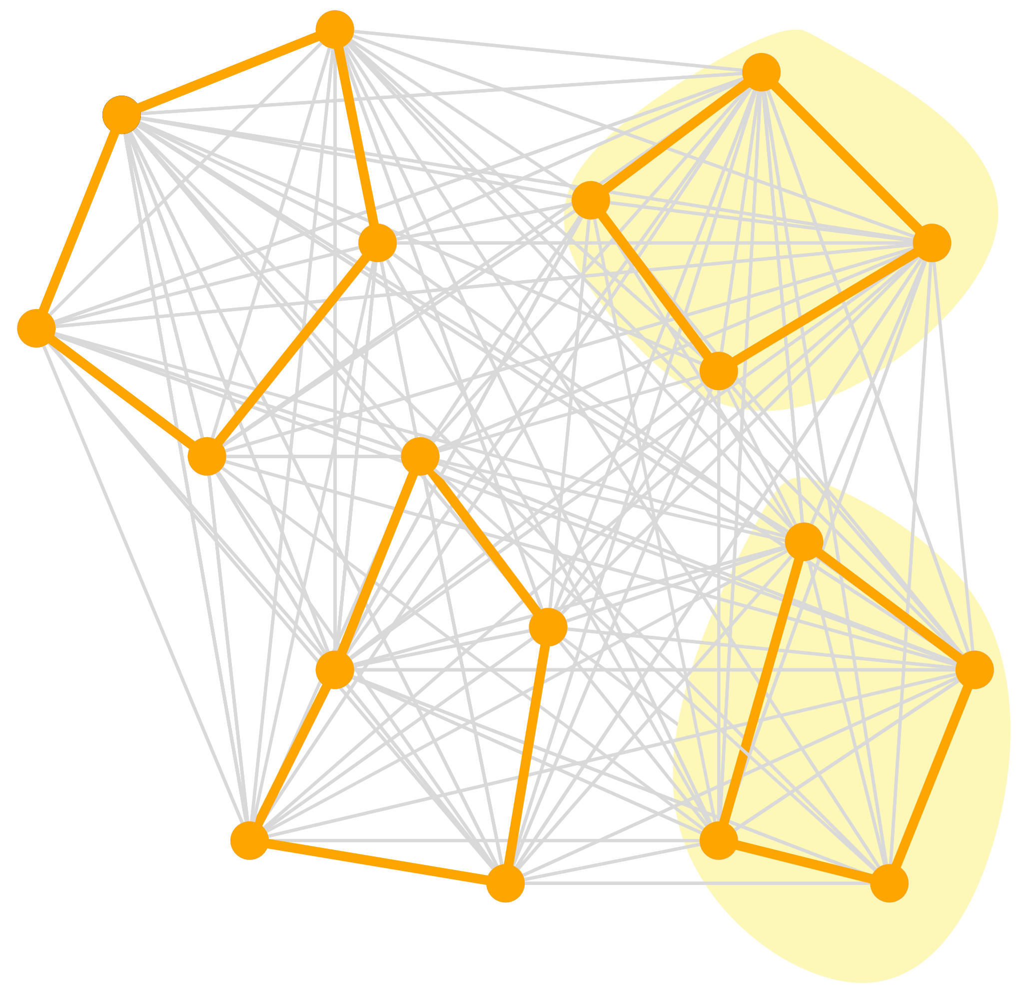
"SUBTOUR PATCHING"

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen



"SUBTOUR PATCHING"

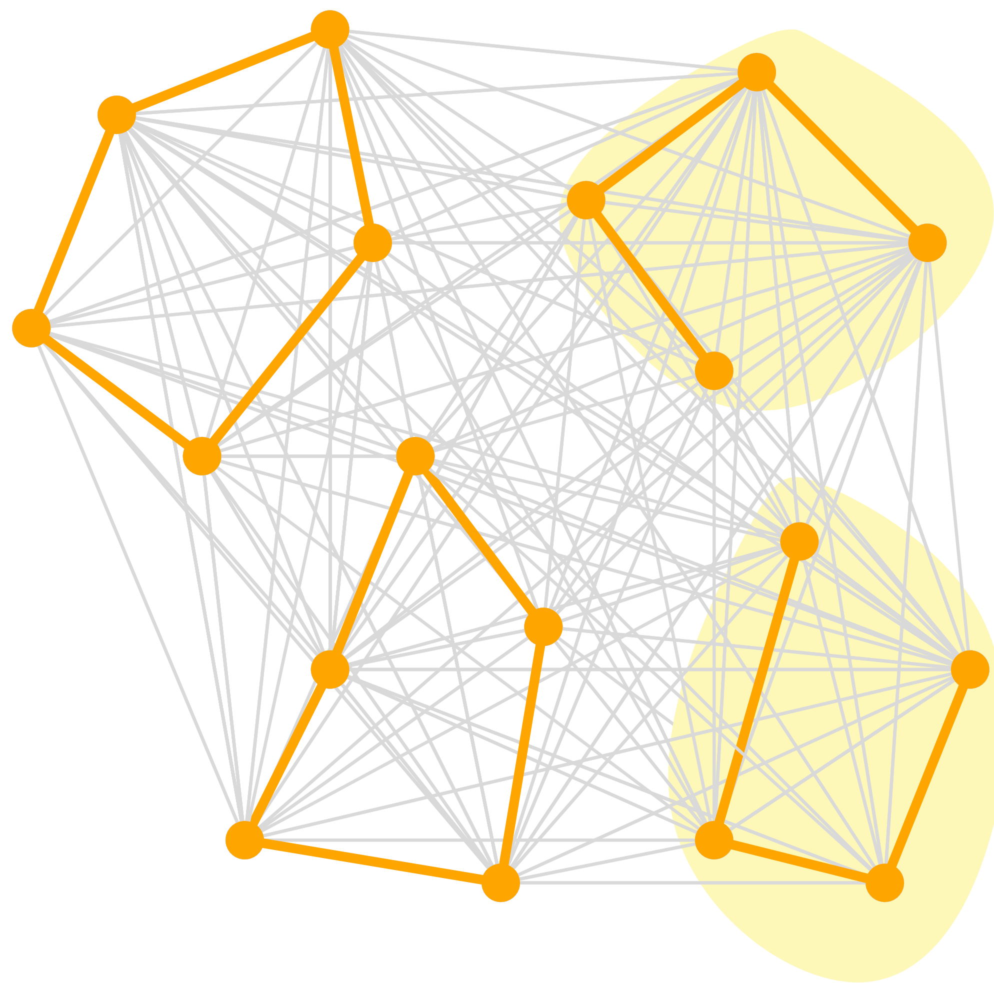
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen



Institute of Theoretical Informatics
Algorithmics Group

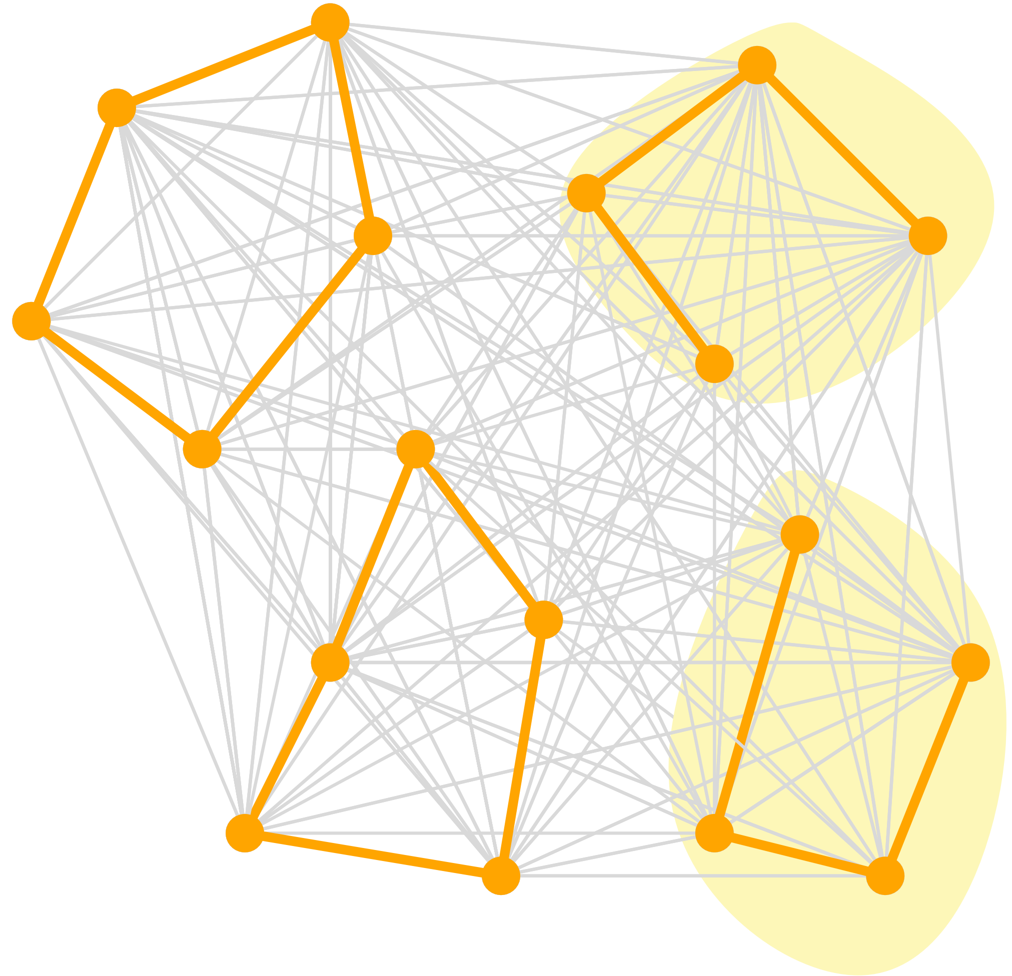
"SUBTOUR PATCHING"

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen



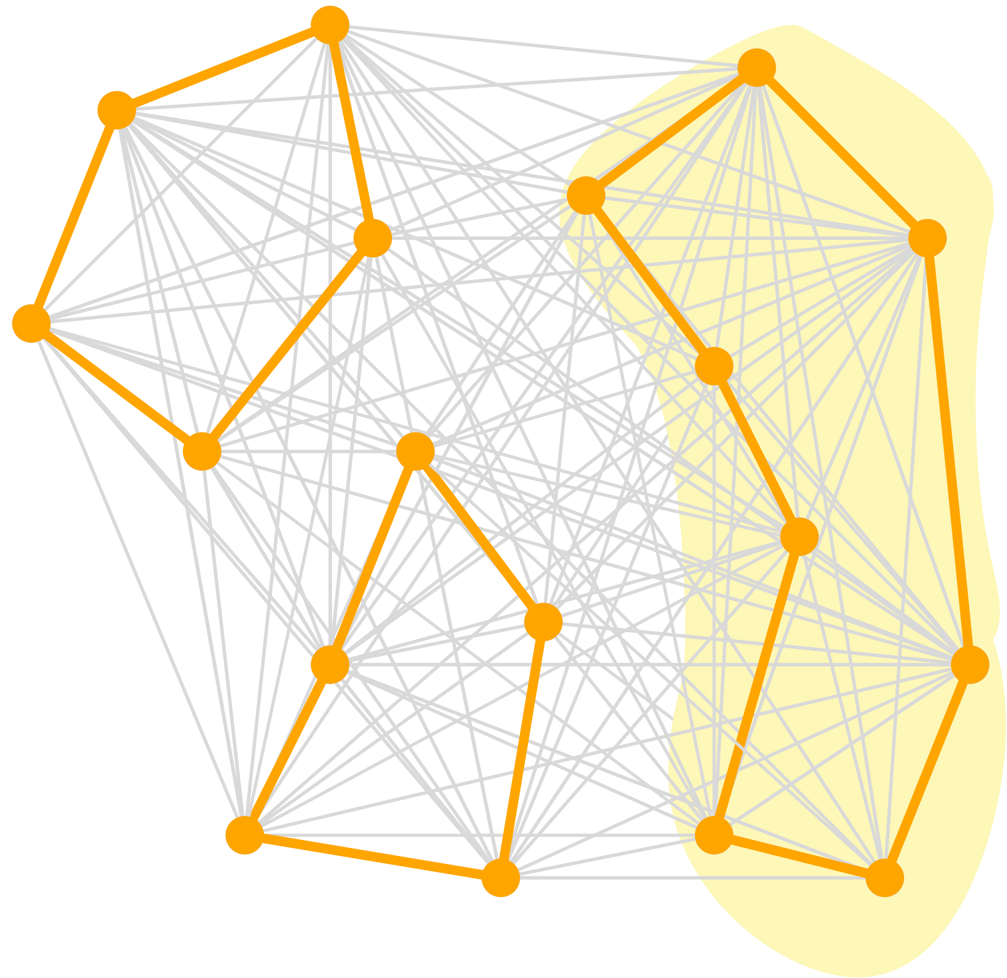
"SUBTOUR PATCHING"

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



"SUBTOUR PATCHING"

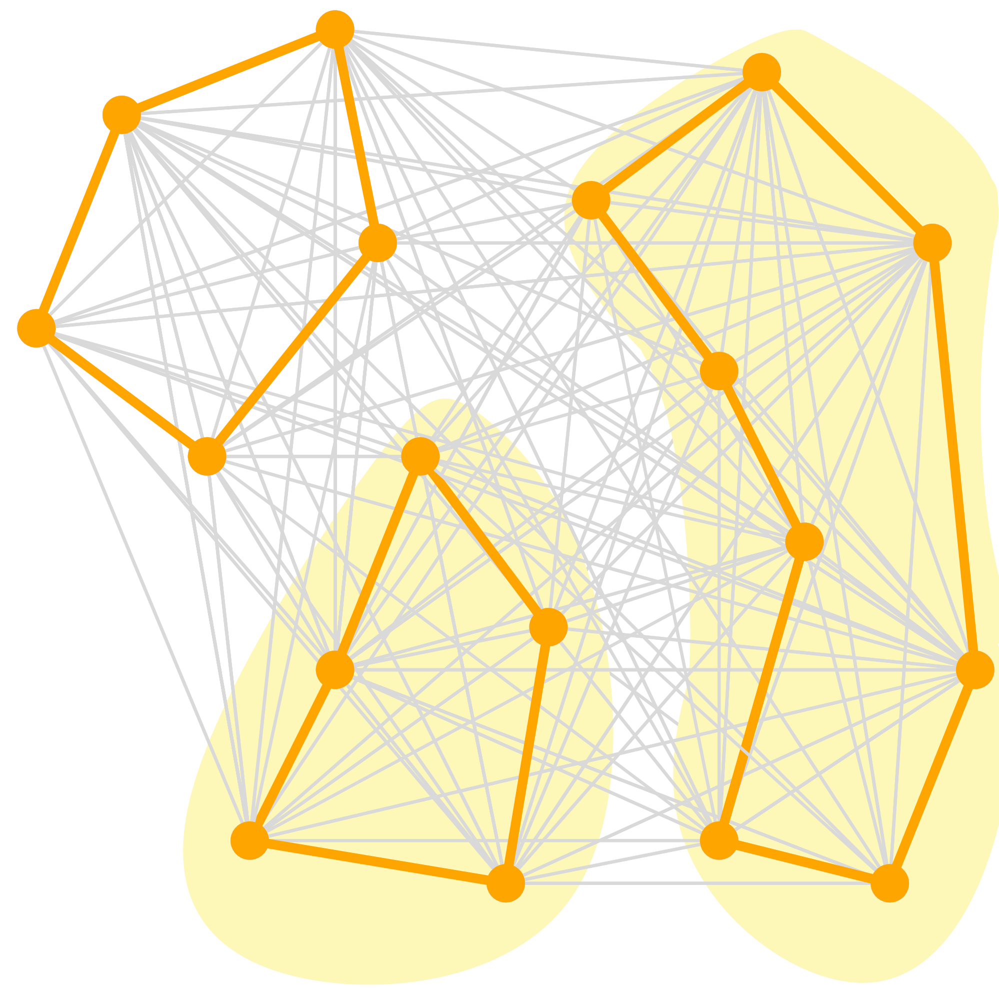
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



Institute of Theoretical Informatics
Algorithmics Group

"SUBTOUR PATCHING"

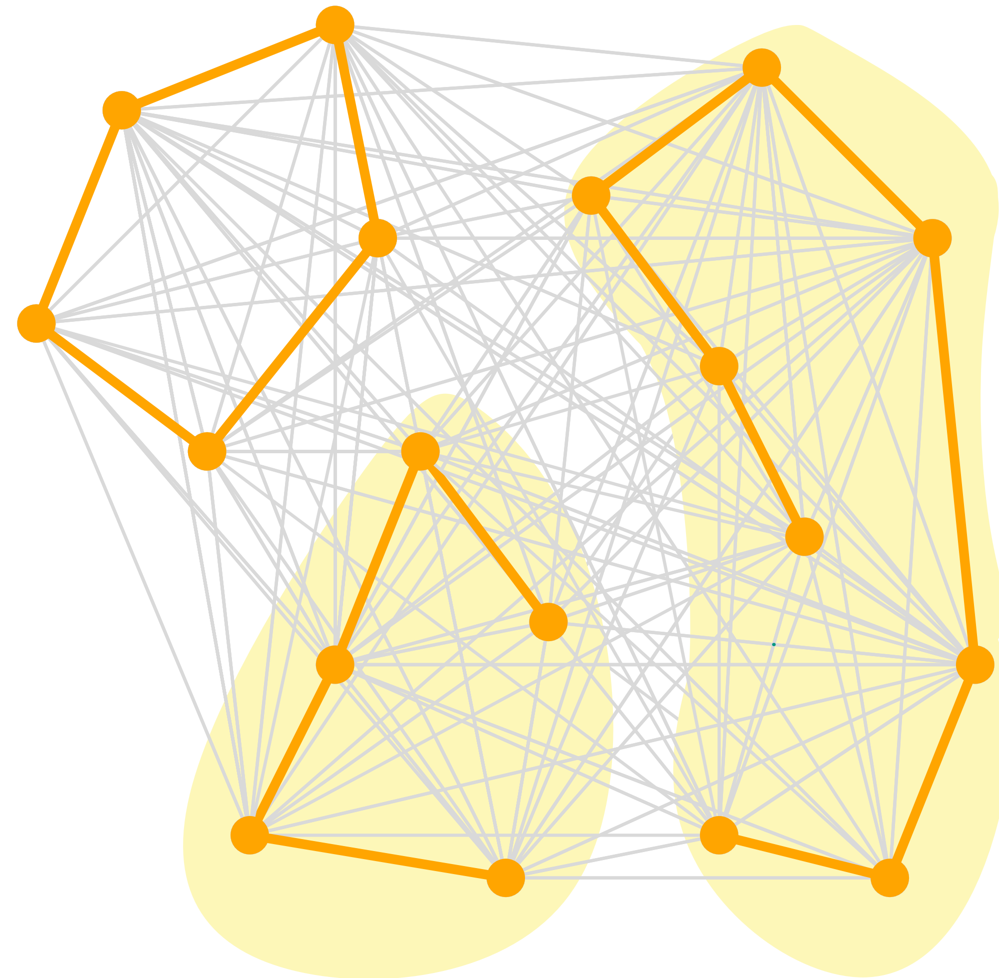
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



Institute of Theoretical Informatics
Algorithmics Group

"SUBTOUR PATCHING"

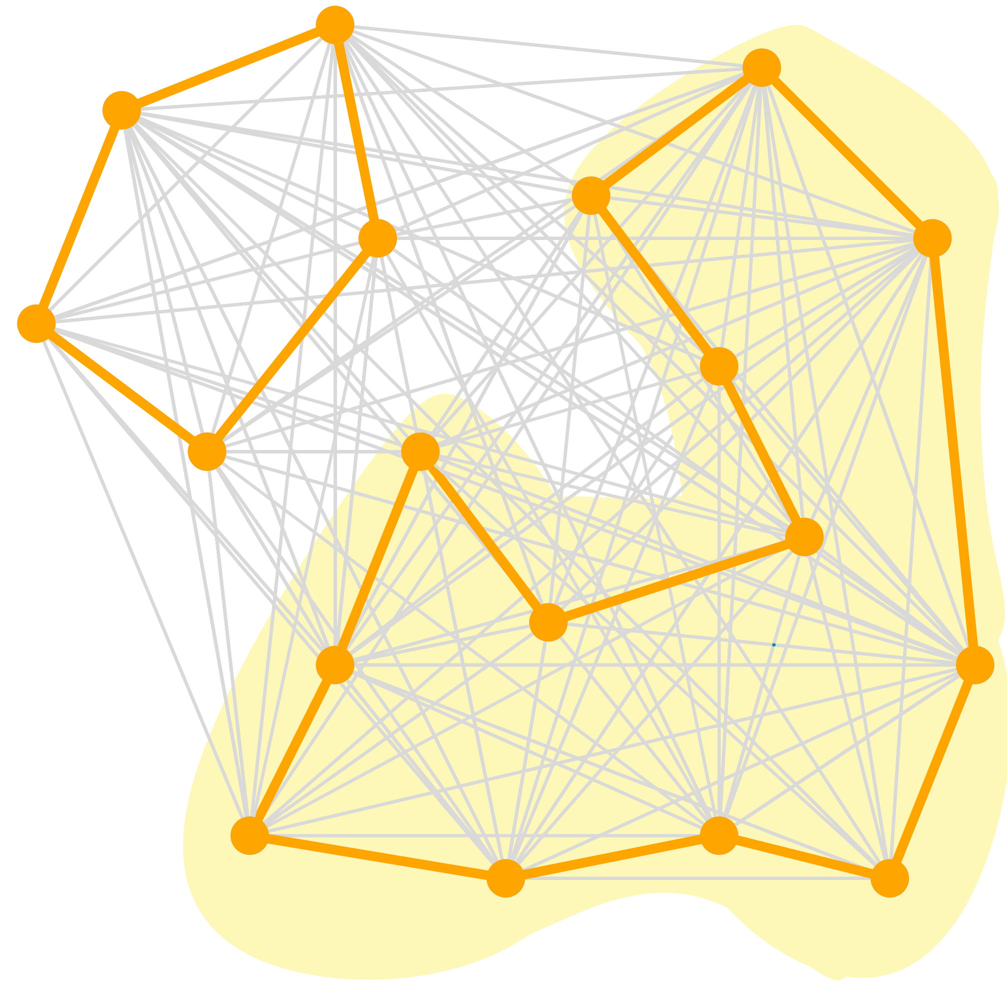
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



Institute of Theoretical Informatics
Algorithmics Group

"SUBTOUR PATCHING"

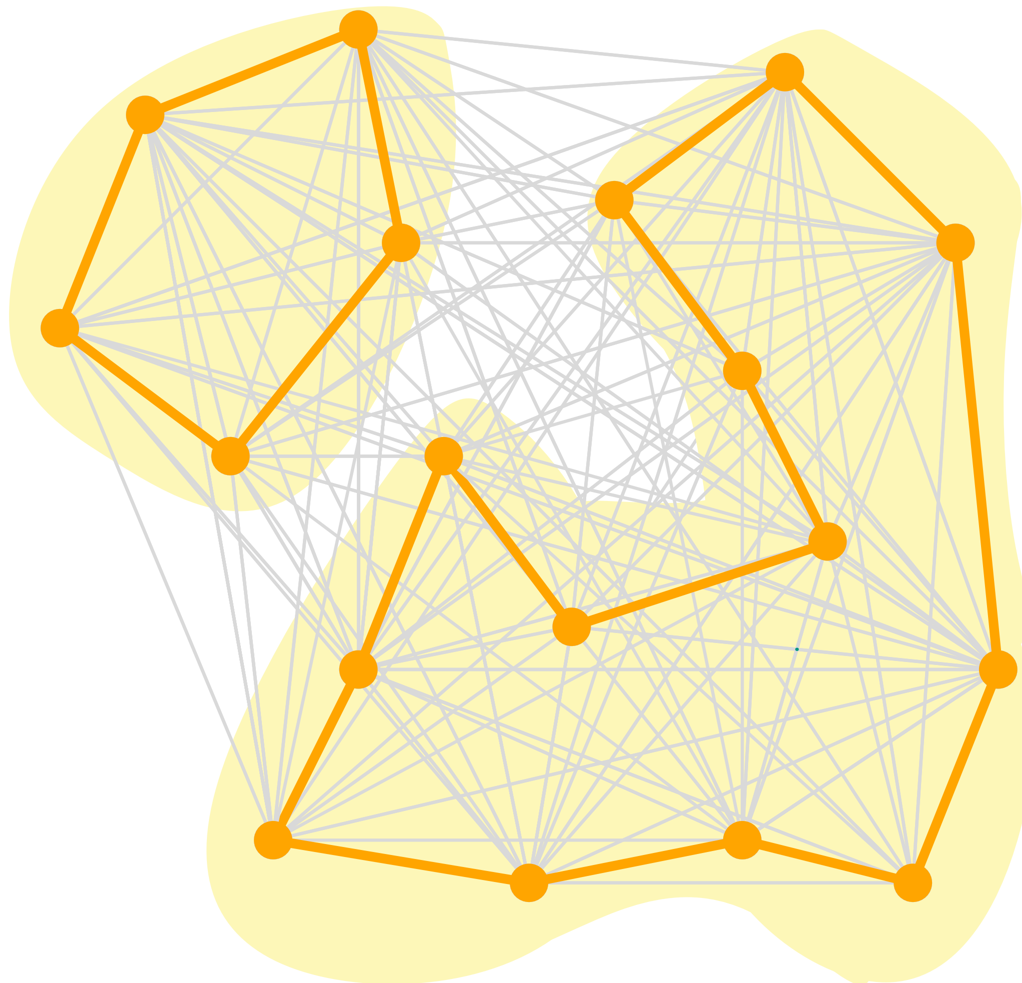
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



Institute of Theoretical Informatics
Algorithmics Group

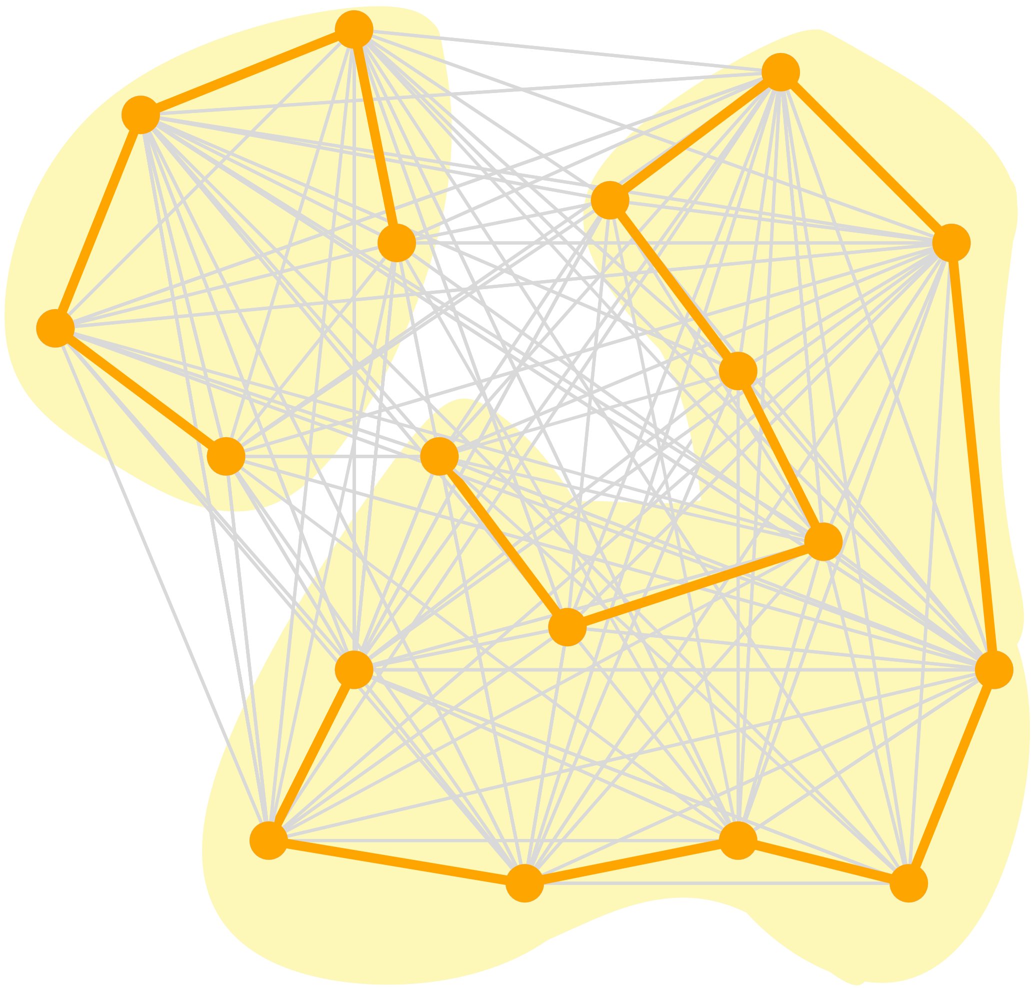
"SUBTOUR PATCHING"

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



"SUBTOUR PATCHING"

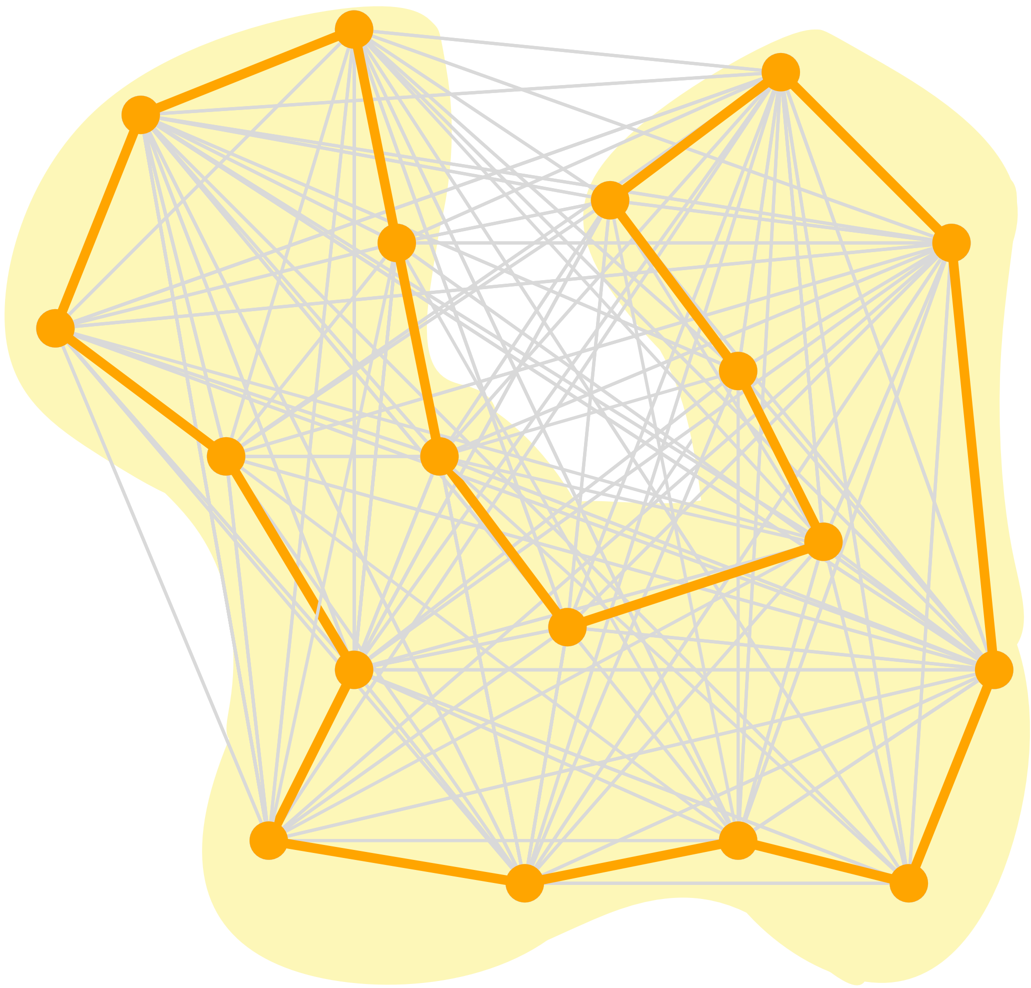
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



Institute of Theoretical Informatics
Algorithmics Group

"SUBTOUR PATCHING"

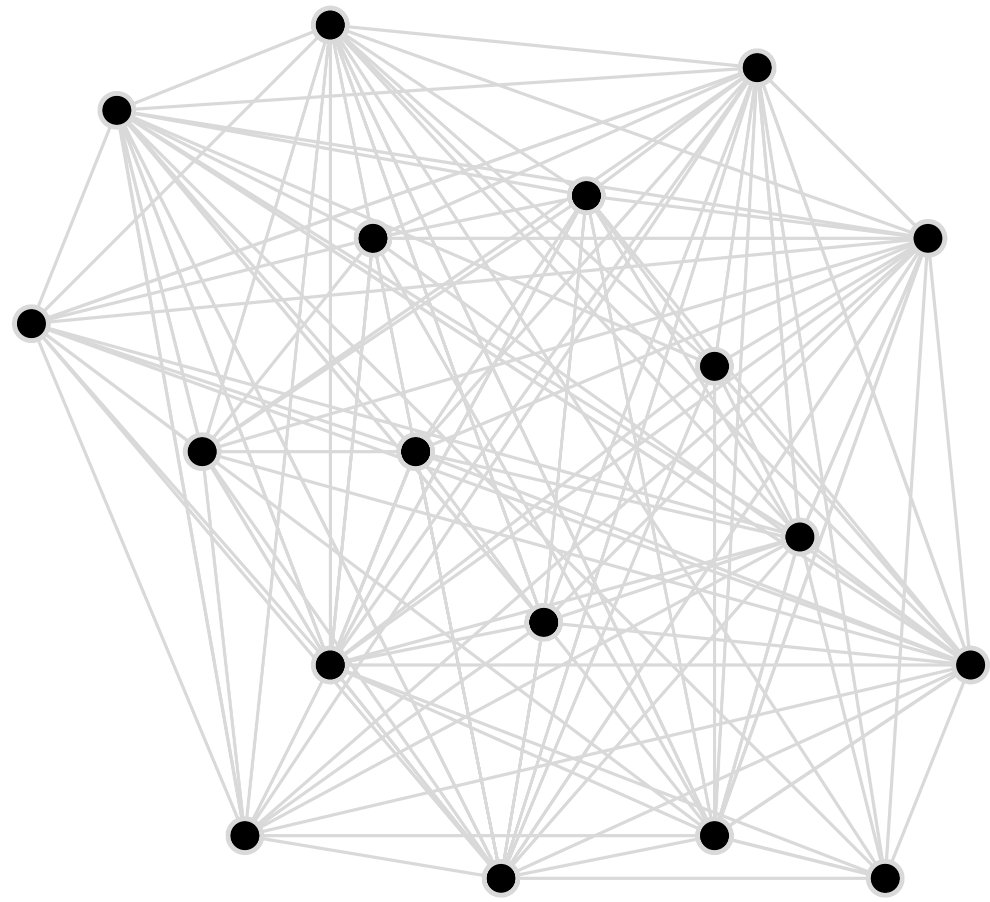
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



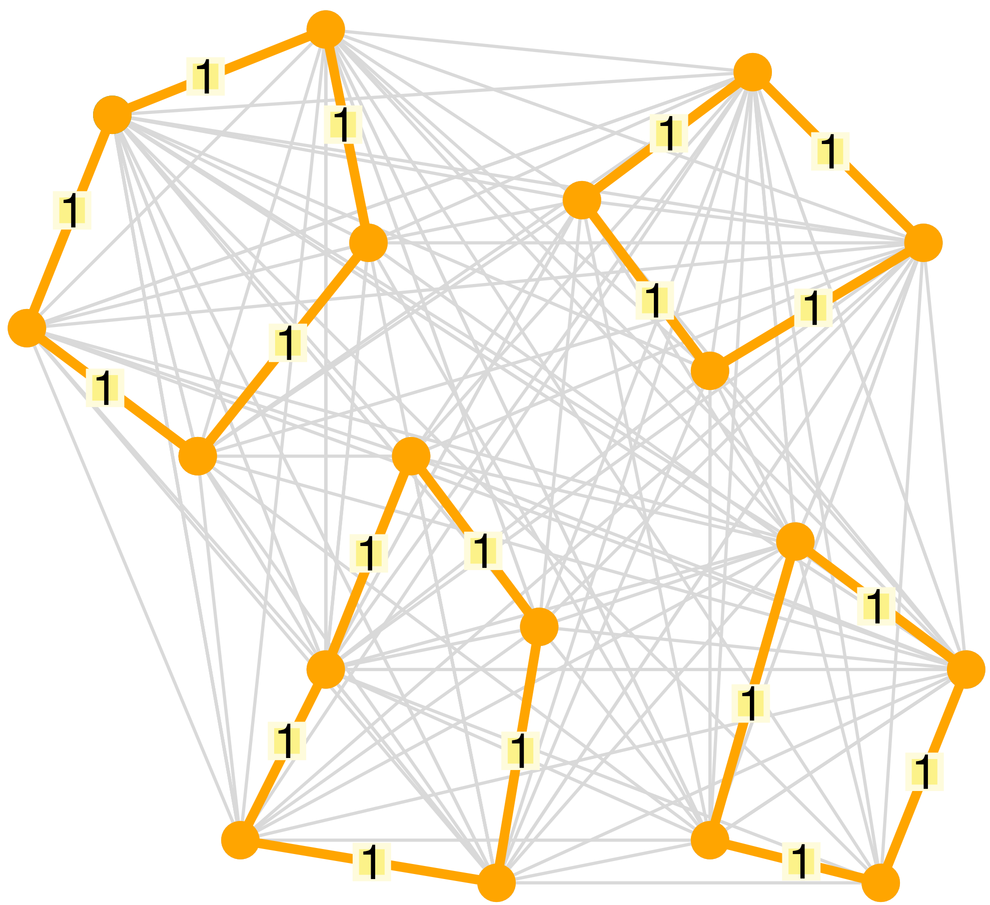
KOSTEN IM WORST CASE

Karlsruhe Institute of Technology

- Start:
  Optimales 2-Matching

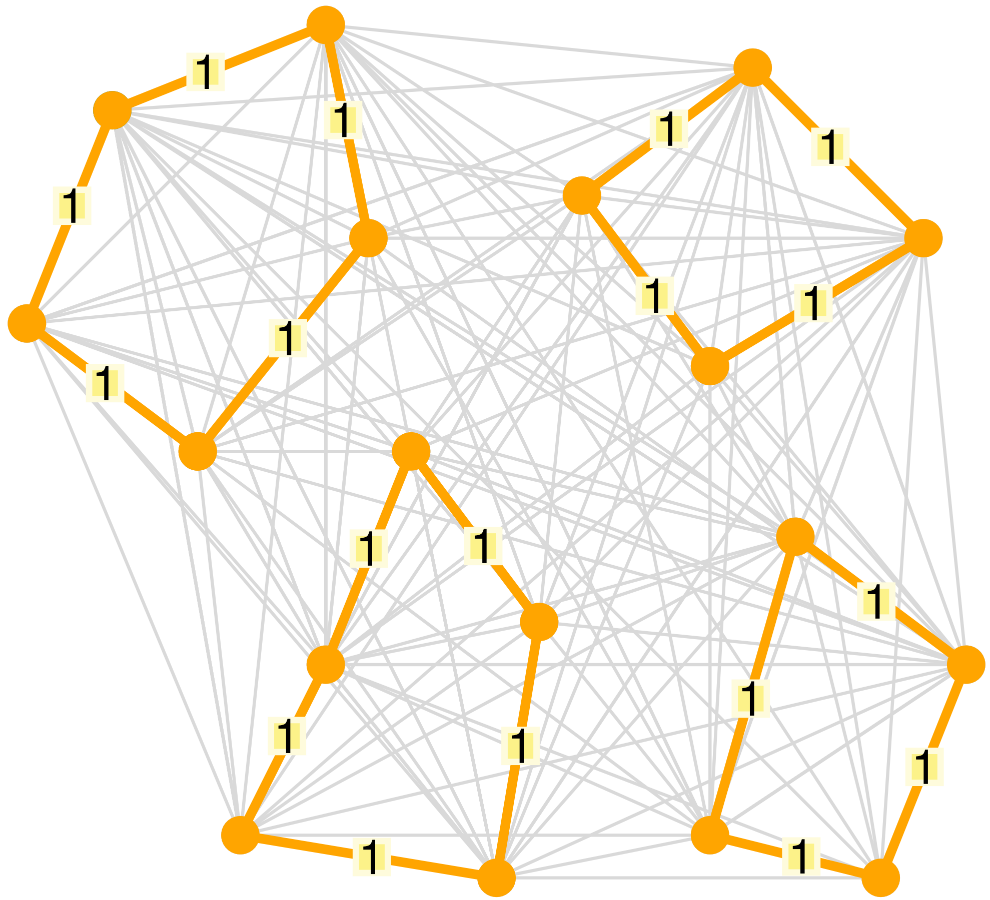


- Start:
  Optimales 2-Matching



KOSTEN IM WORST CASE

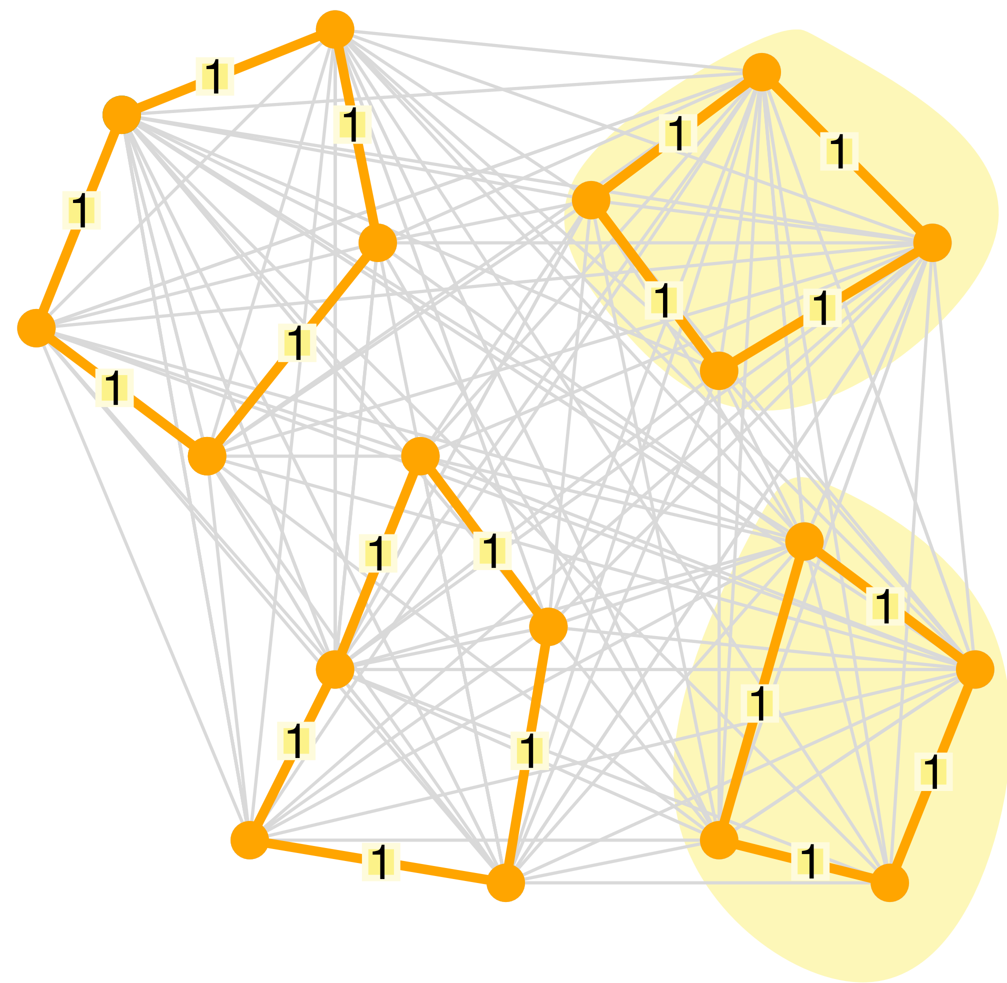
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen



"SUBTOUR PATCHING" KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching

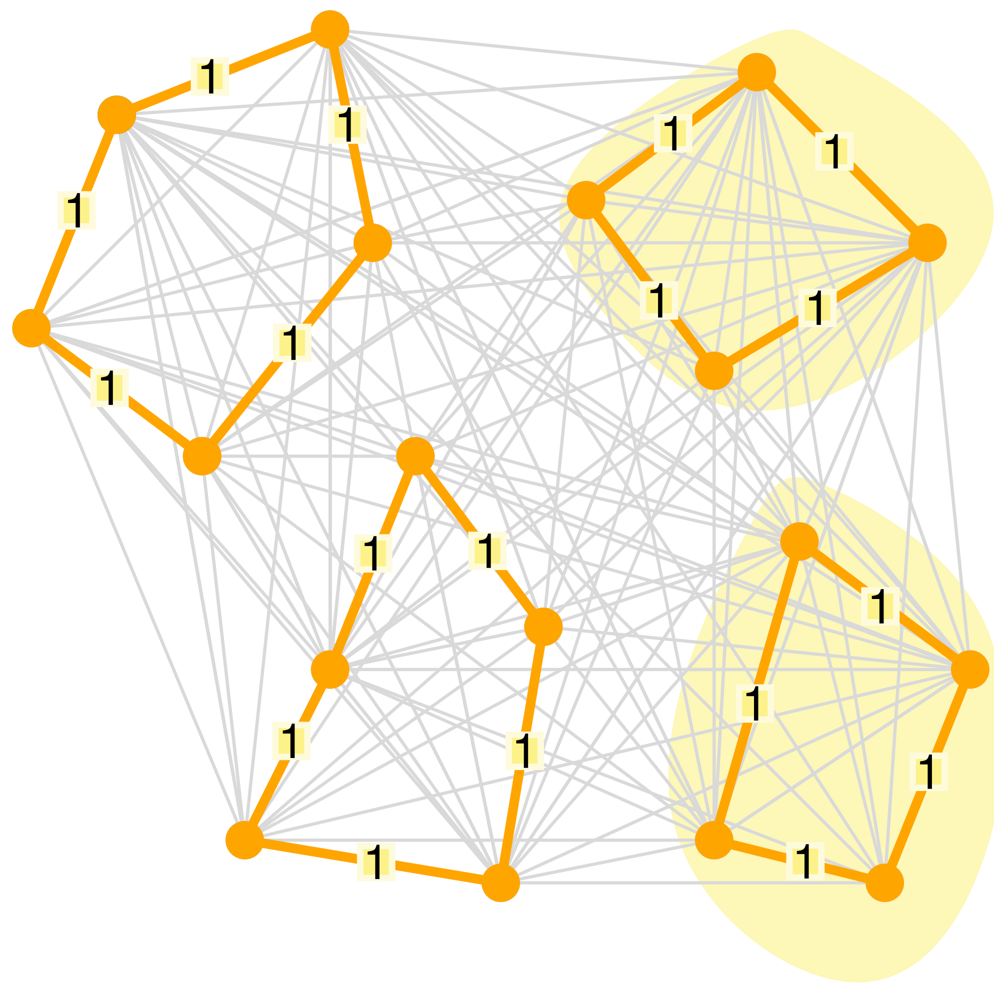
- Zwei Kreise wählen



Institute of Theoretical Informatics
Algorithmics Group

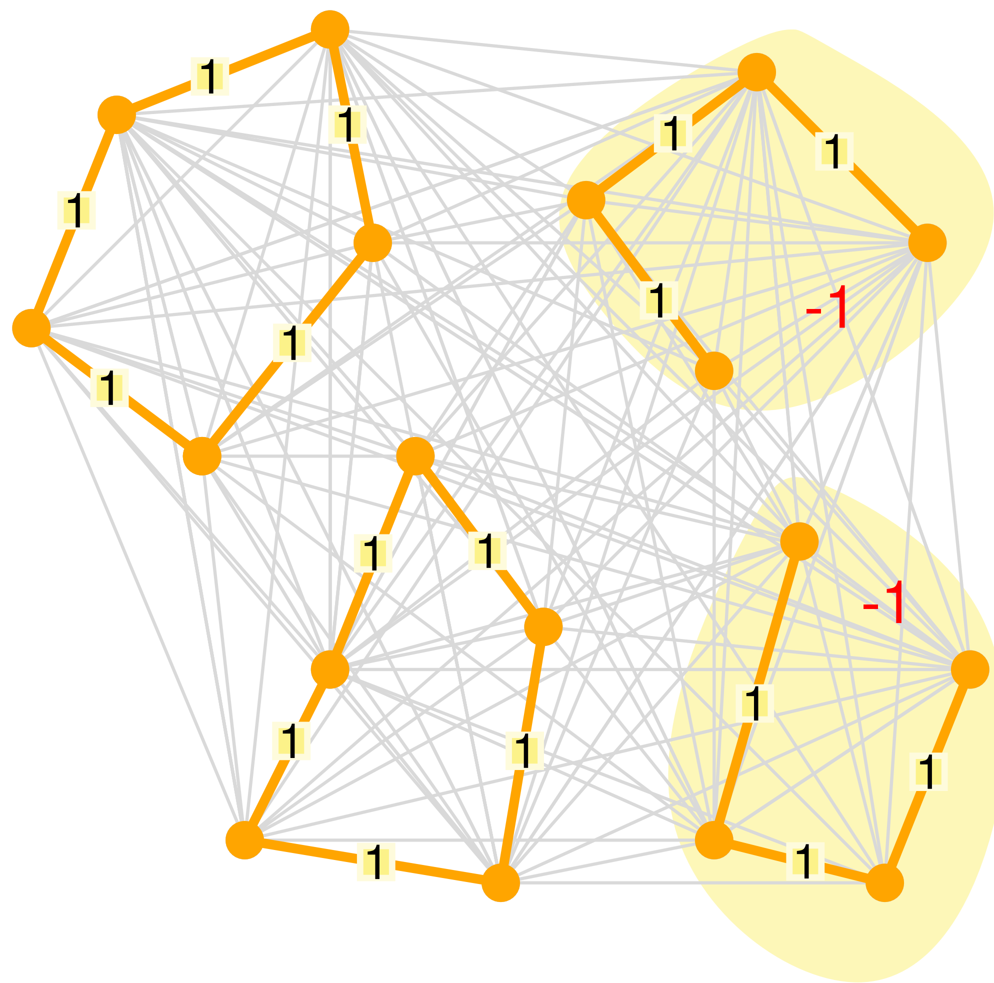
KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen



"SUBTOUR PATCHING" KOSTEN IM WORST CASE

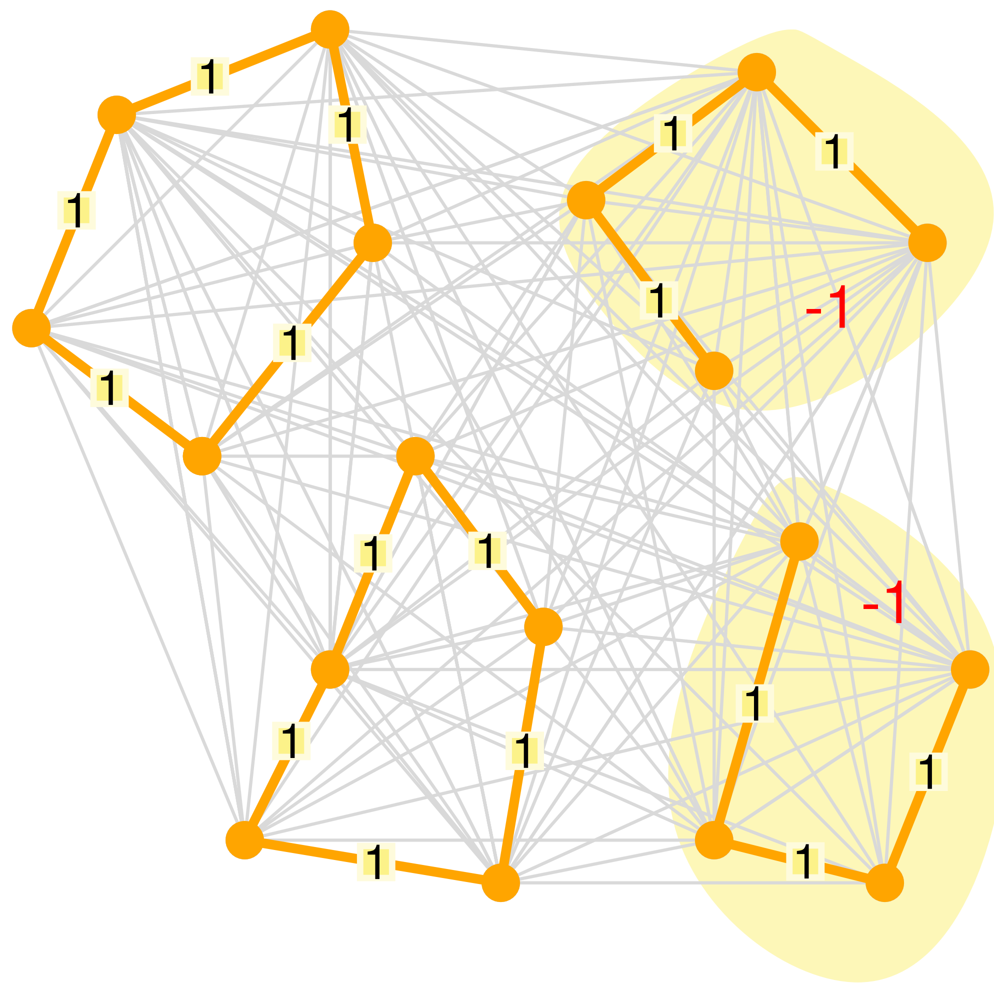
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen



Institute of Theoretical Informatics
Algorithmics Group

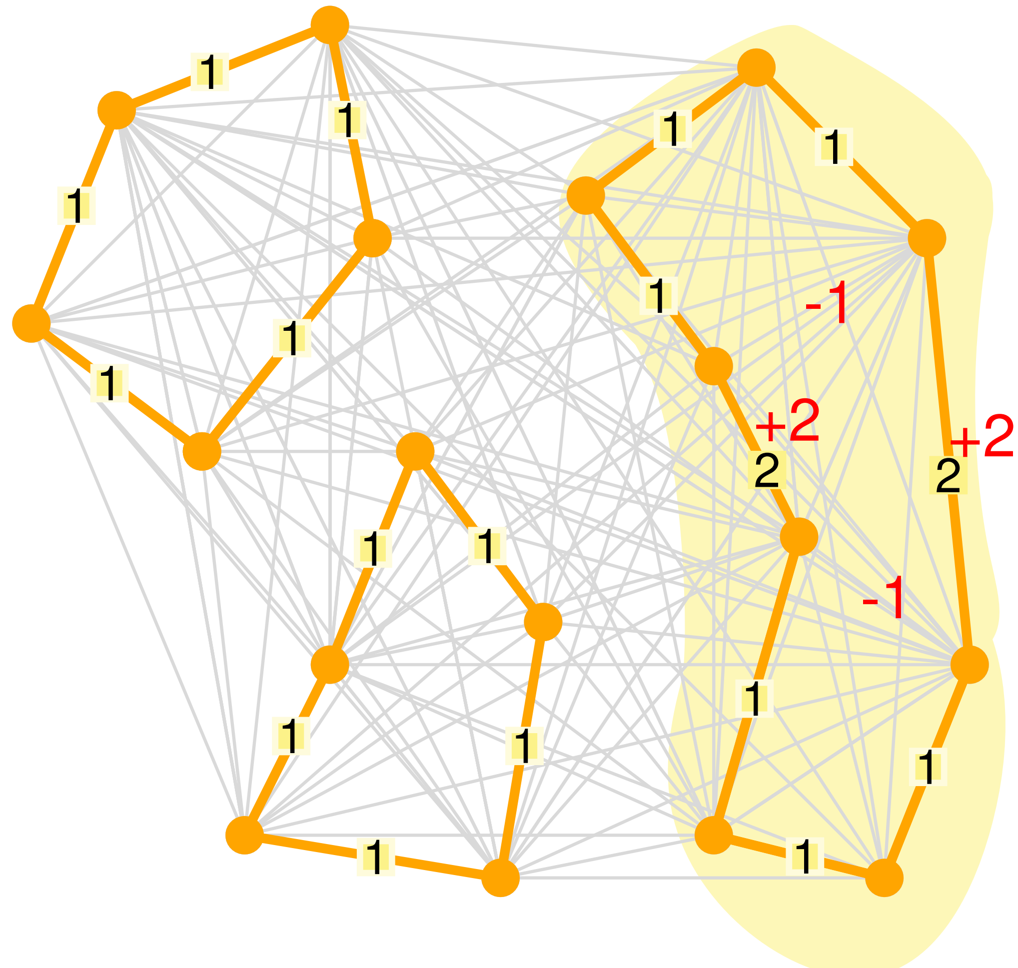
KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



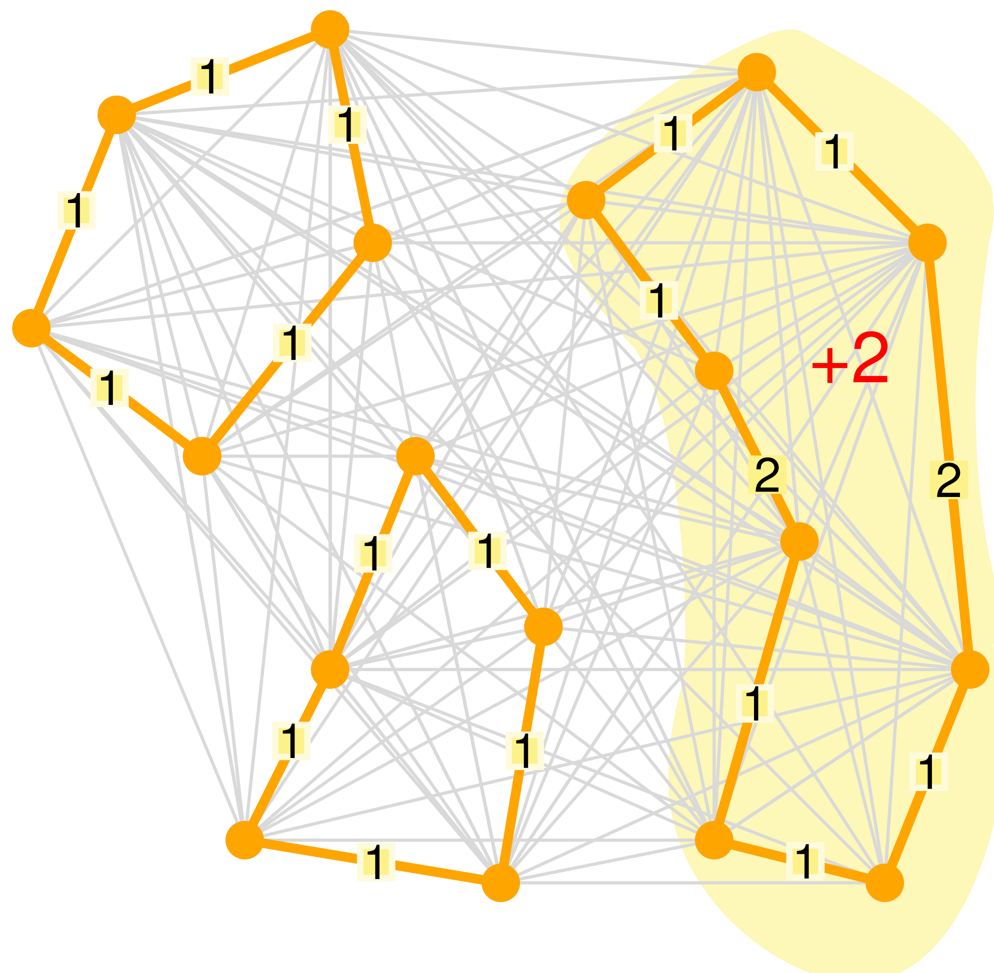
KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



KOSTEN IM WORST CASE

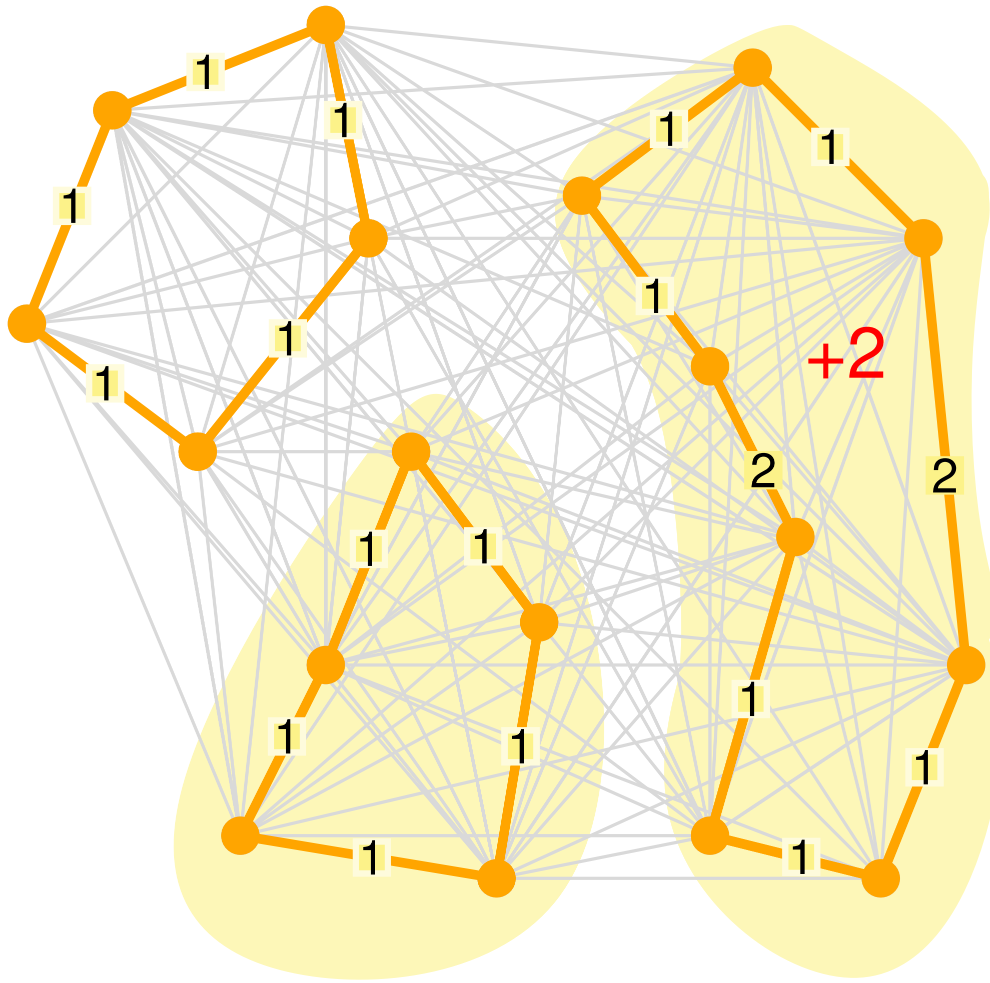
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



Institute of Theoretical Informatics
Algorithmics Group

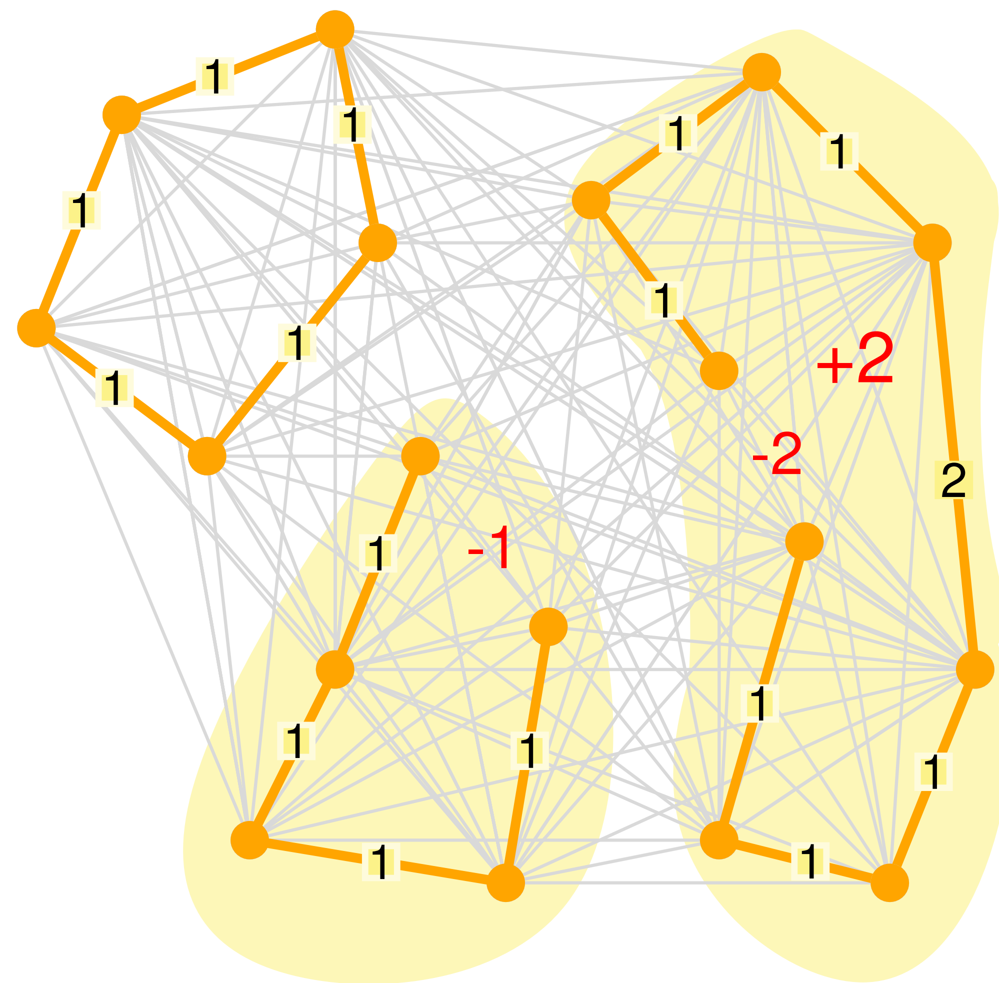
KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



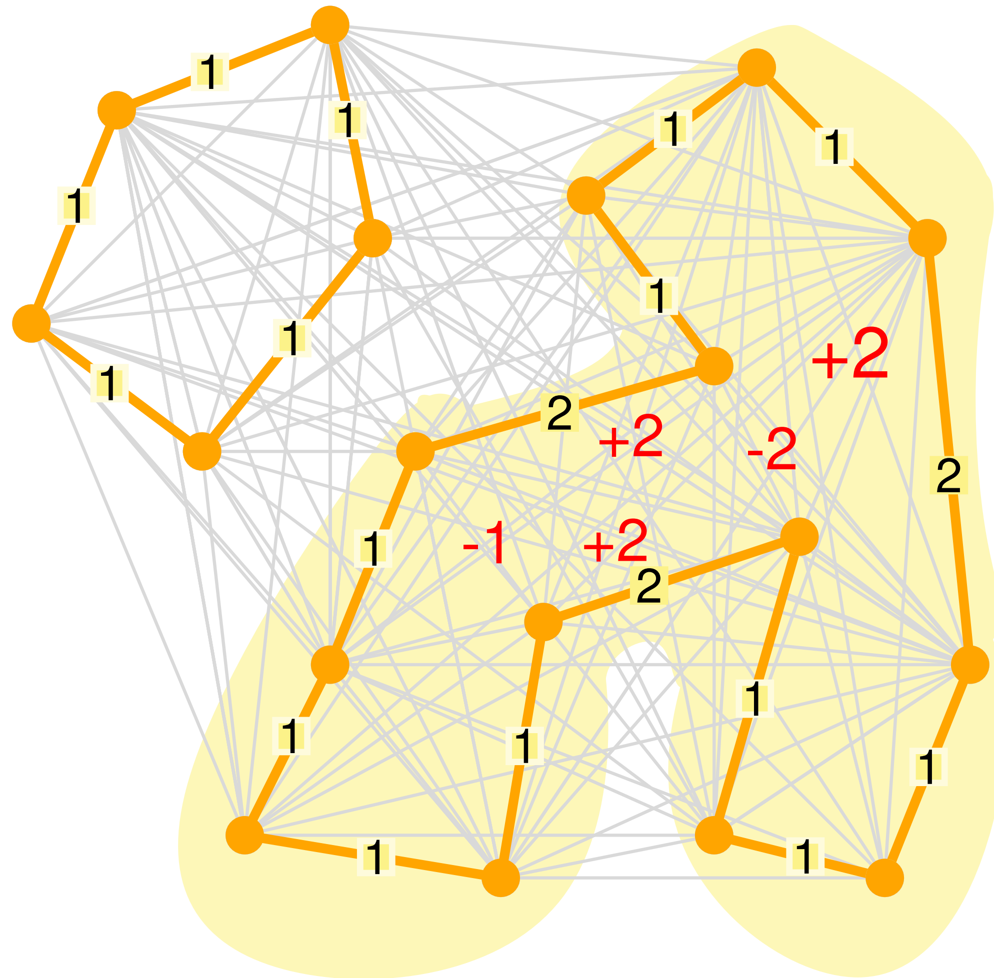
KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



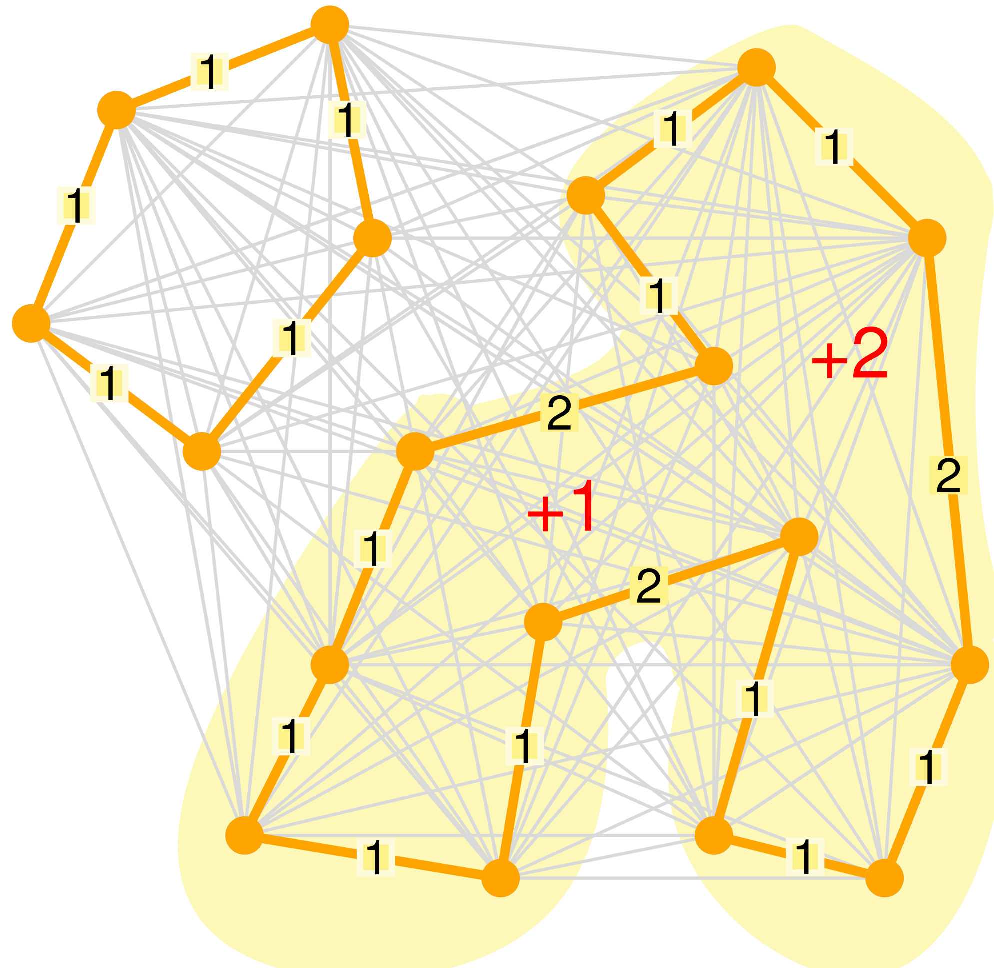
"SUBTOUR PATCHING" KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



KOSTEN IM WORST CASE

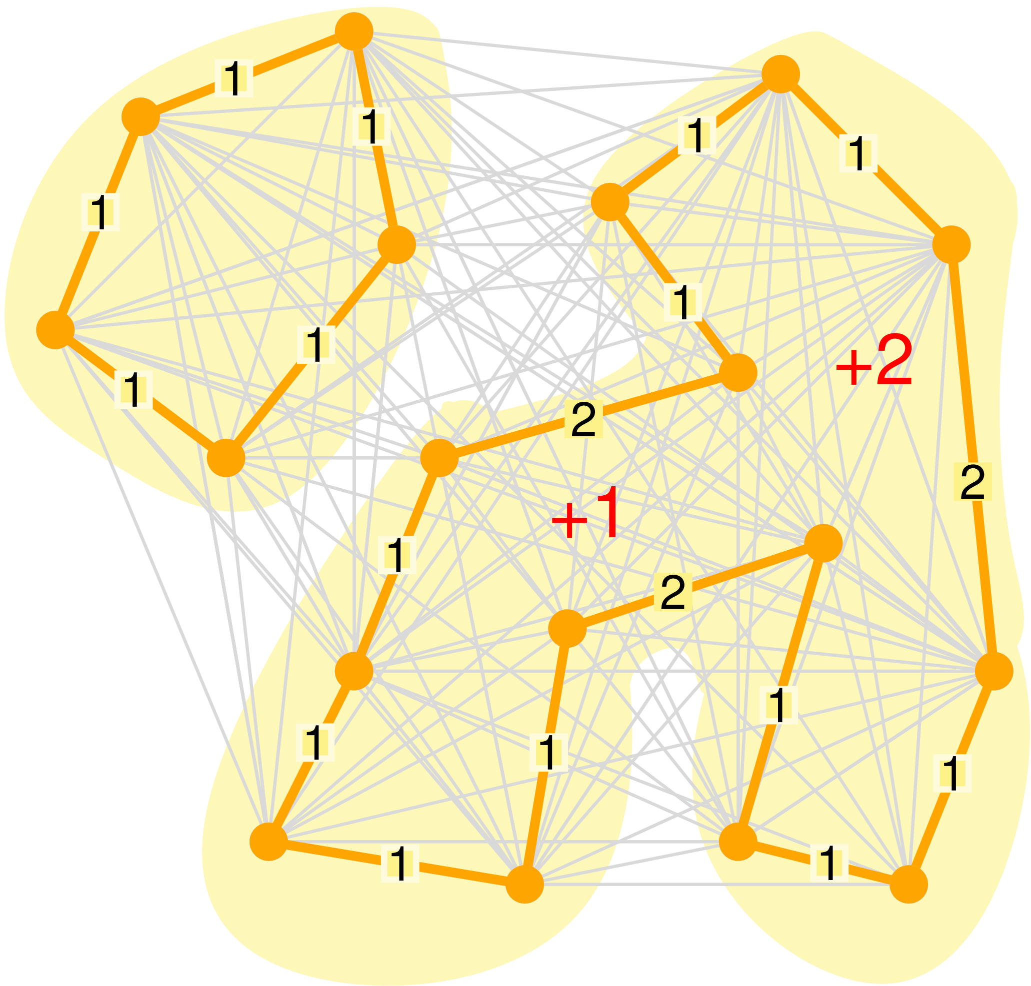
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



KOSTEN IM WORST CASE

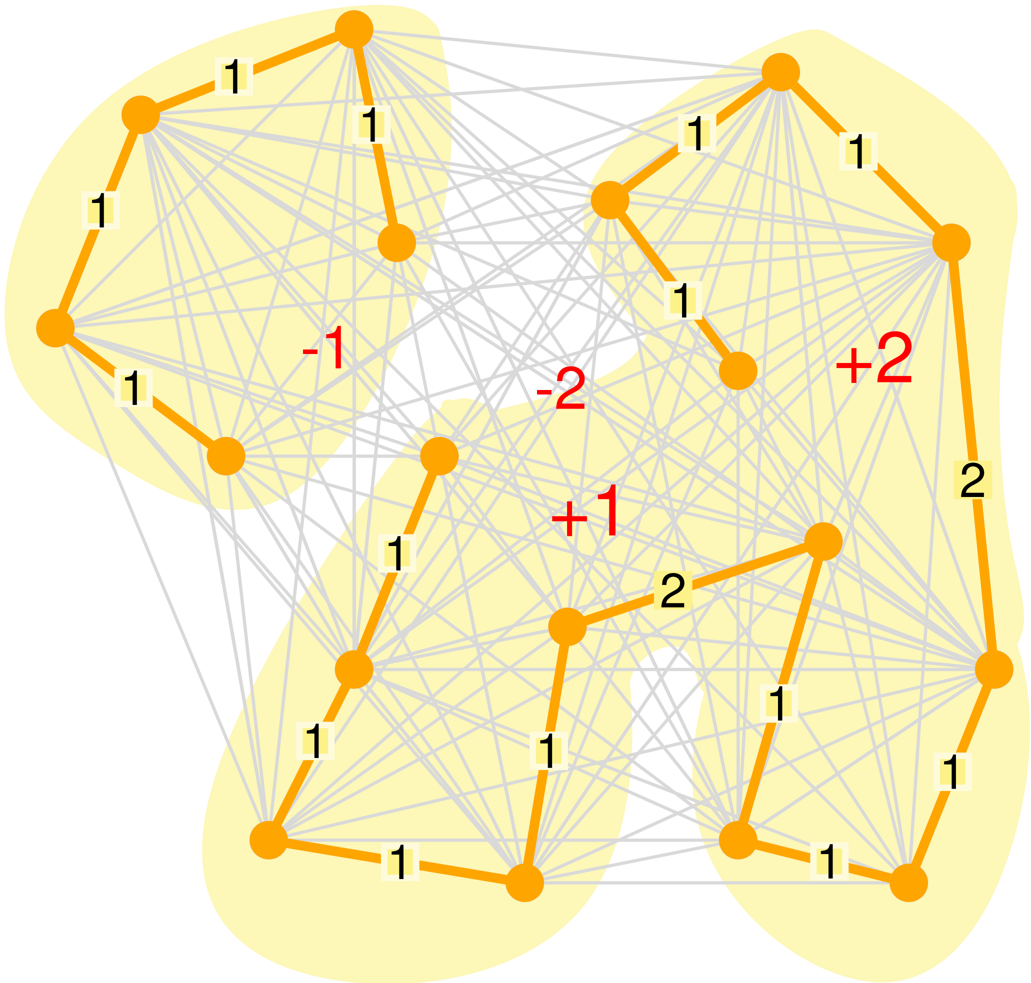
Karlsruhe Institute of Technology

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



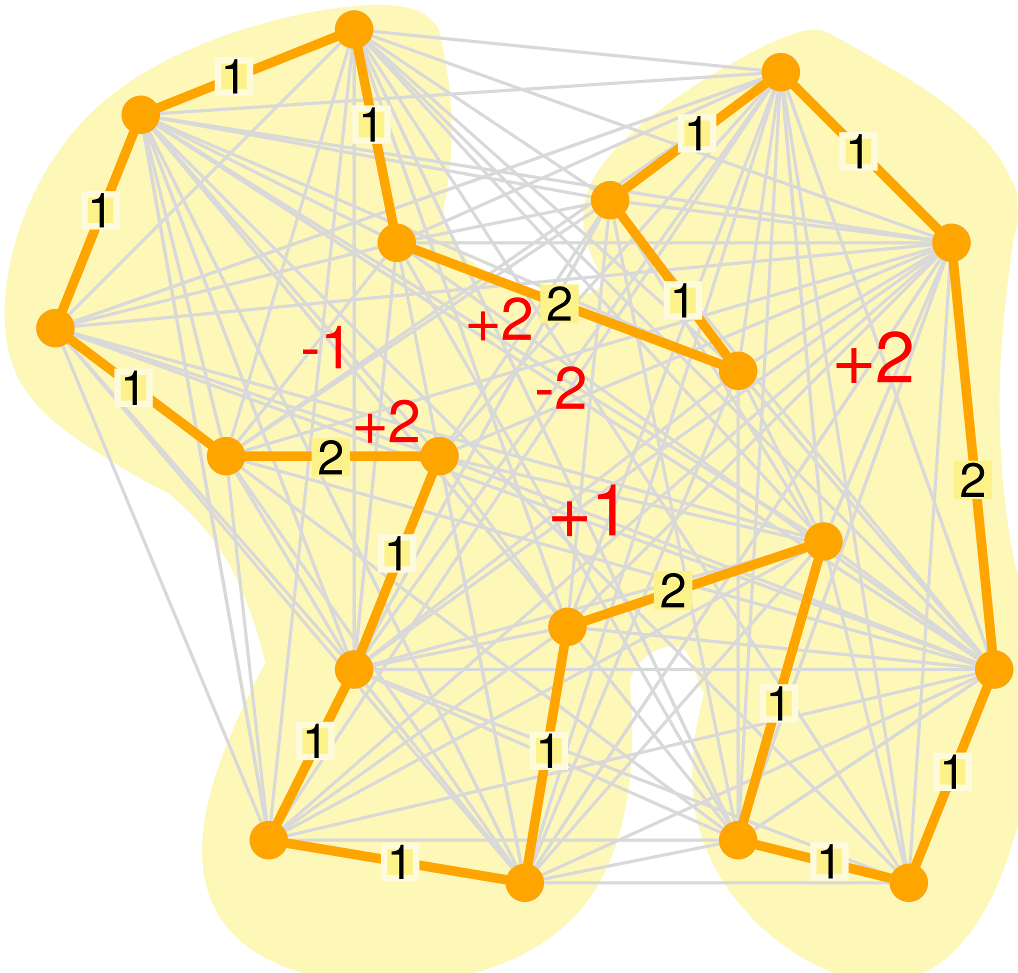
"SUBTOUR PATCHING" KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



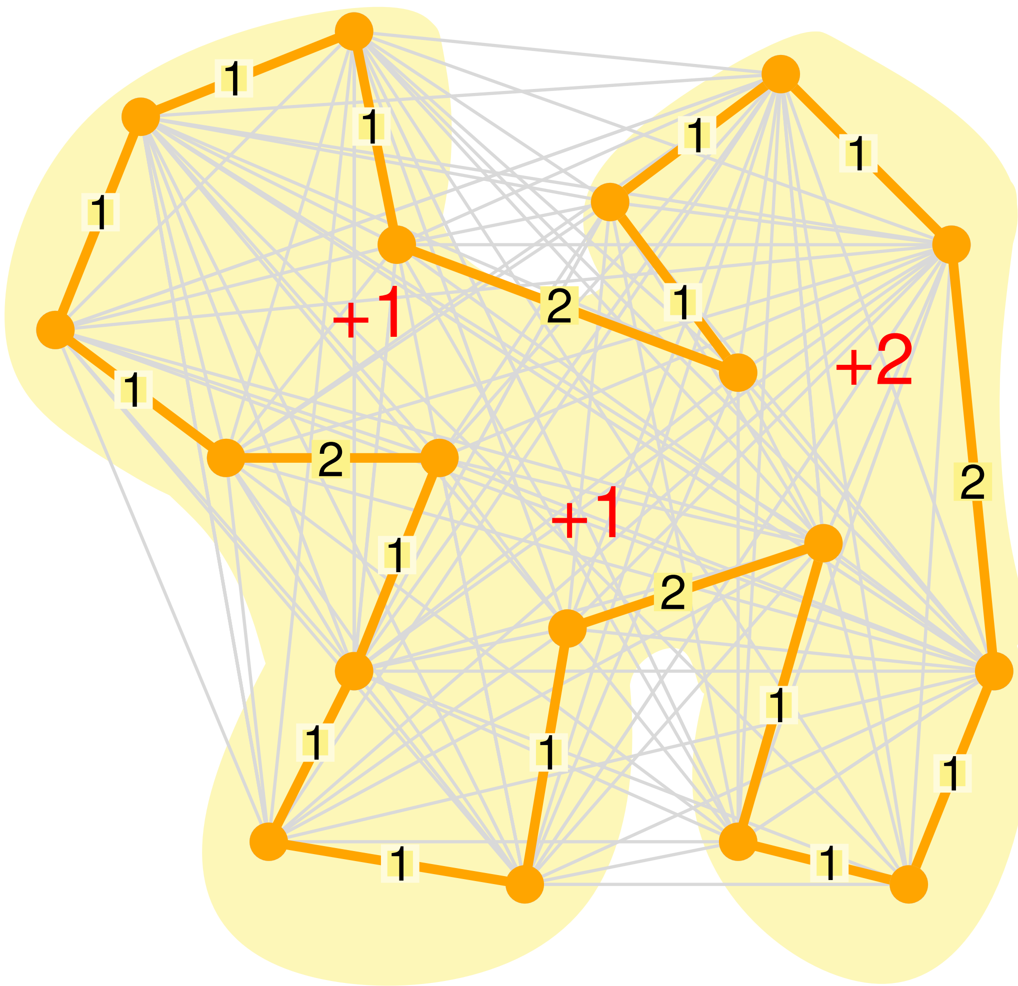
"SUBTOUR PATCHING" KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden



"SUBTOUR PATCHING" KOSTEN IM WORST CASE

- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden

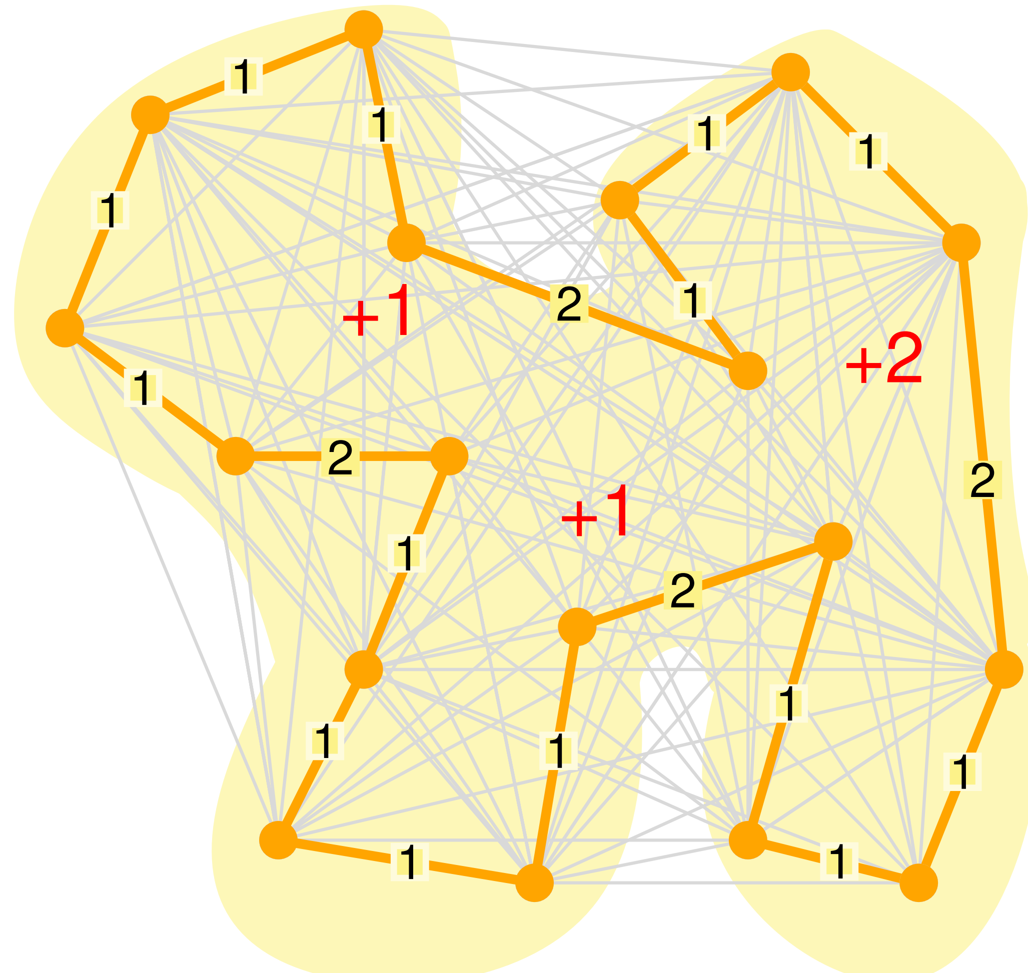


KOSTEN IM WORST CASE

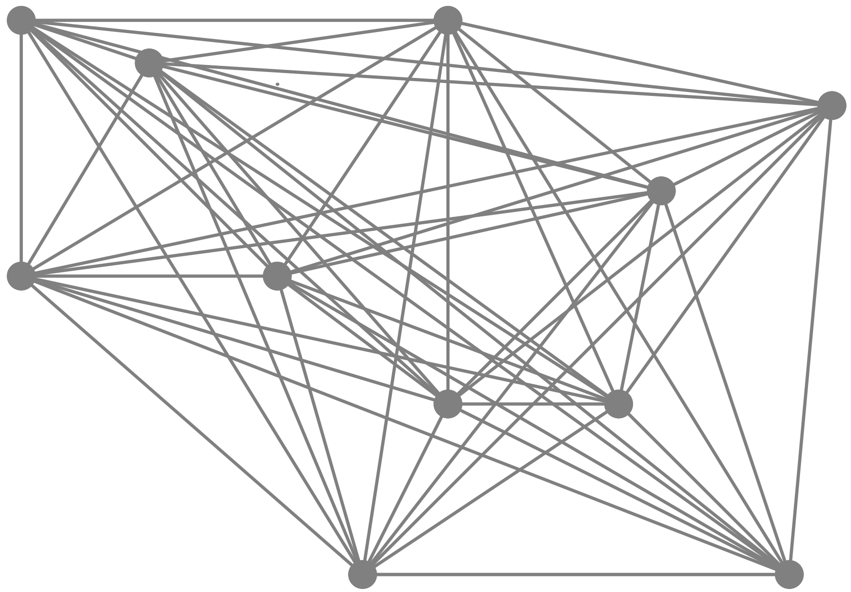
- Start:
  Optimales 2-Matching
- Zwei Kreise wählen
- Je eine Kante löschen
- Kreise verbinden

→ Kosten des optimalen 2-Matchings $+ \; n/3$

→ Güte: $4/3$

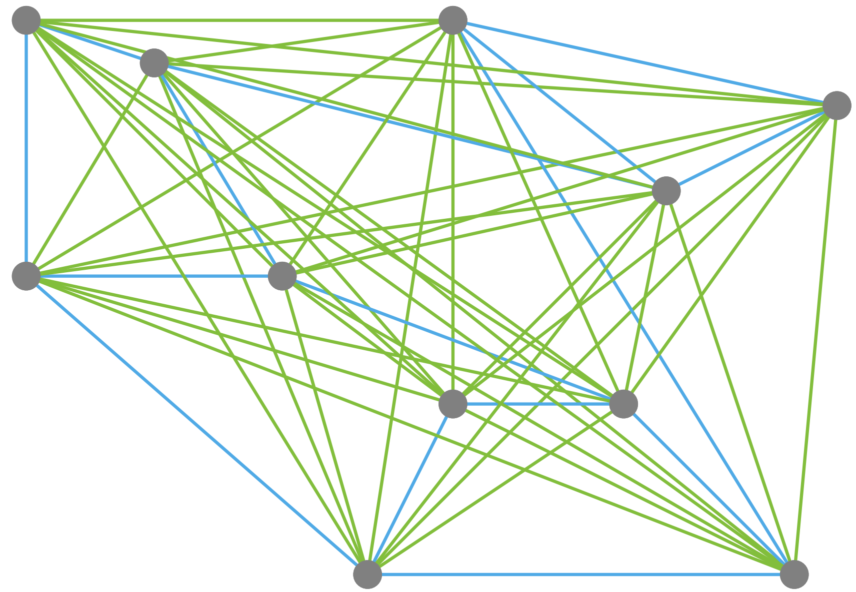


APPROXIMATIONSALGORITHMUS



$$G = (V, E) \quad w : E \to \{1, 2\}$$

APPROXIMATIONSALGORITHMUS



$$G = (V, E) \quad w : E \rightarrow \{1, 2\}$$

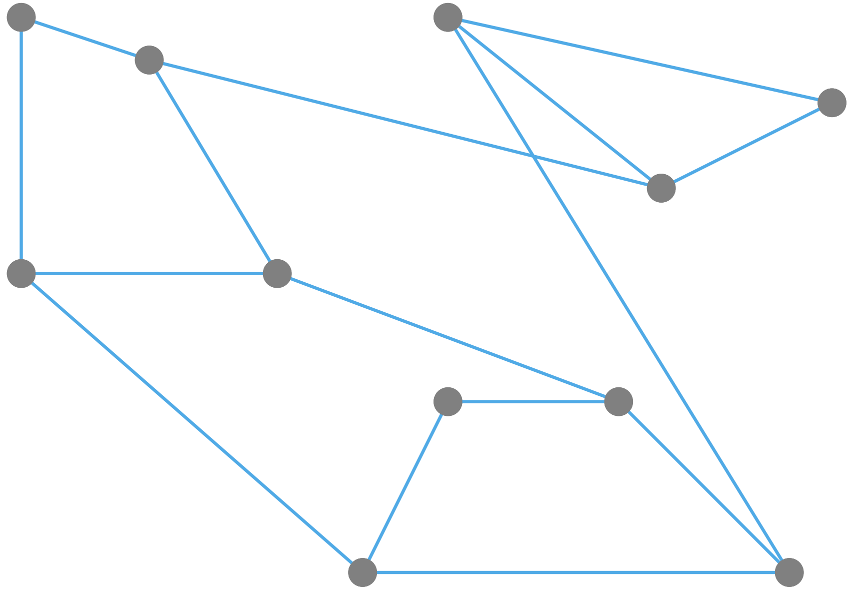
$$e \in E' \leftrightarrow e \in E, w(e) = 1$$

$$e \in E'' \leftrightarrow e \in E, w(e) = 2$$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

$G_d = (V, E')$



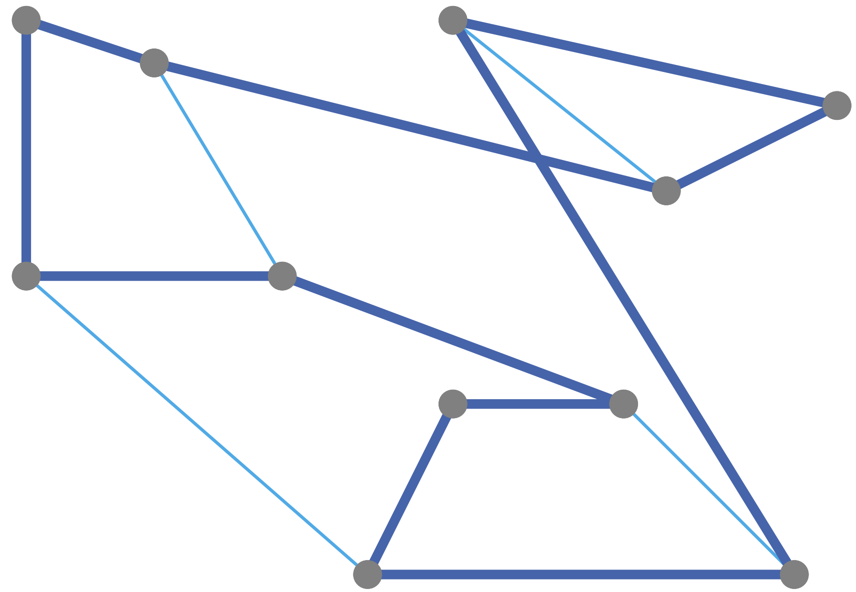
$G = (V, E) \quad w : E \to \{1, 2\}$

$e \in E' \leftrightarrow e \in E, w(e) = 1$

APPROXIMATIONSALGORITHMUS

Hamiltonkreis in G_d?

$G_d = (V, E')$



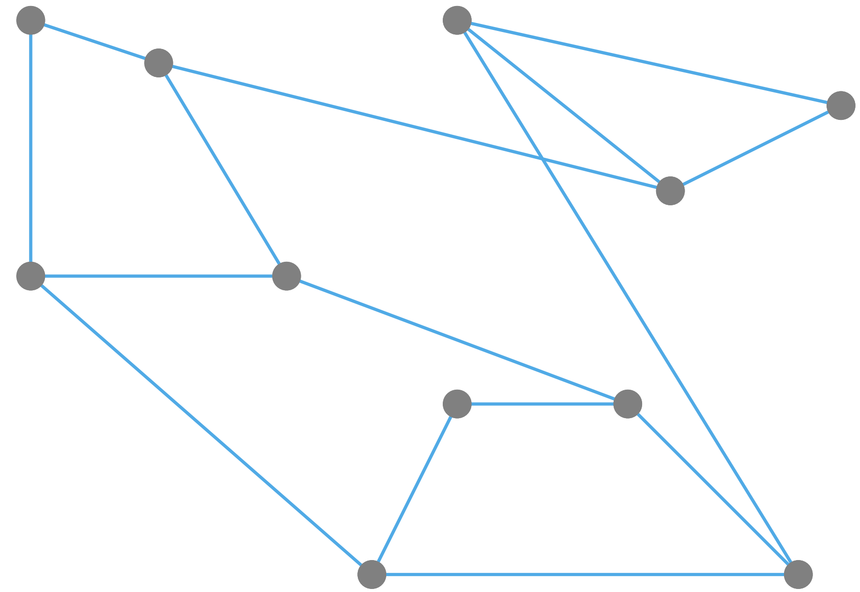
$G = (V, E) \quad w : E \to \{1, 2\}$

$e \in E' \leftrightarrow e \in E, w(e) = 1$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

$G_d = (V, E')$



$G = (V, E) \quad w : E \to \{1, 2\}$

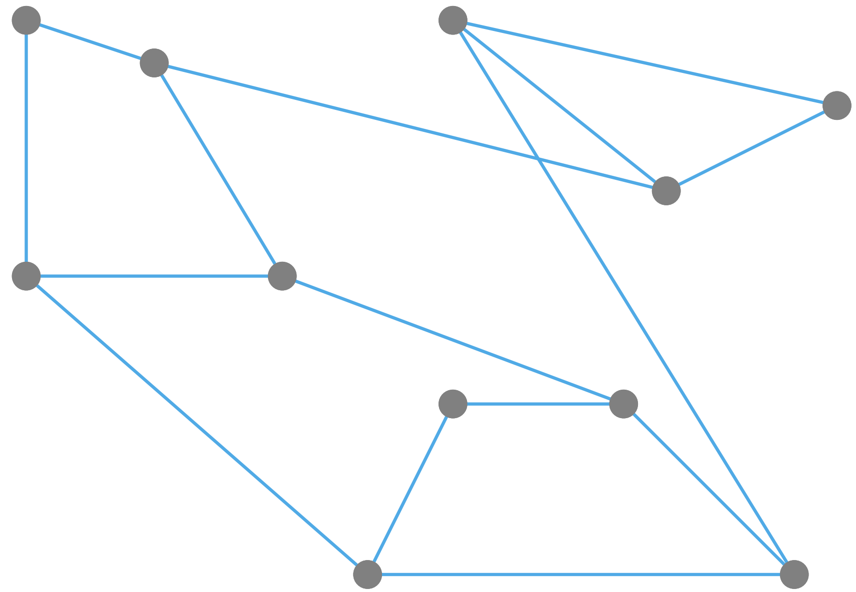
$e \in E' \leftrightarrow e \in E, w(e) = 1$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

$G_d = (V, E')$



$G = (V, E) \quad w : E \to \{1, 2\}$

$e \in E' \leftrightarrow e \in E, w(e) = 1$

Institute of Theoretical Informatics
Algorithmics Group

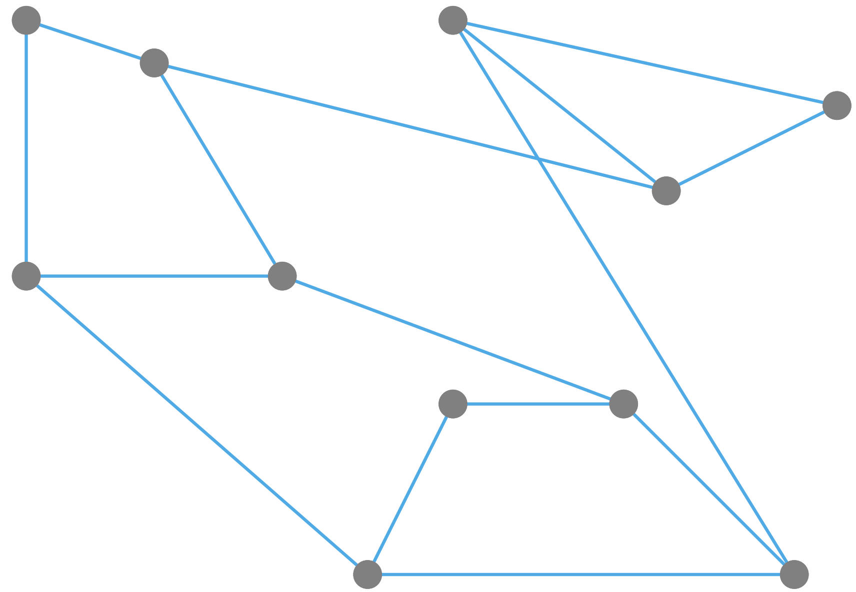
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen $C = \{c_1, ..., c_k\}$

$G_d = (V, E')$



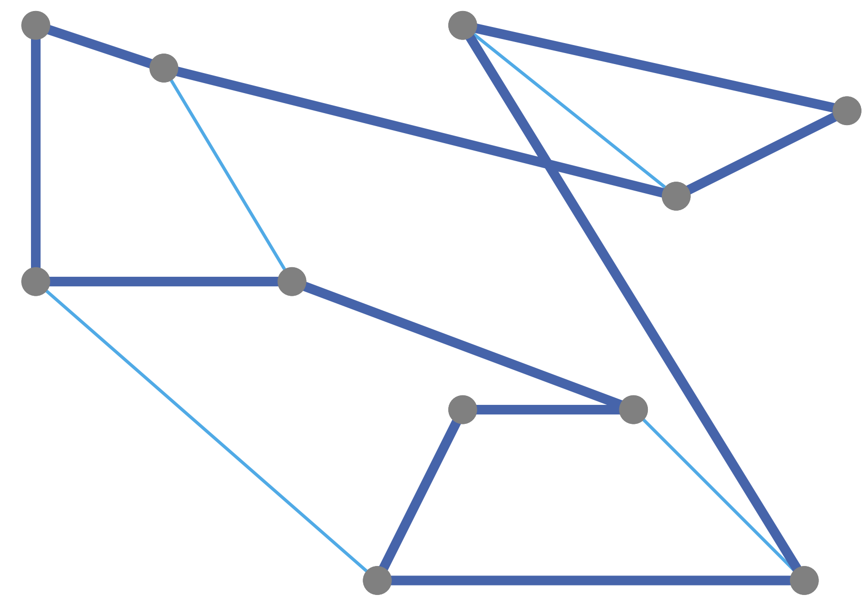
$G = (V, E) \quad w : E \to \{1, 2\}$

$e \in E' \leftrightarrow e \in E, w(e) = 1$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

$G_d = (V, E')$



- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen $C = \{c_1, \ldots, c_k\}$

→ Fall $|C| = 1$: fertig

$G = (V, E) \quad w : E \to \{1, 2\}$

$e \in E' \leftrightarrow e \in E, w(e) = 1$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

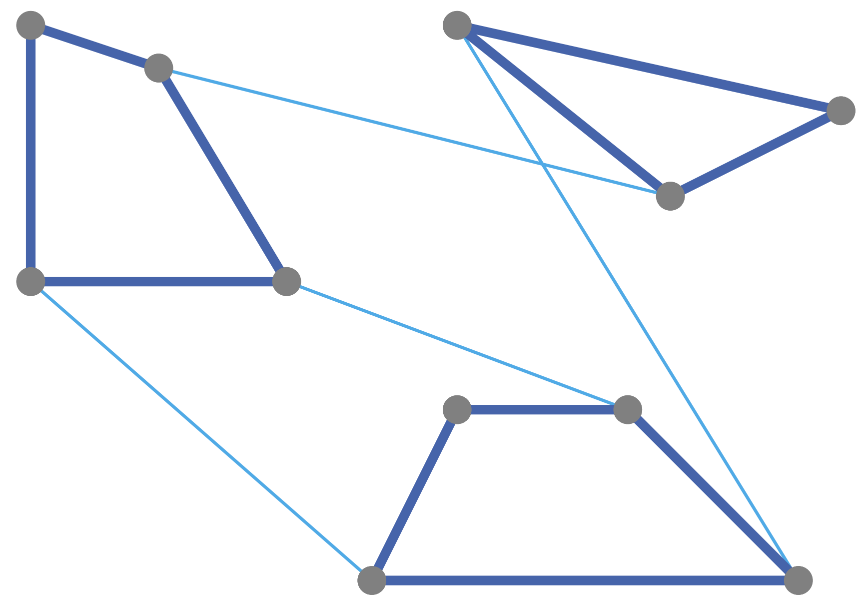
FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, ..., c_k\}$

→ Fall $|C| > 1$

$G_d = (V, E')$



$G = (V, E) \quad w : E \to \{1, 2\}$

$e \in E' \leftrightarrow e \in E, w(e) = 1$

APPROXIMATIONSALGORITHMUS

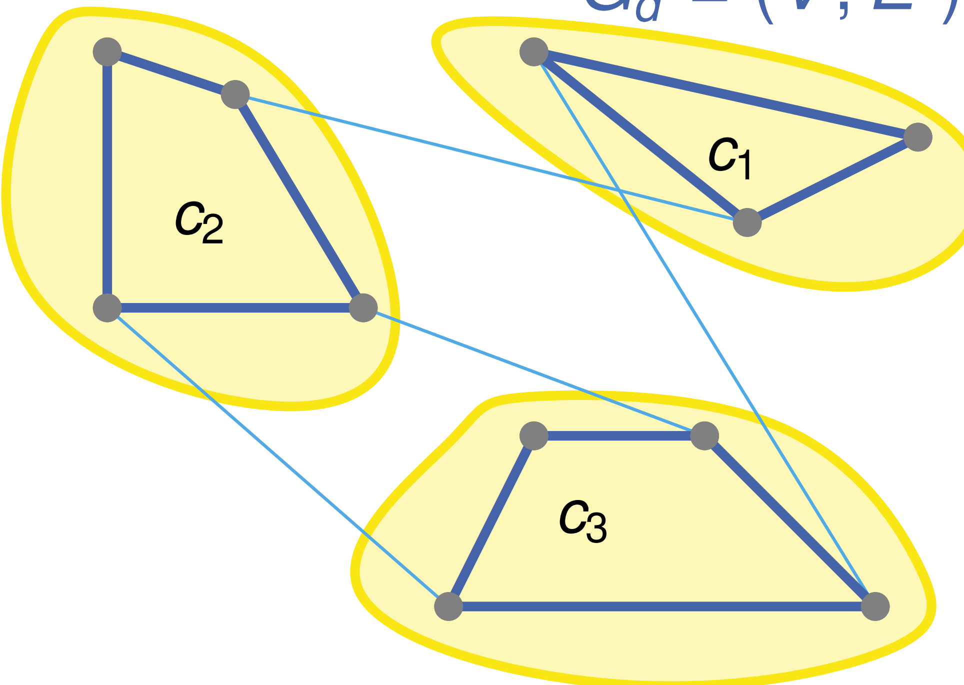
FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, ..., c_k\}$

→ Fall $|C| > 1$

$G_d = (V, E')$



$G = (V, E) \quad w : E \to \{1, 2\}$

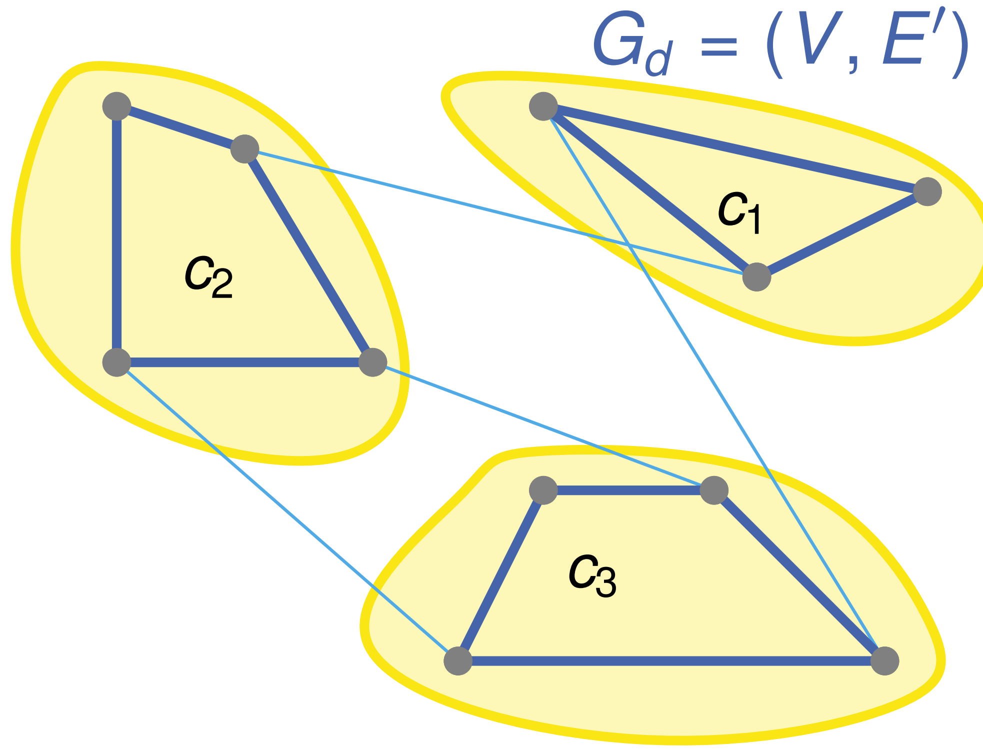
$e \in E' \leftrightarrow e \in E, w(e) = 1$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$

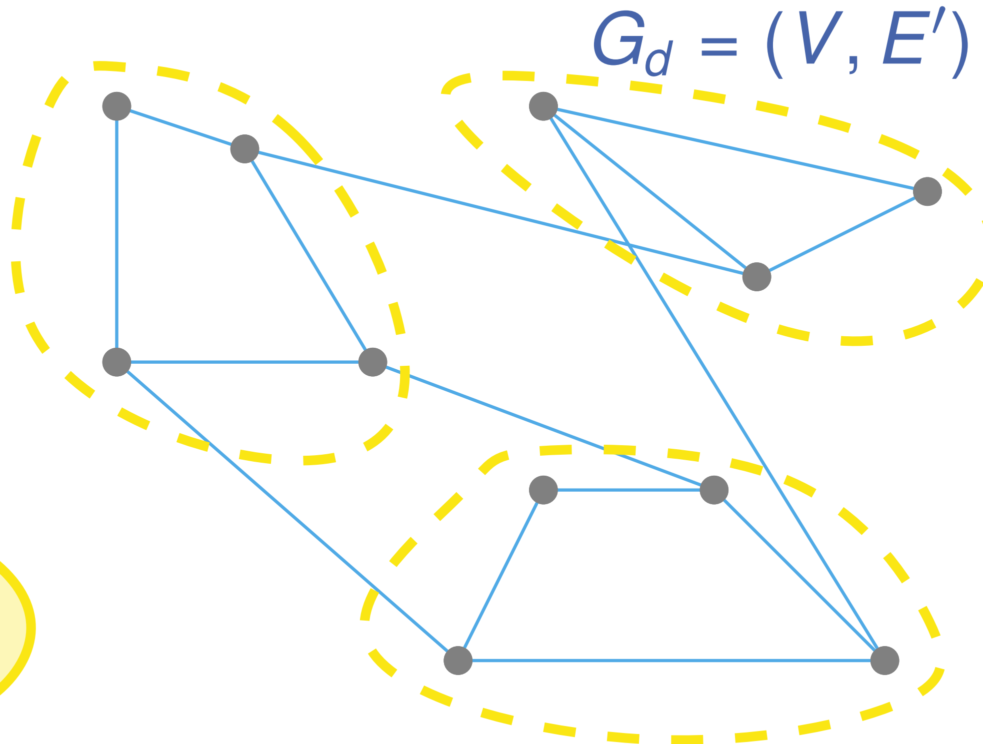


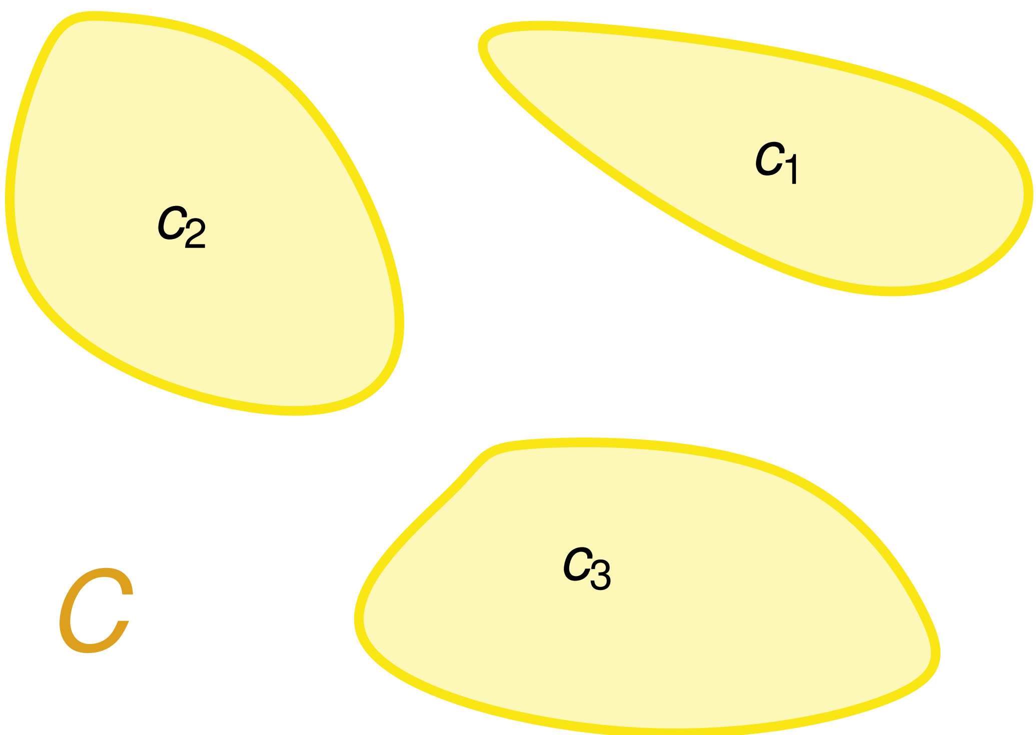
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, ..., c_k\}$





Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

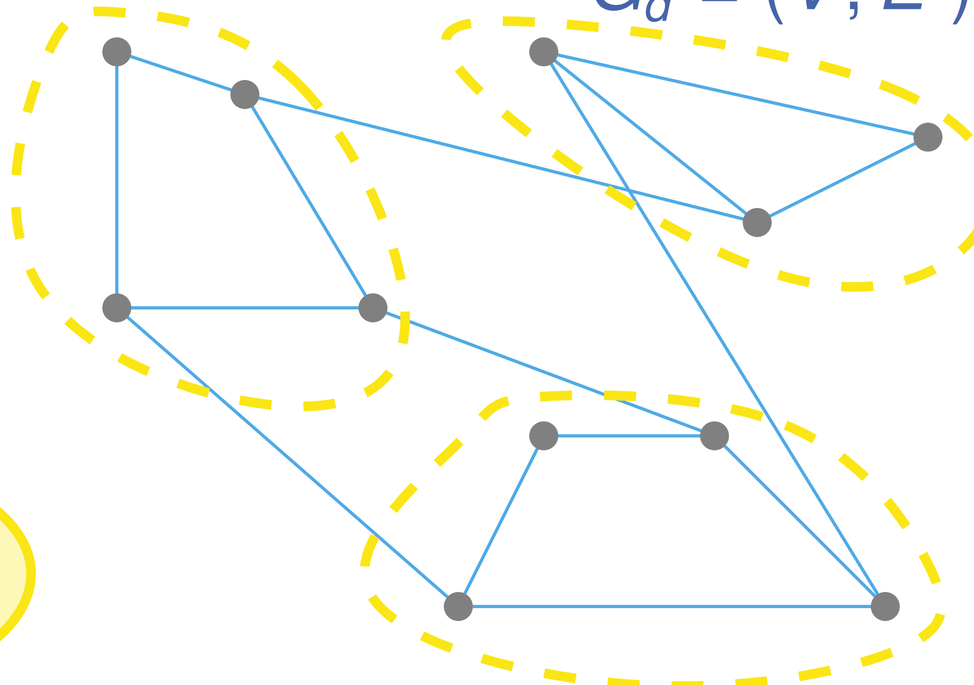
FALL 1: G_d hat Hamiltonkreis

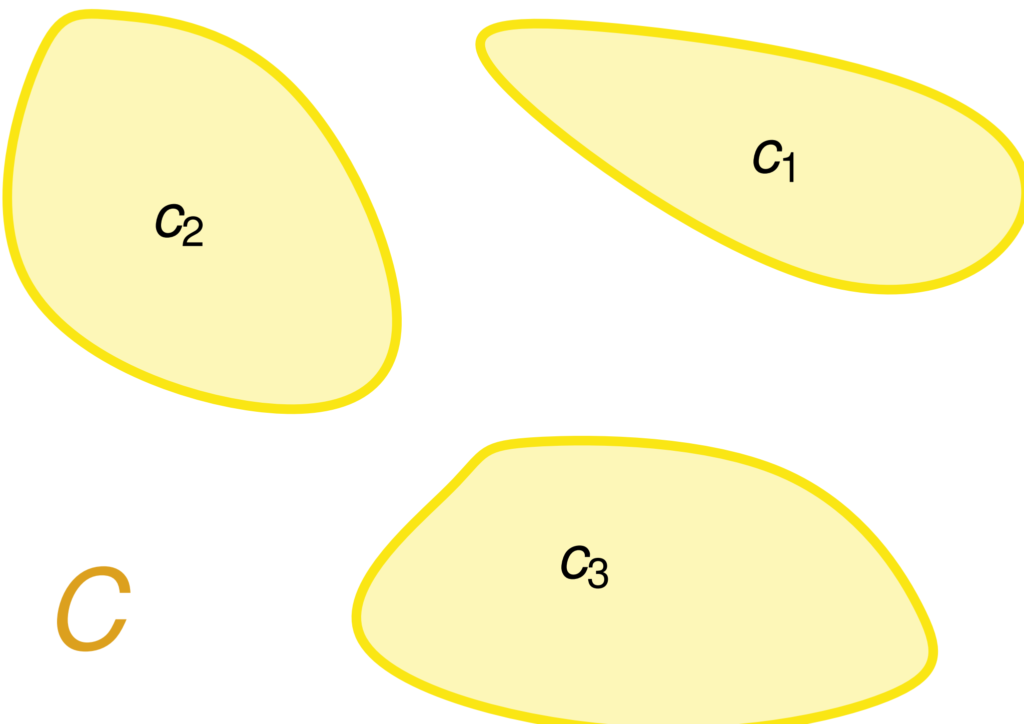
Karlsruhe Institute of Technology

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, ..., c_k\}$

$G_d = (V, E')$





C

$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

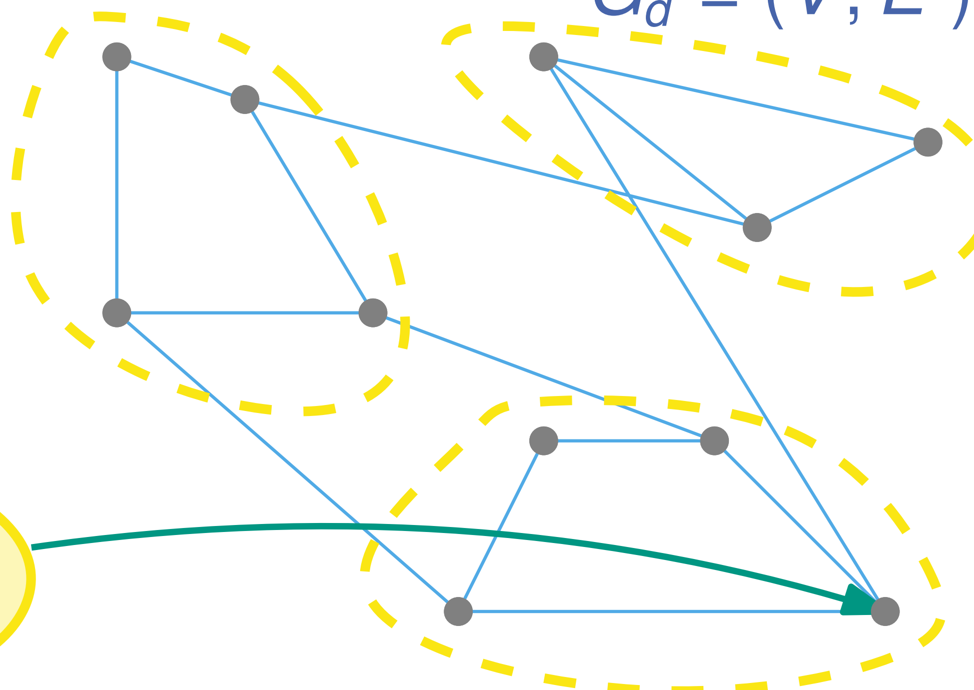
APPROXIMATIONSALGORITHMUS

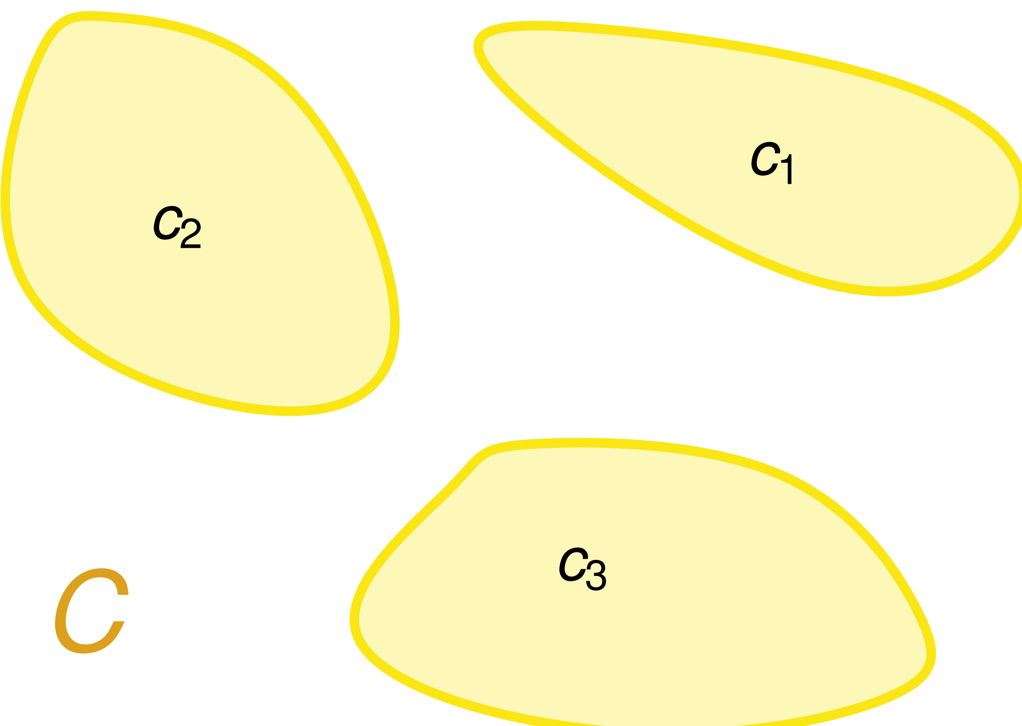
FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$

$G_d = (V, E')$





$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

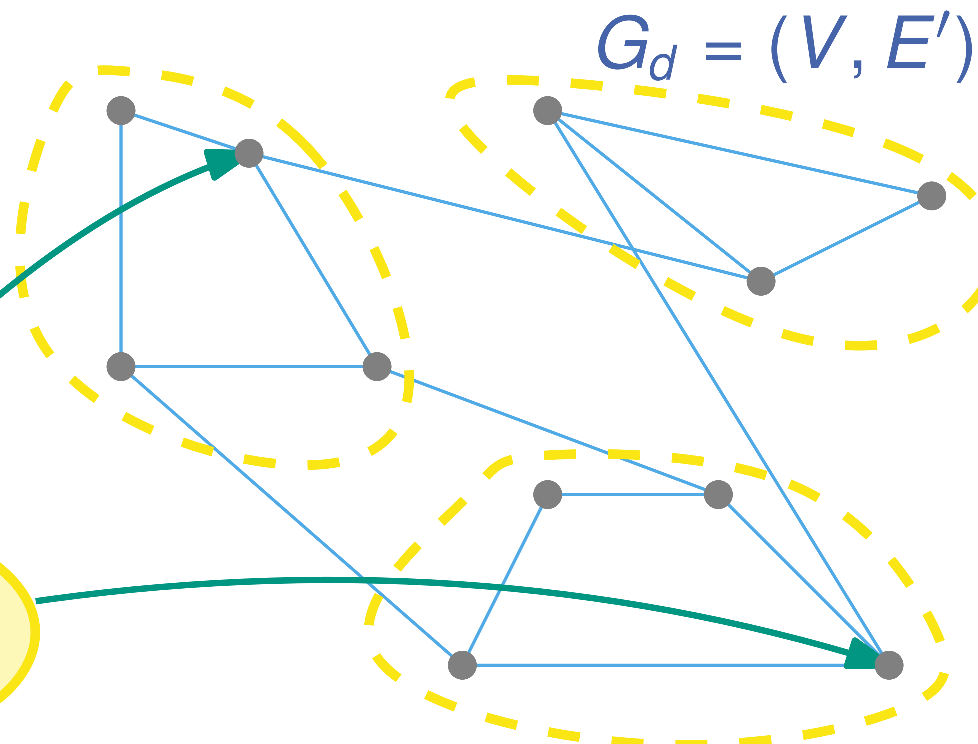
Institute of Theoretical Informatics
Algorithmics Group

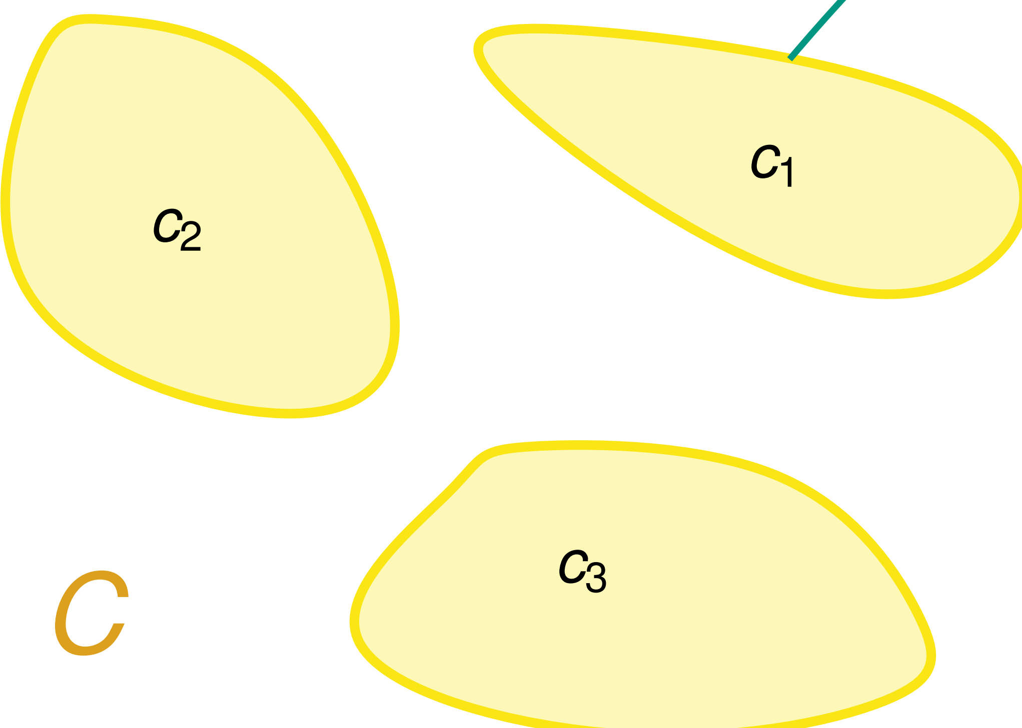
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen $C = \{c_1, \ldots, c_k\}$





$$B = (V \cup C, D)$$

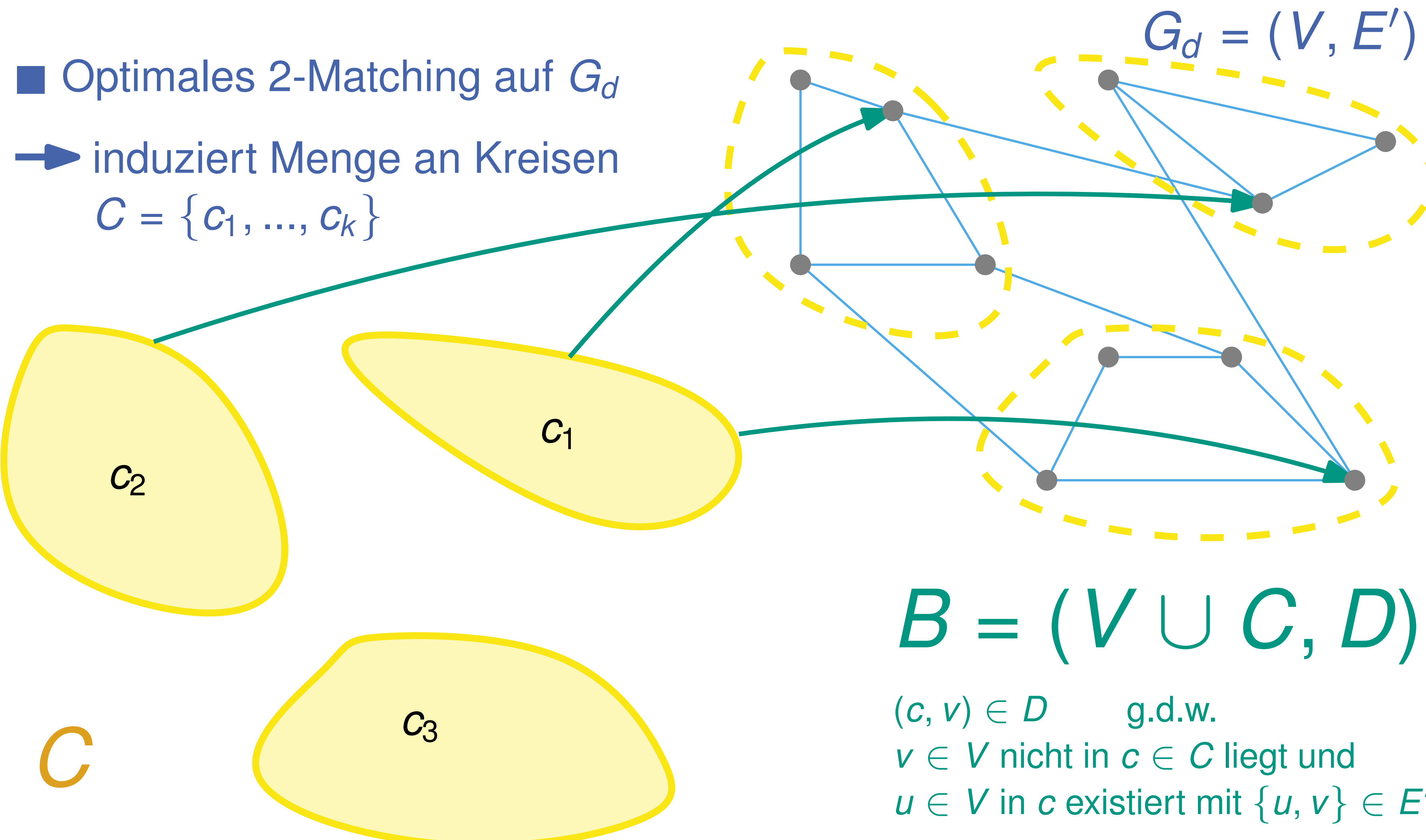
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$



$$B = (V \cup C, D)$$

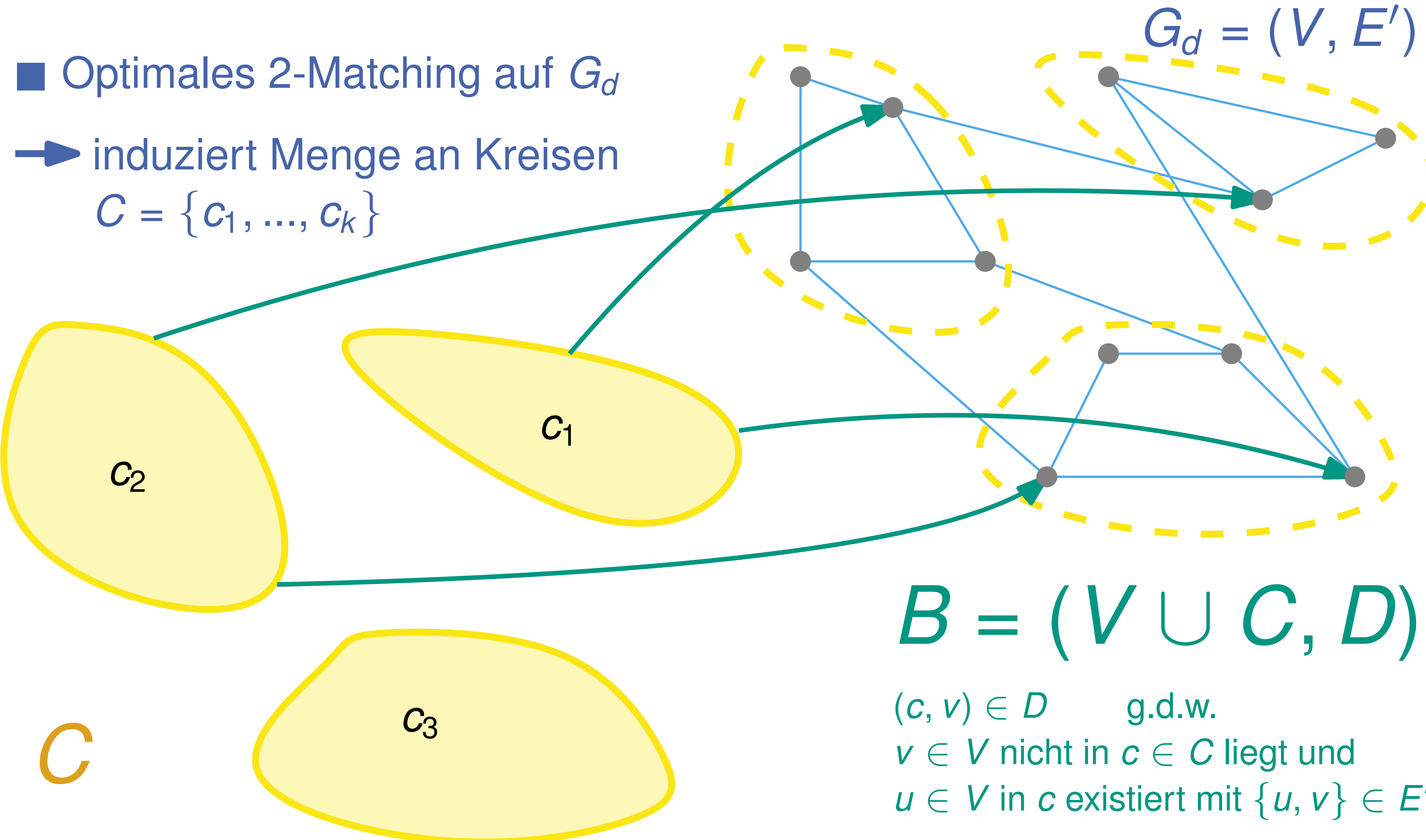
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$



$B = (V \cup C, D)$

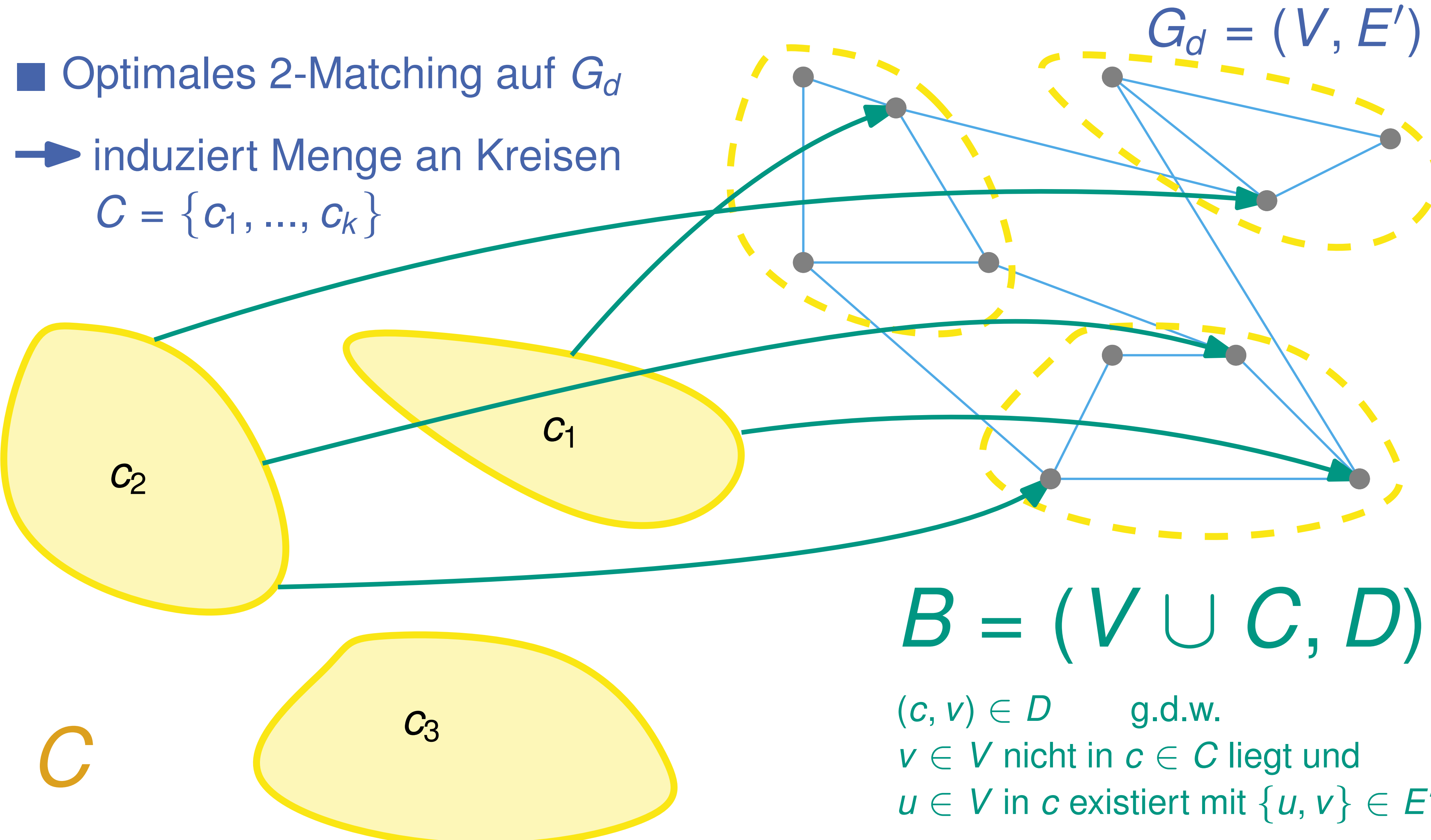
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$



$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

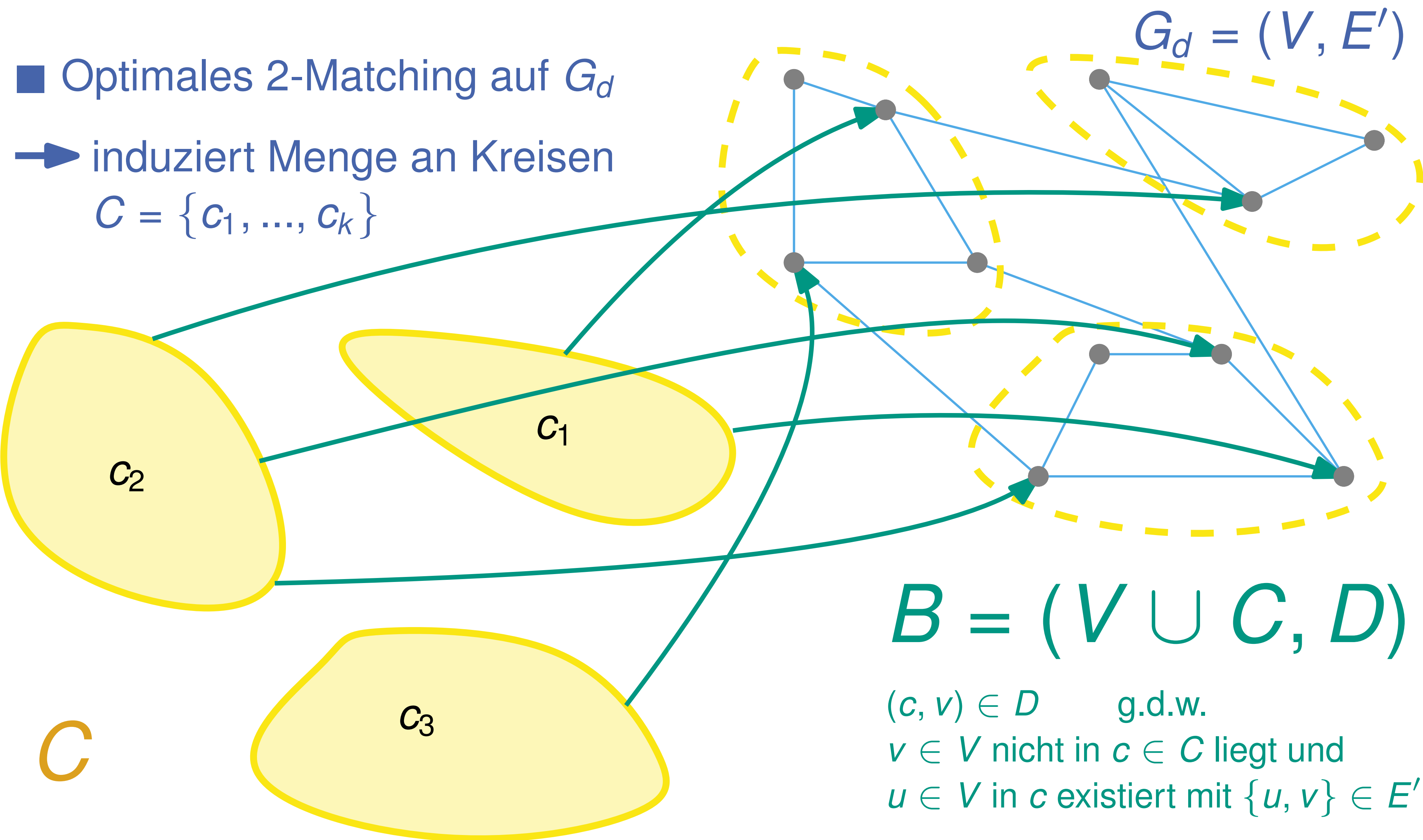
Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$



$B = (V \cup C, D)$

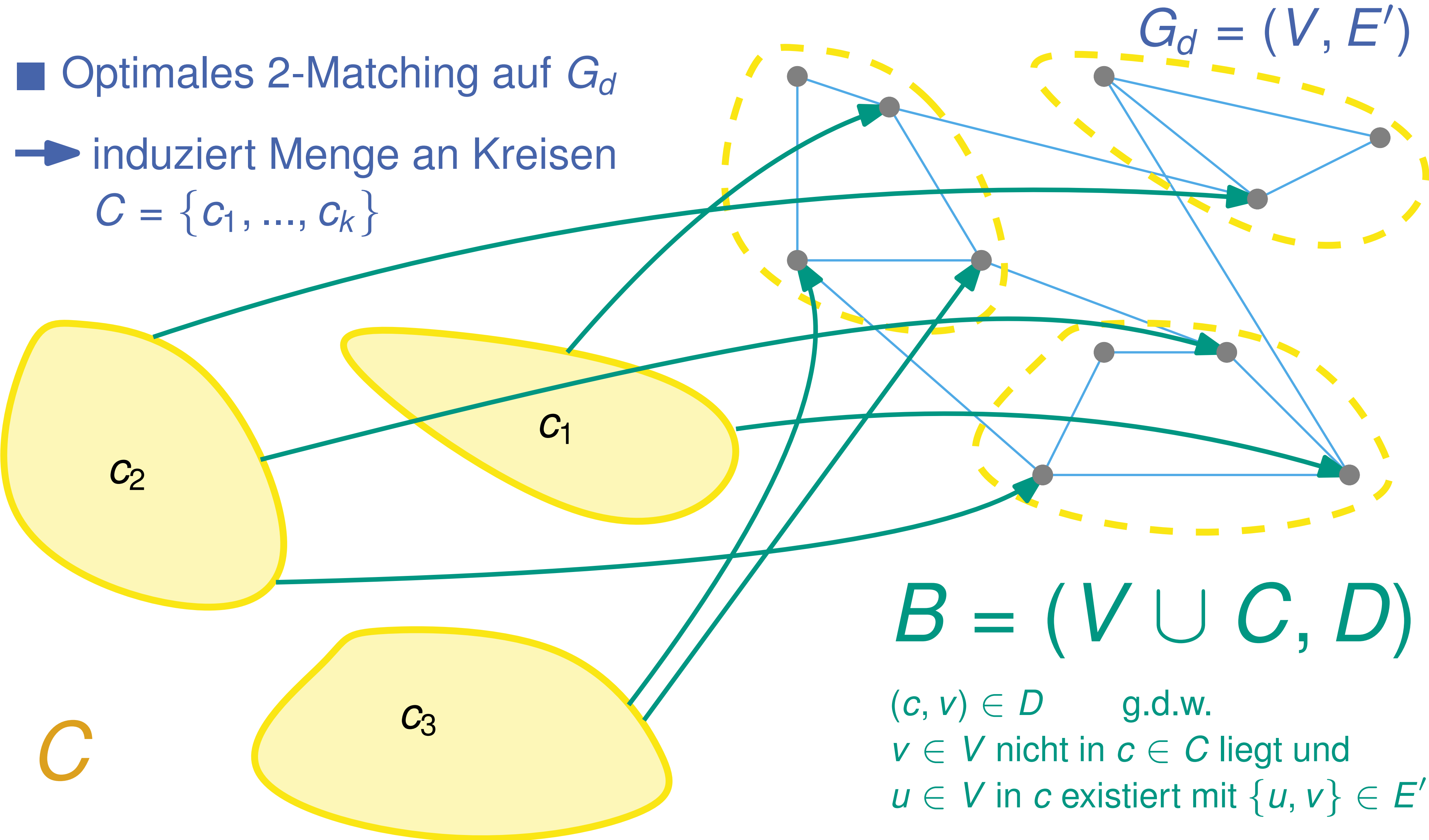
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen $C = \{c_1, \dots, c_k\}$



$B = (V \cup C, D)$

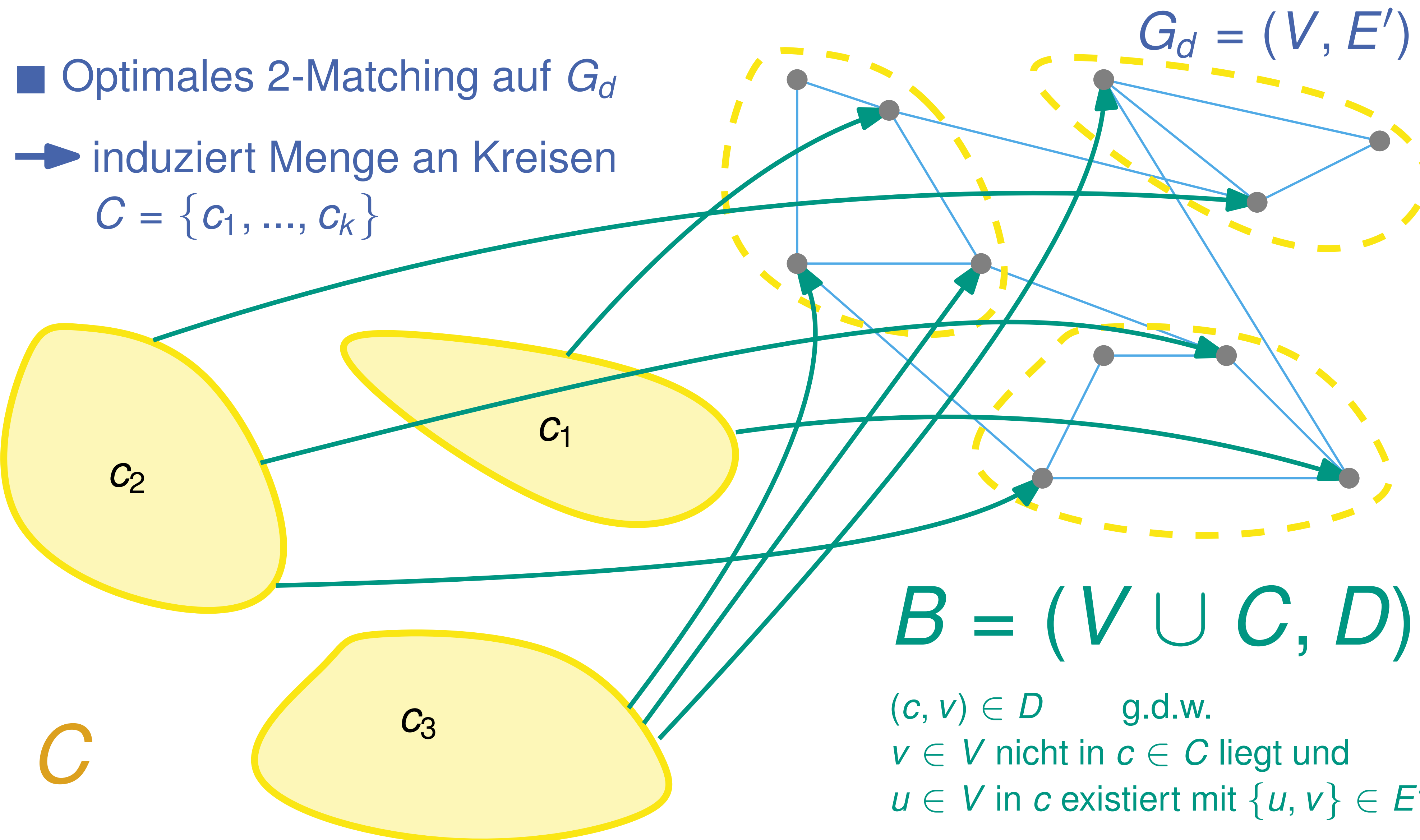
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Optimales 2-Matching auf G_d

→ induziert Menge an Kreisen
$C = \{c_1, \ldots, c_k\}$



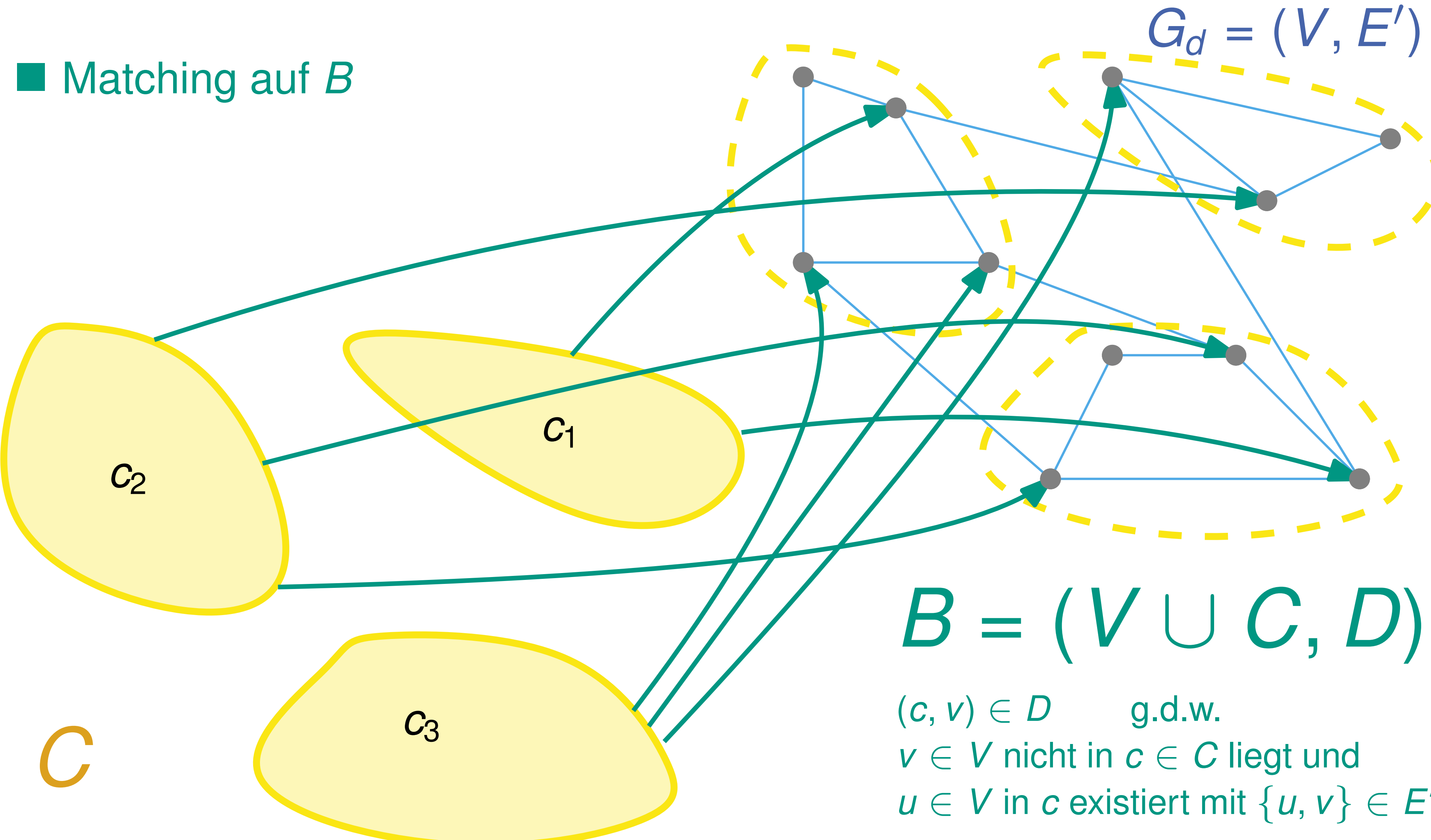
$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B



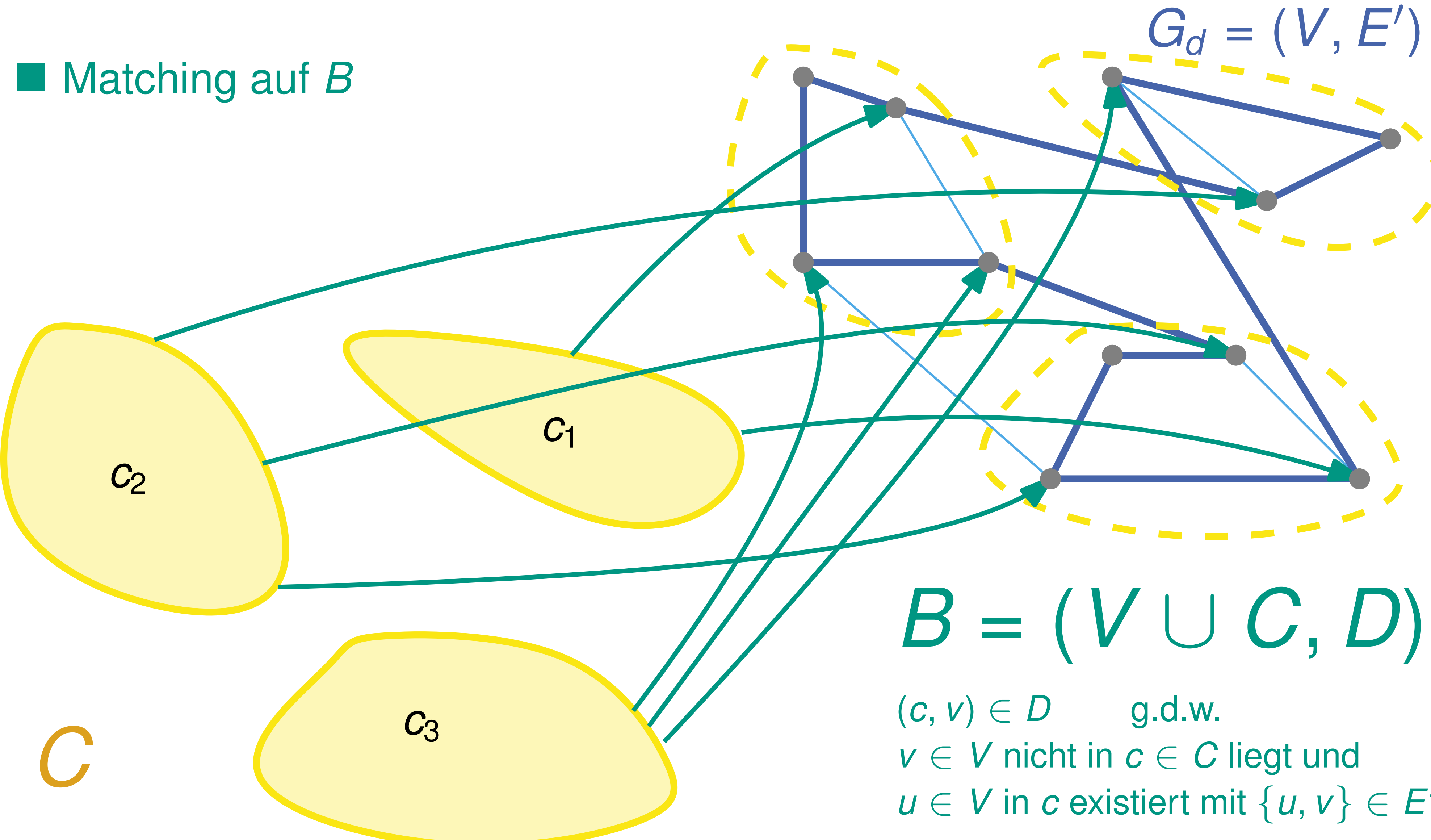
$B = (V \cup C, D)$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B



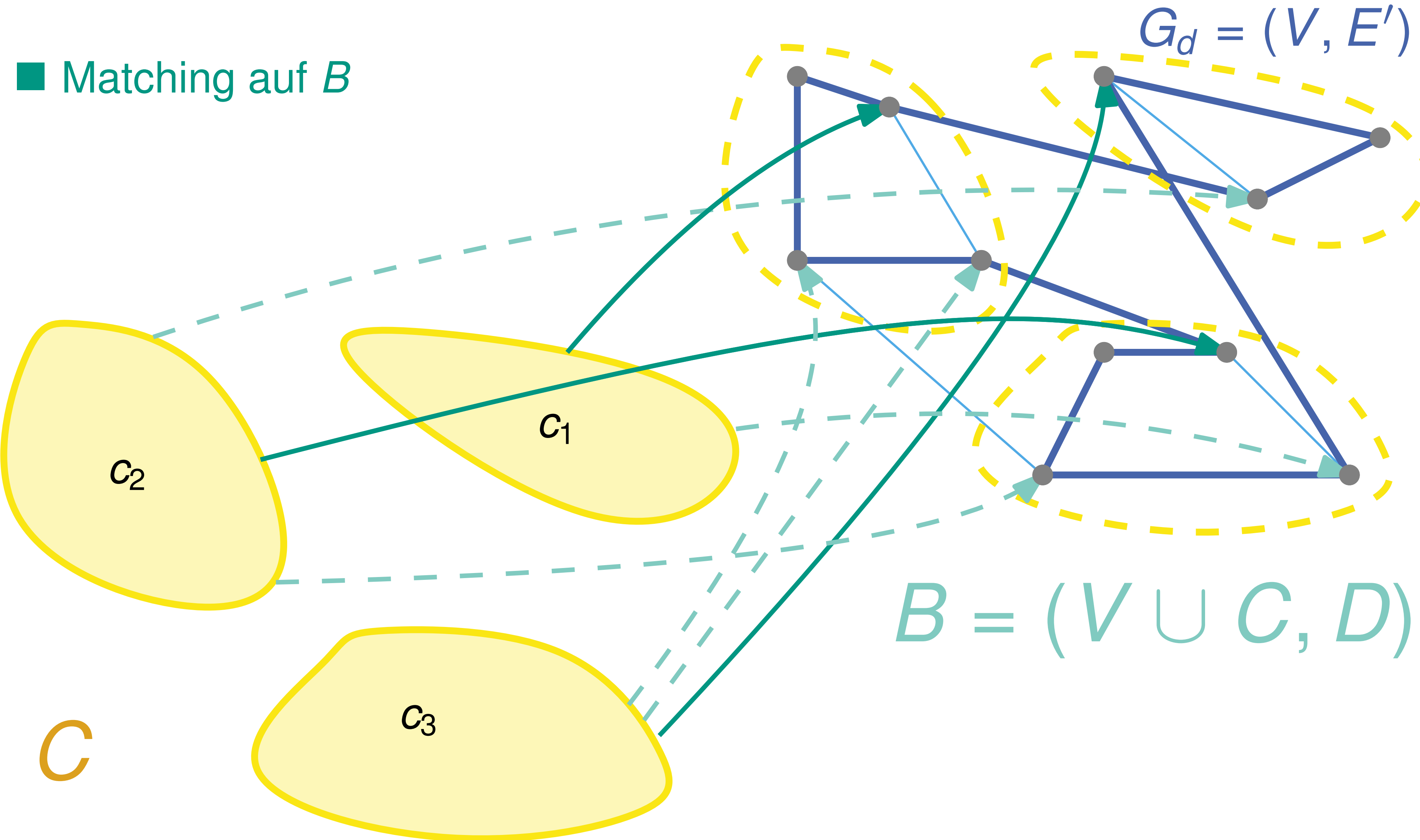
$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

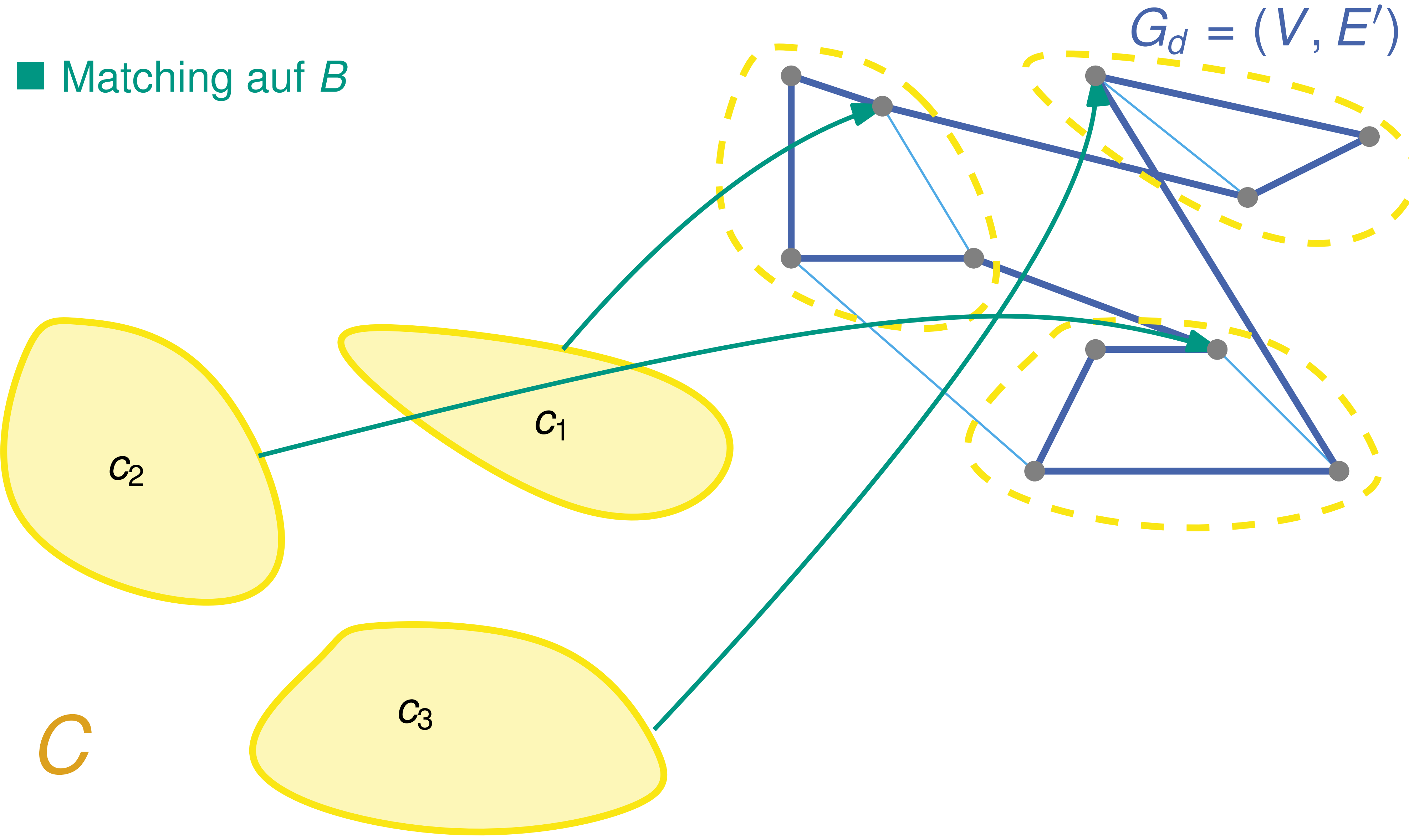
- Matching auf B



APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B

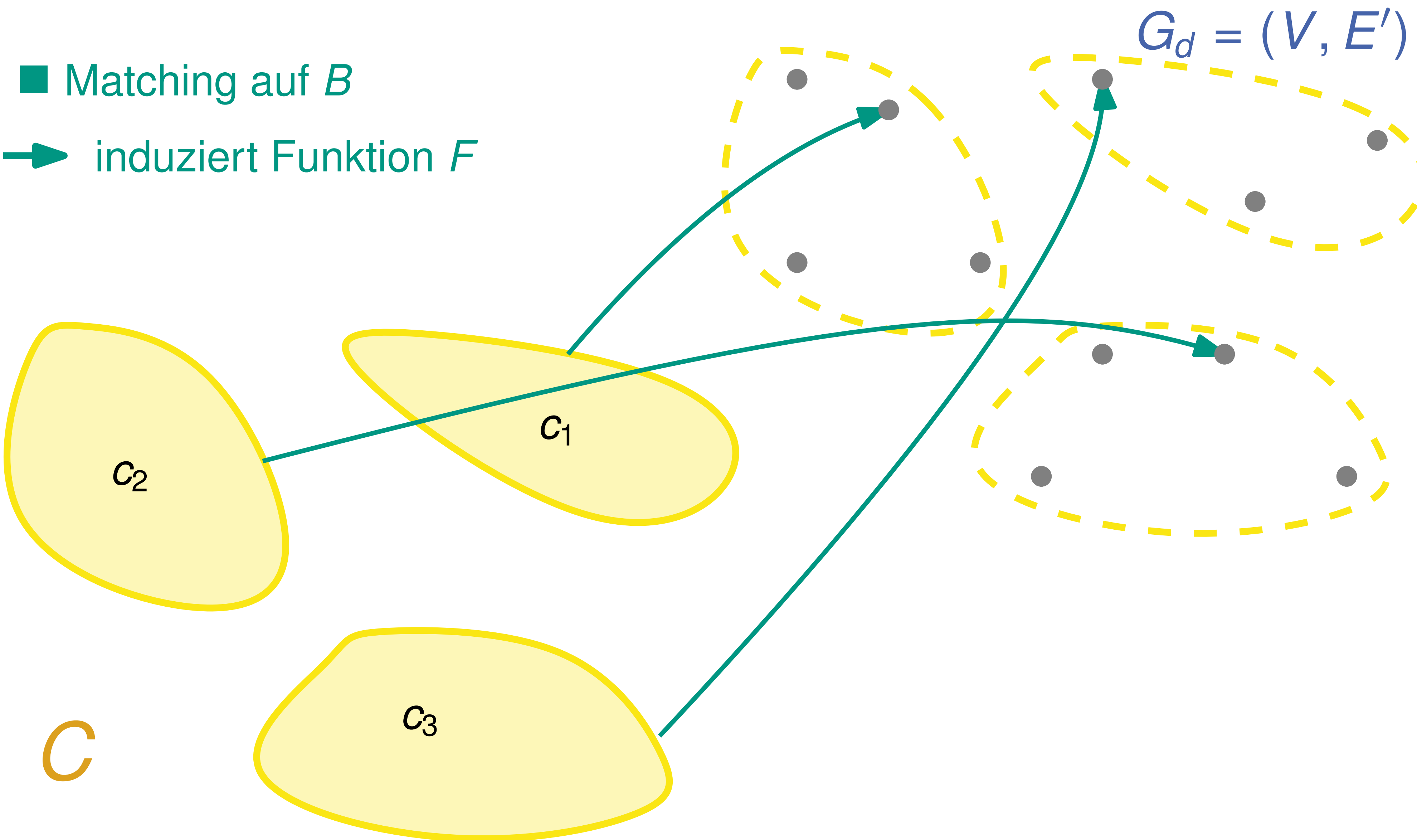


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B
- → induziert Funktion F

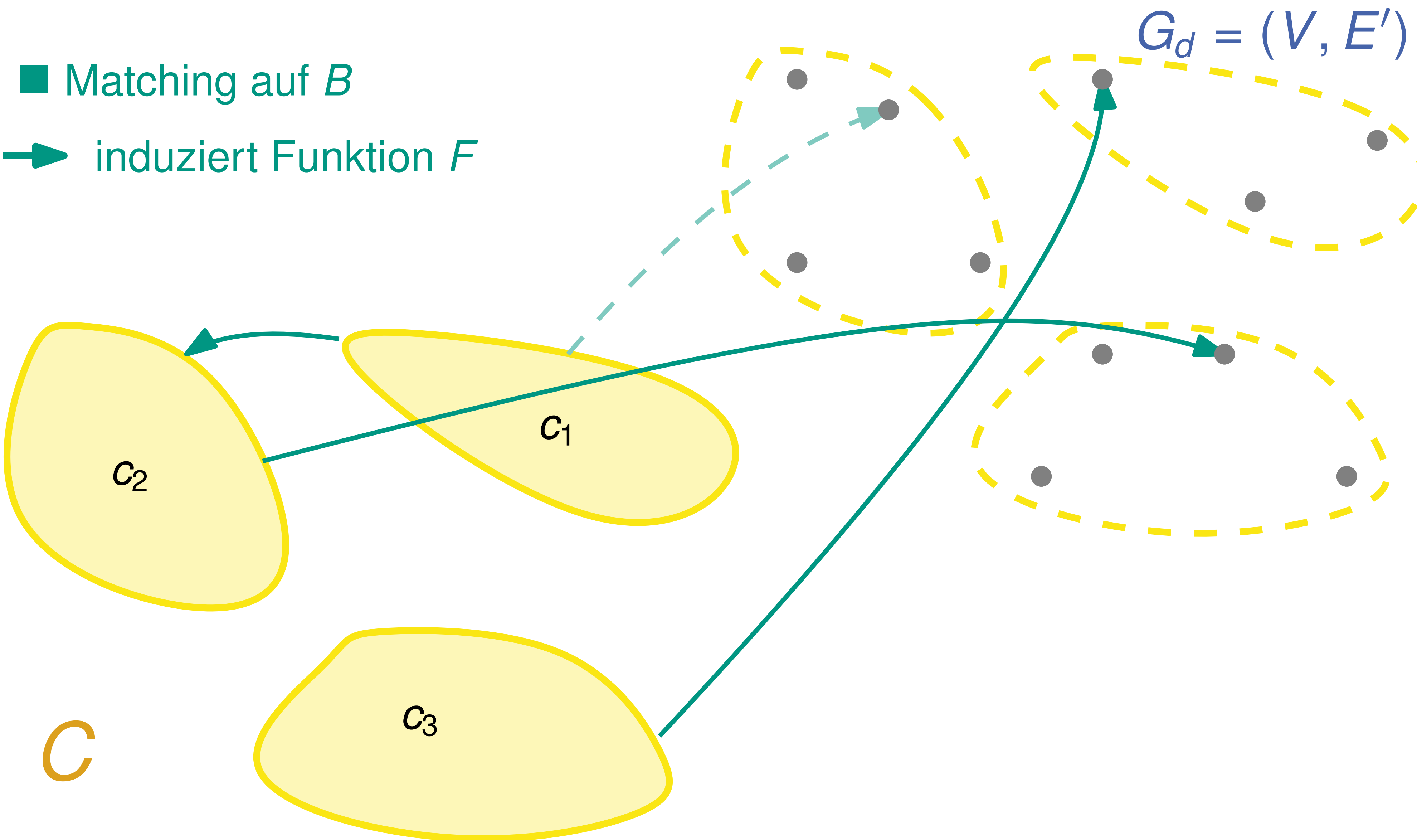


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B
- → induziert Funktion F

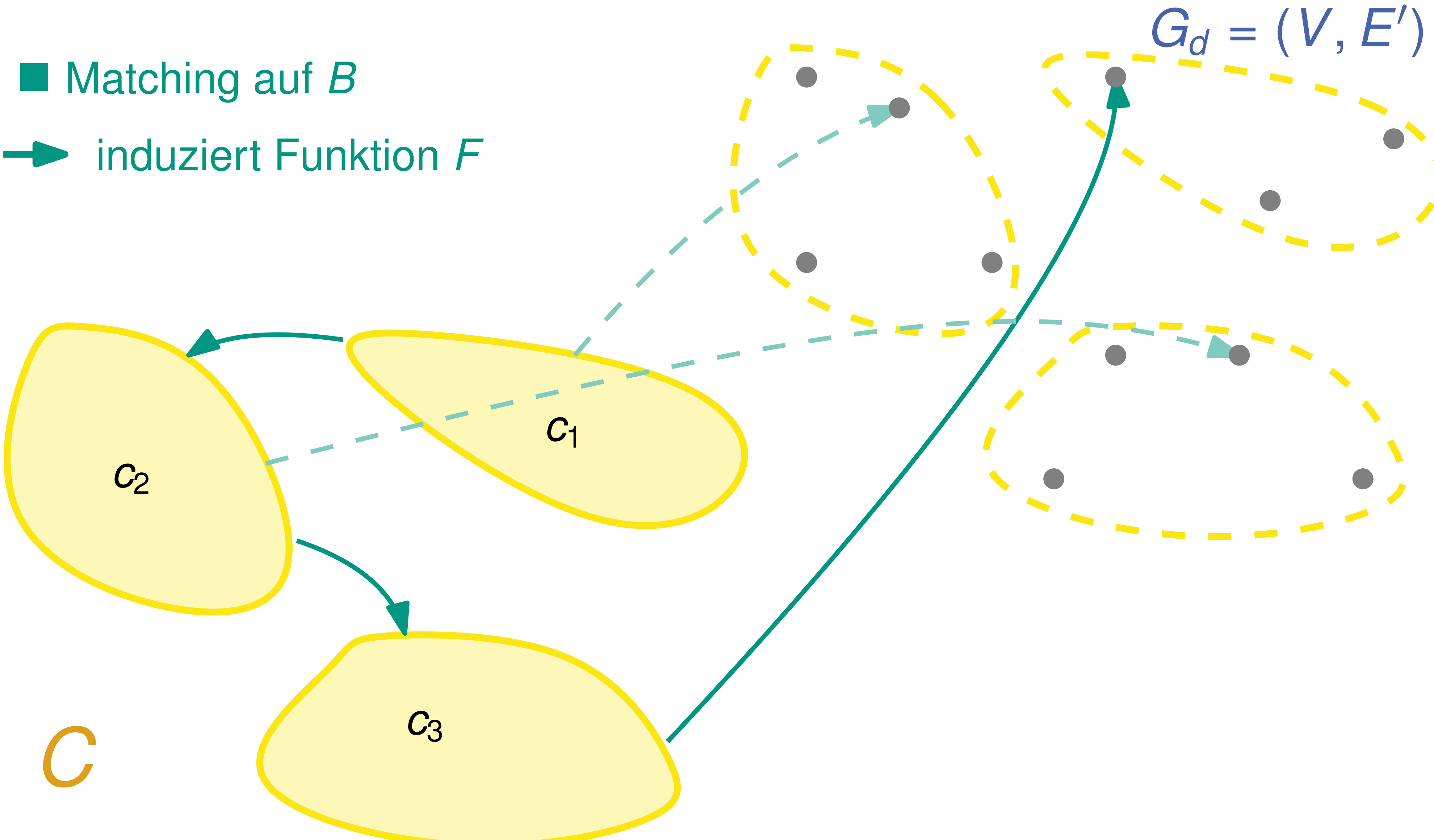


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

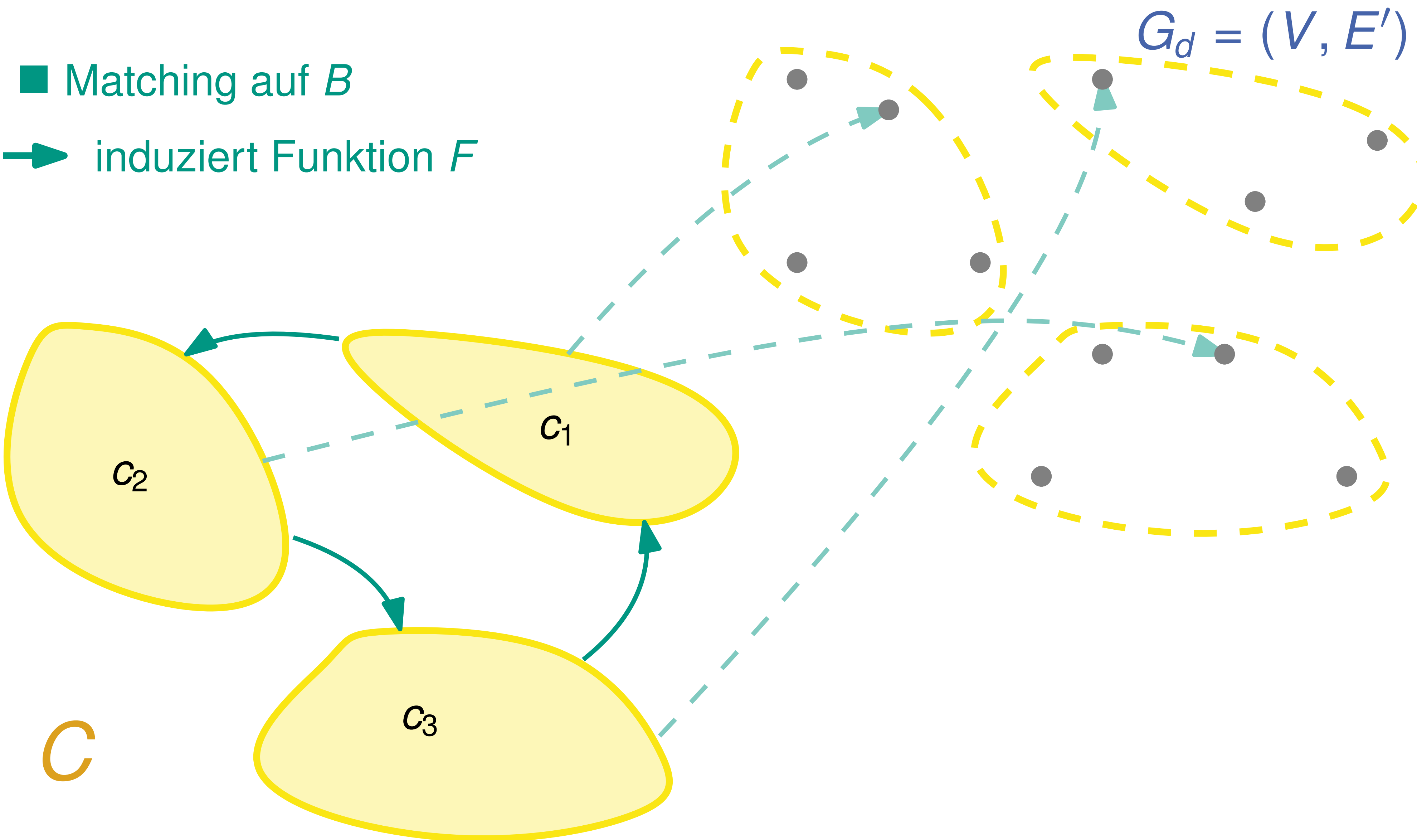
- Matching auf B
- → induziert Funktion F



APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B
- → induziert Funktion F



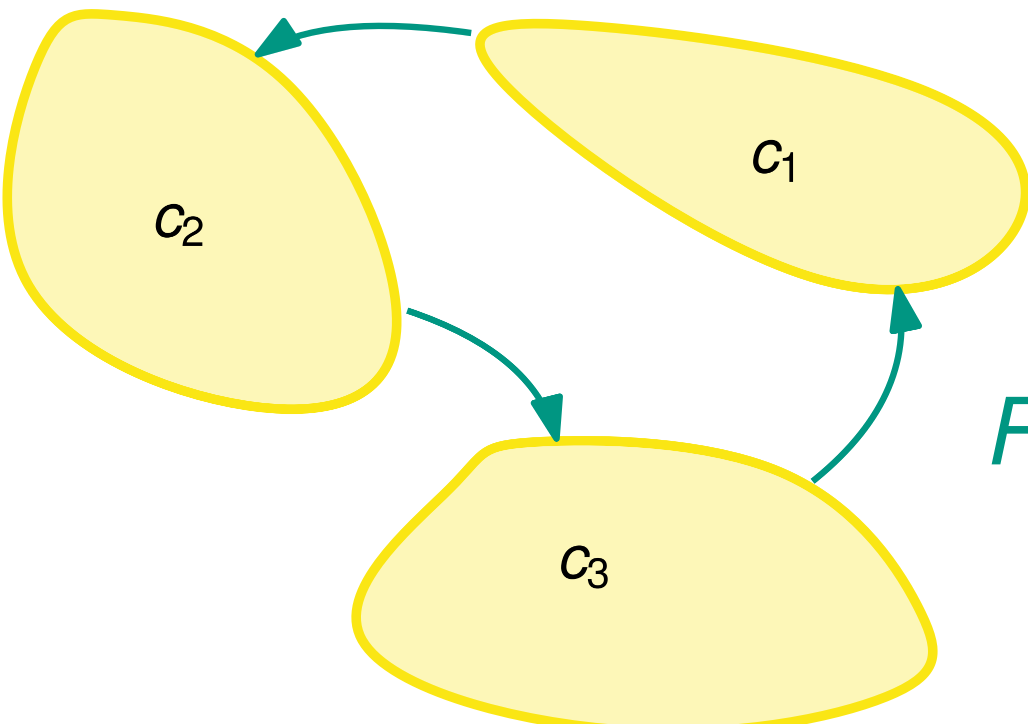
Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

- Matching auf B

→ induziert Funktion F

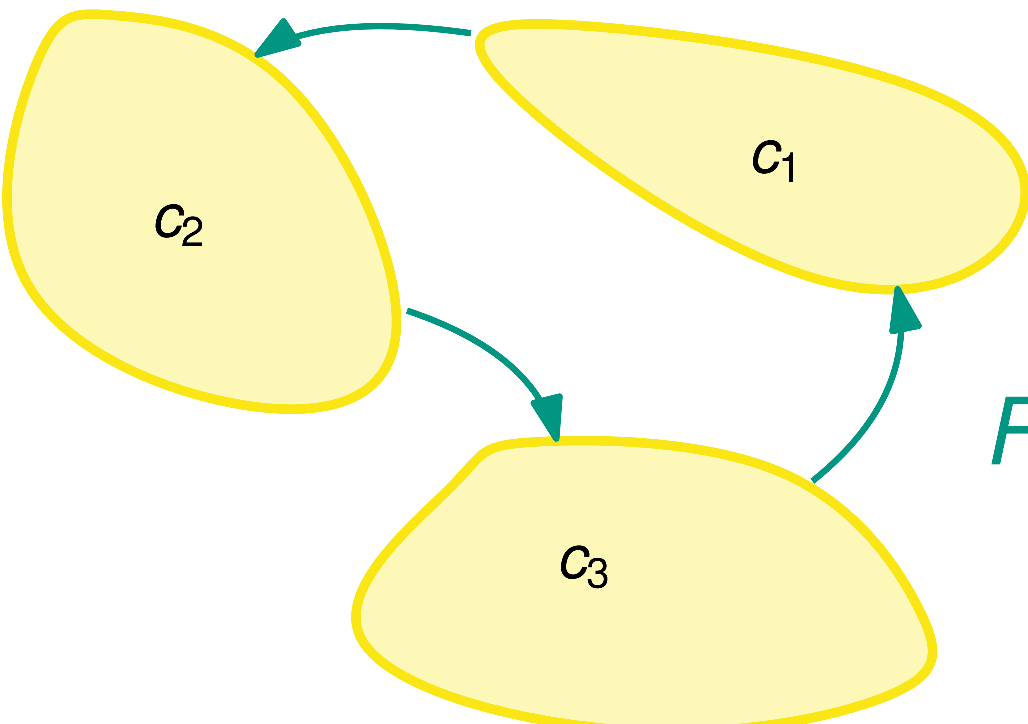


$F = (C, A)$

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Jede Kante von F ist mit einer Kante von G_d assoziiert!



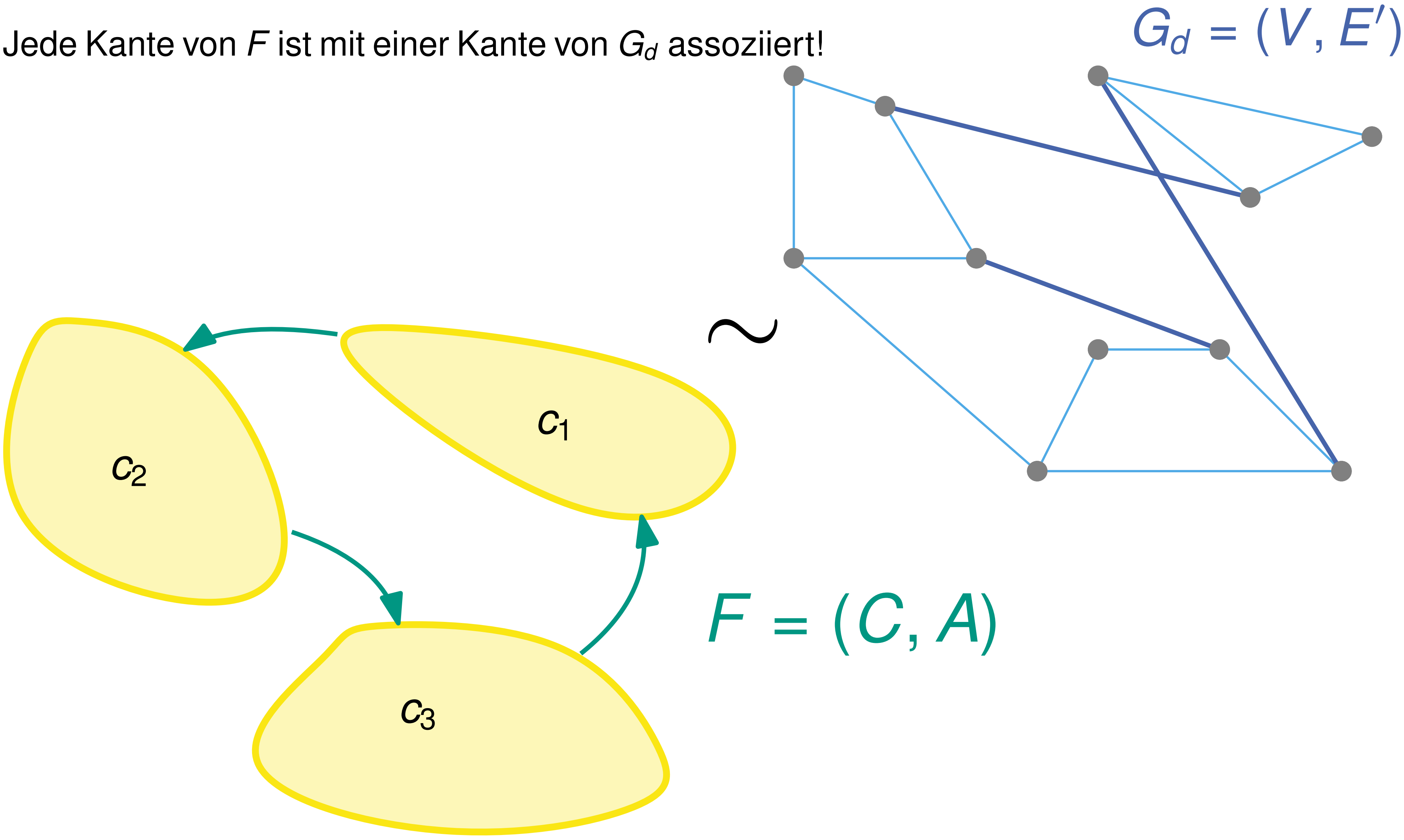
$F = (C, A)$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Jede Kante von F ist mit einer Kante von G_d assoziiert!

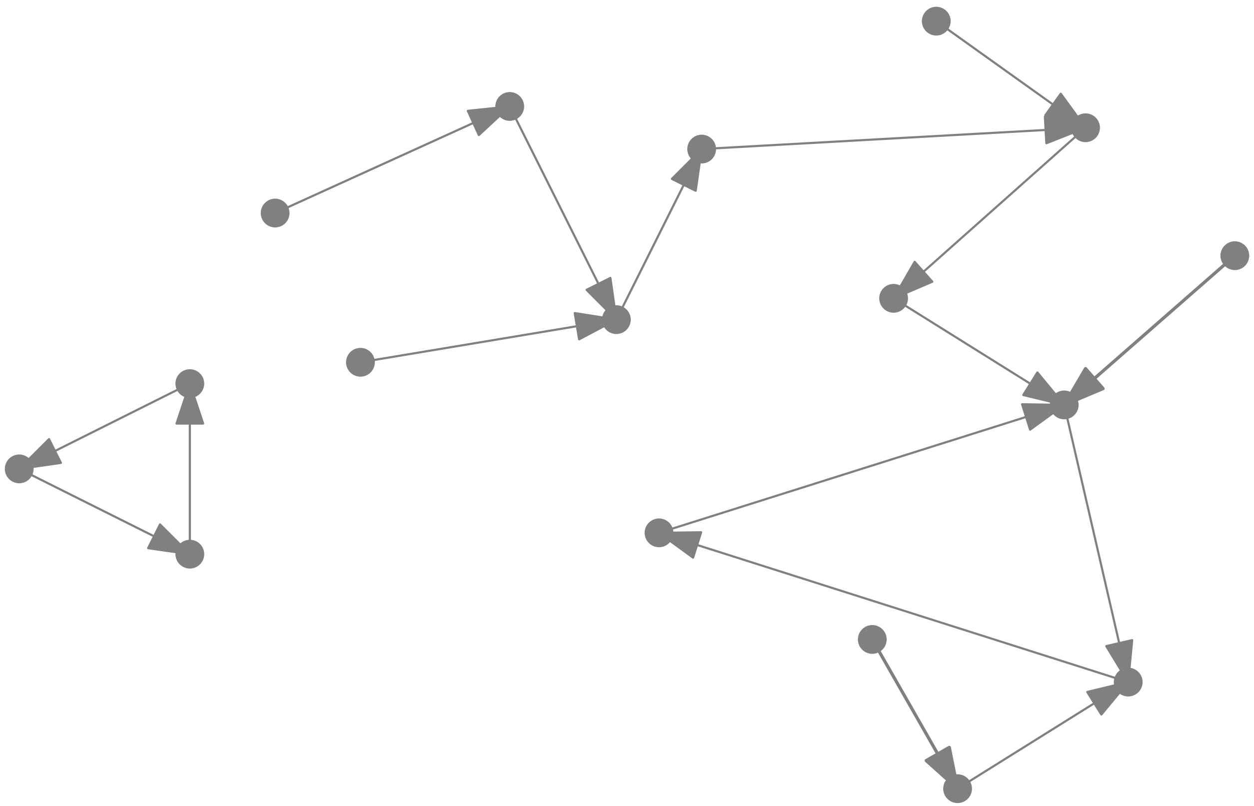


APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.



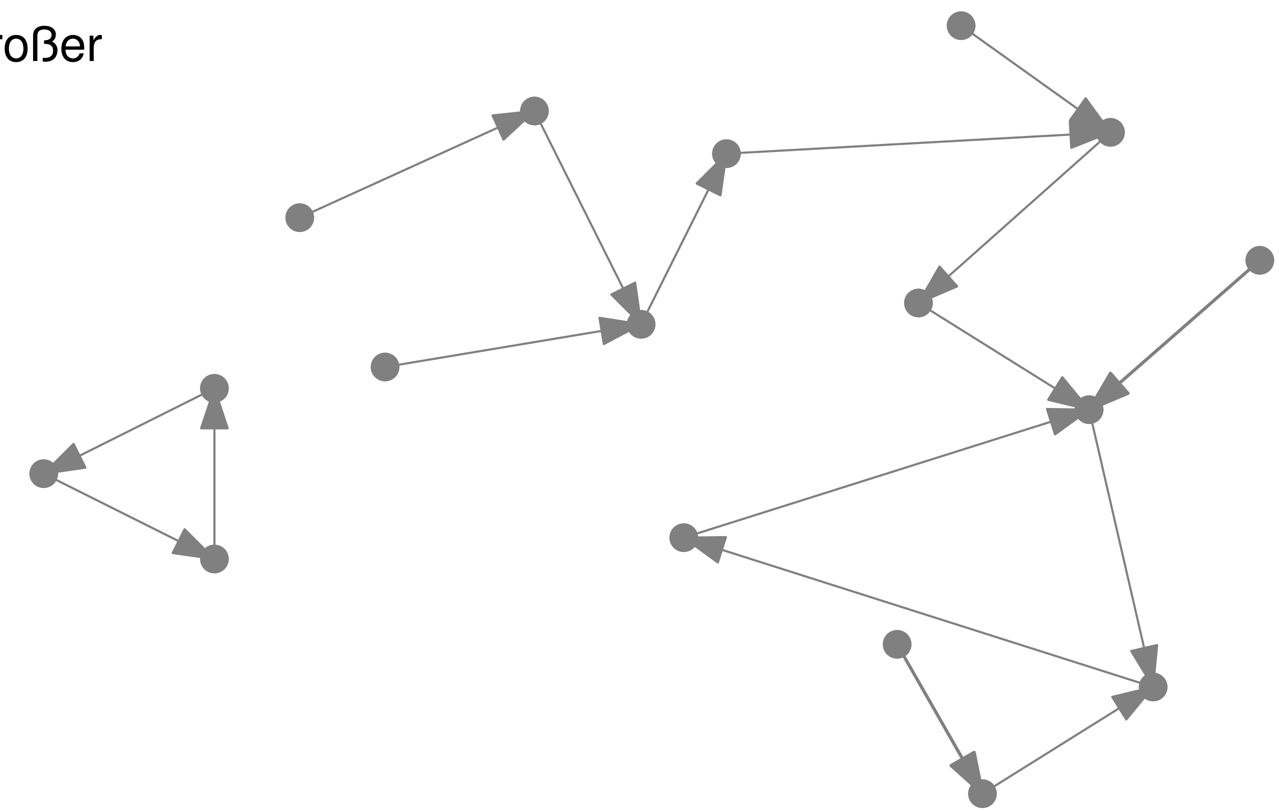
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten



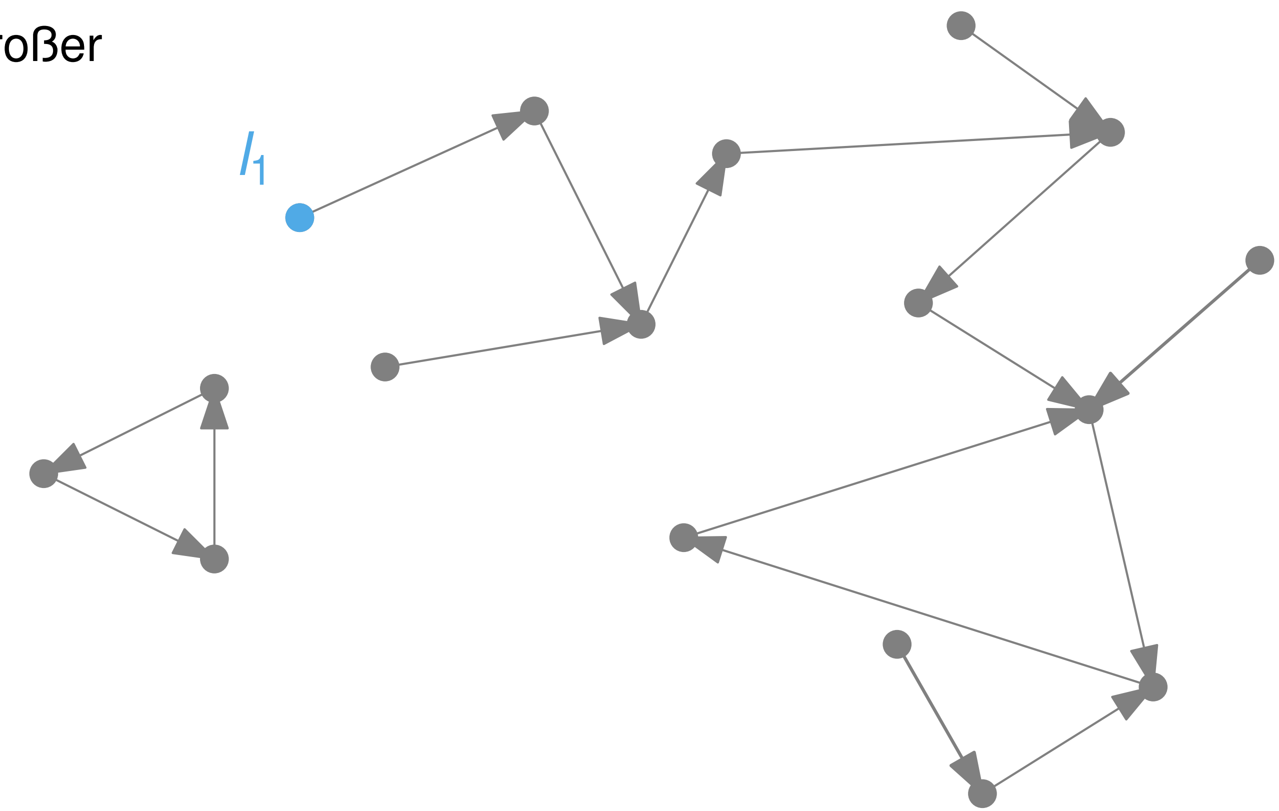
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten



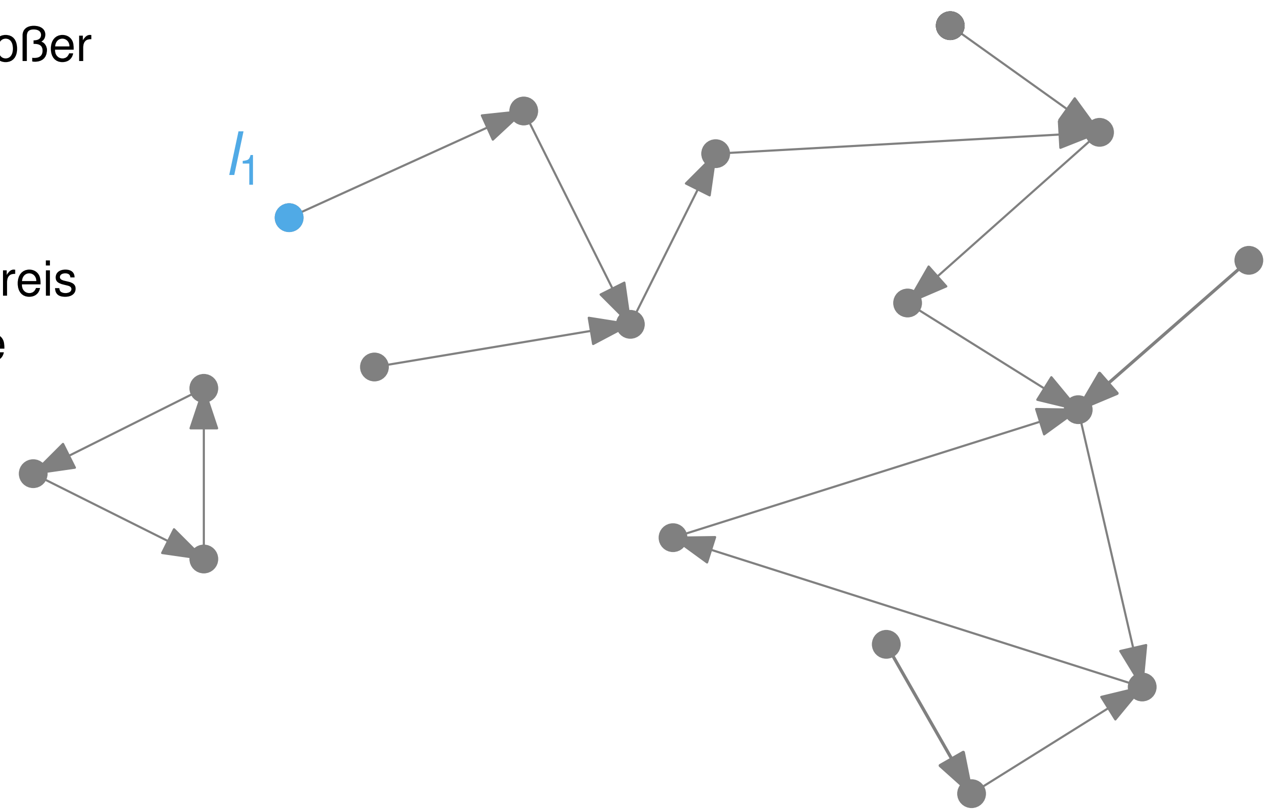
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



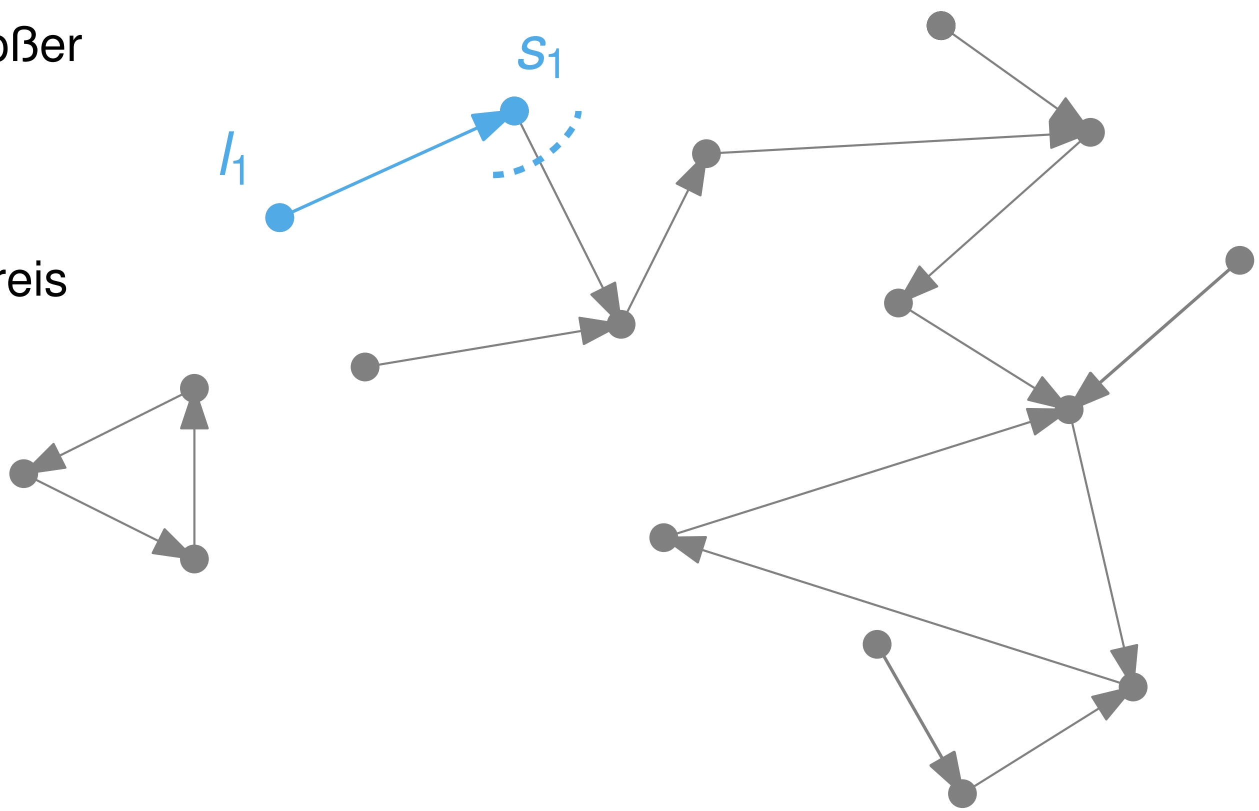
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



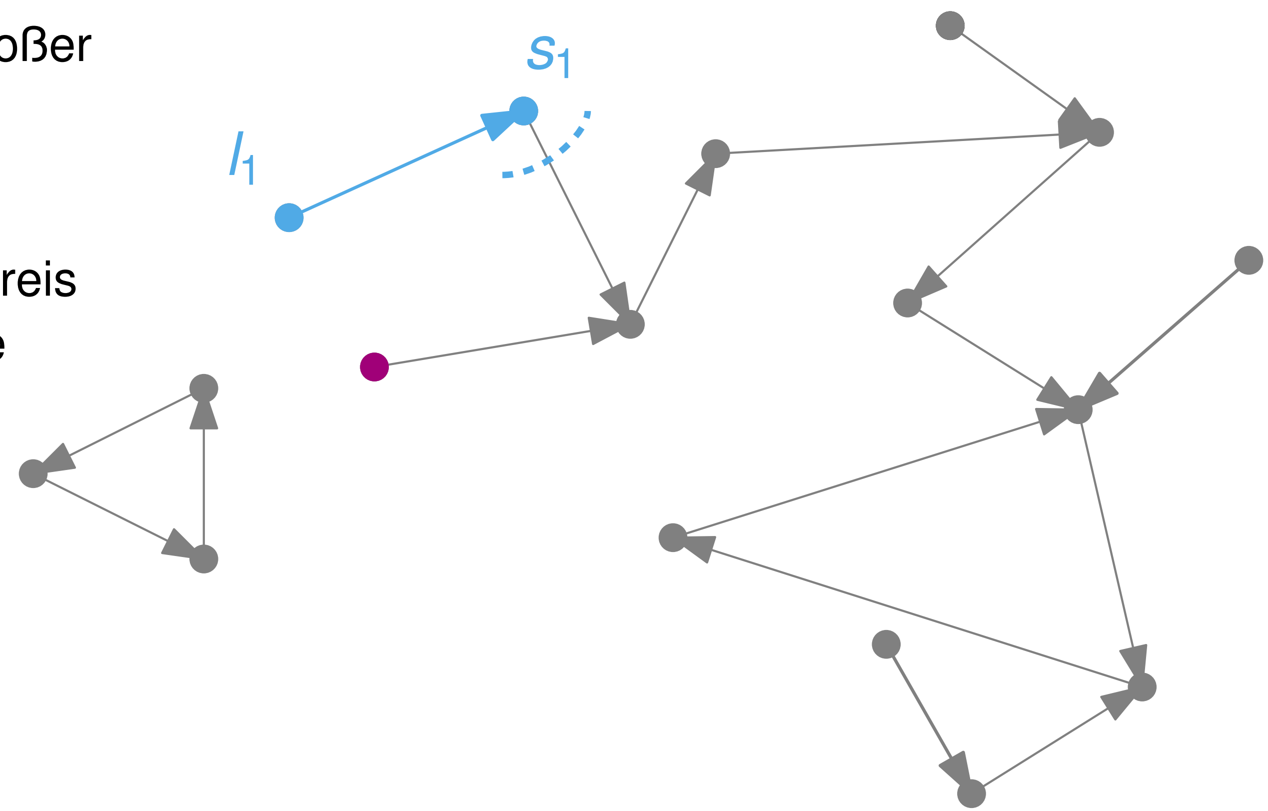
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



Institute of Theoretical Informatics
Algorithmics Group

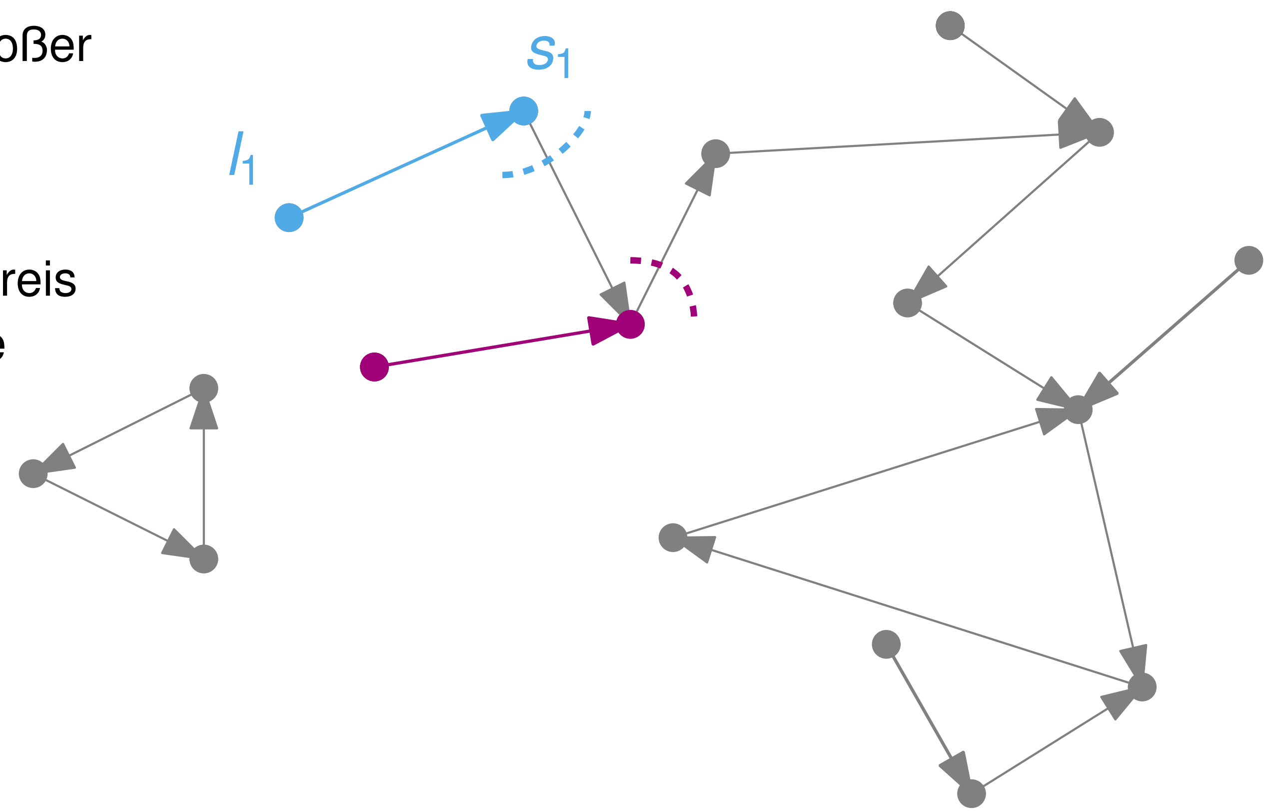
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



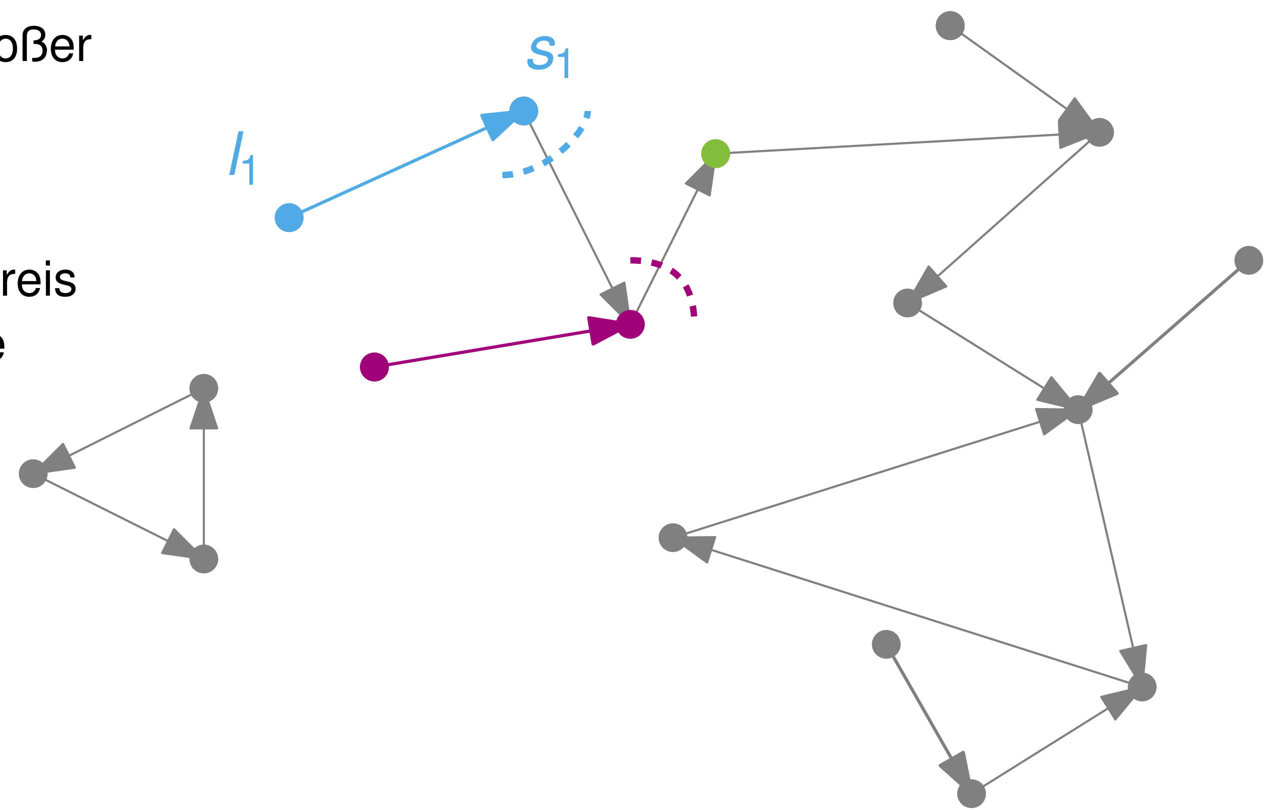
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



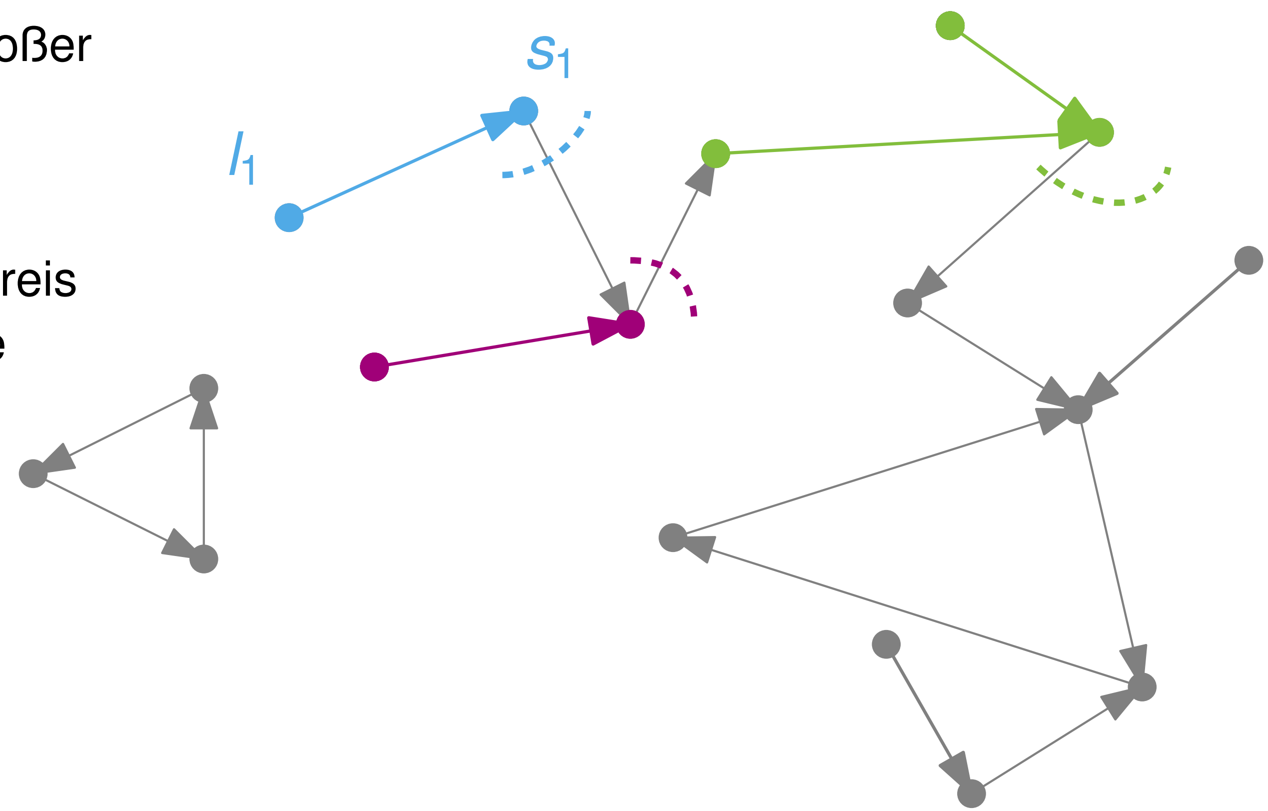
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



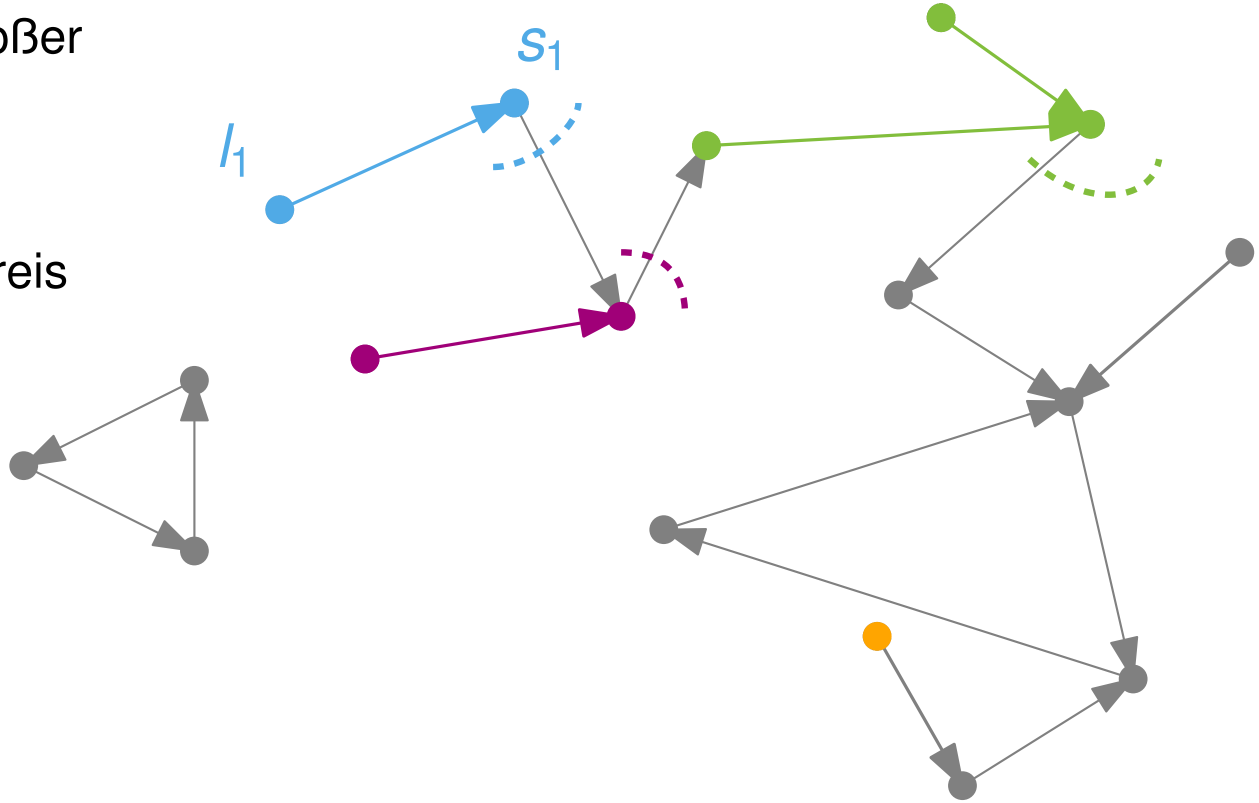
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



Institute of Theoretical Informatics
Algorithmics Group

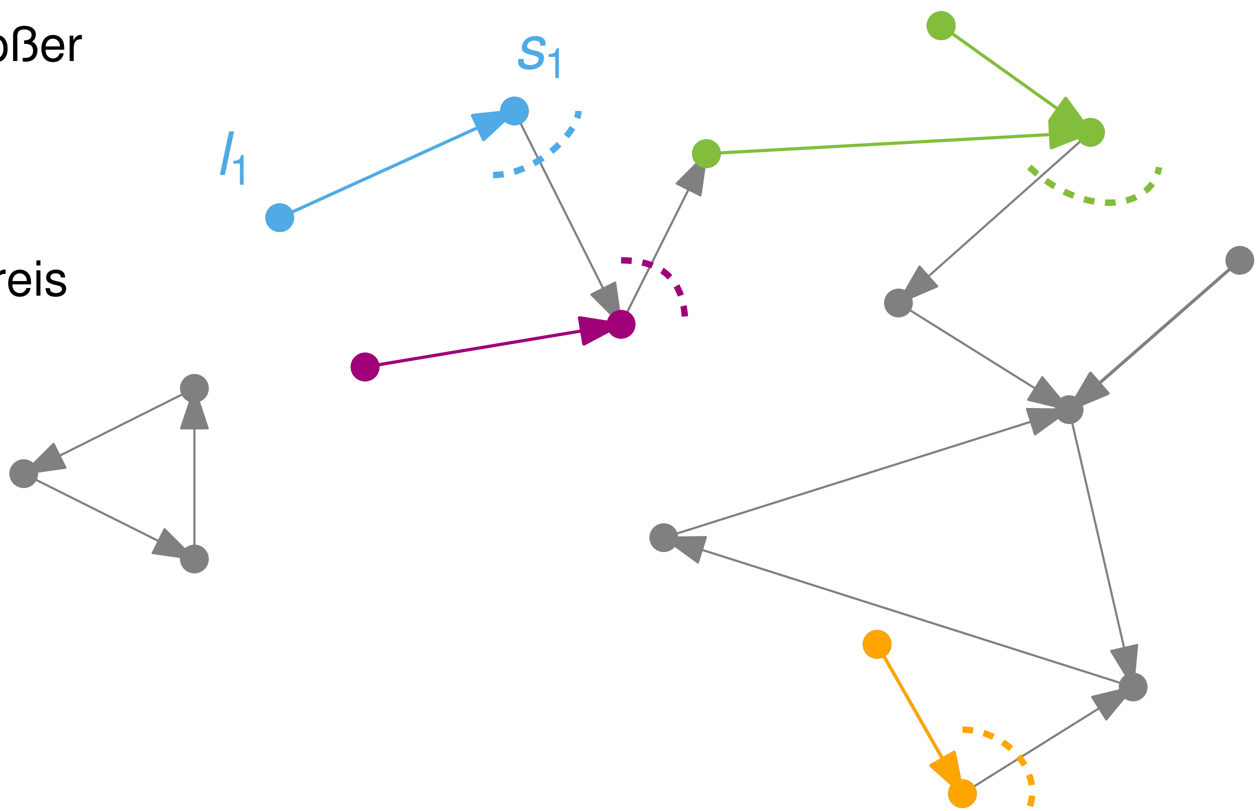
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



Institute of Theoretical Informatics
Algorithmics Group

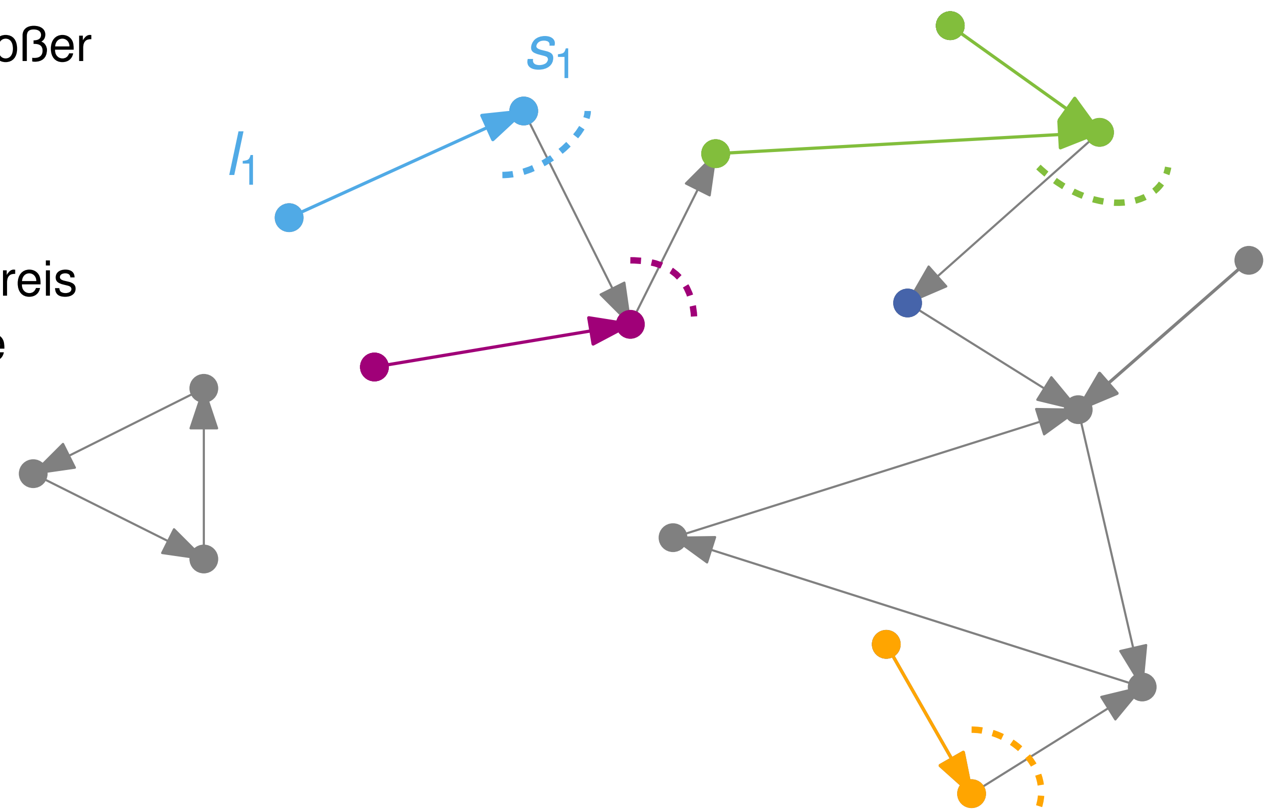
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



Institute of Theoretical Informatics
Algorithmics Group

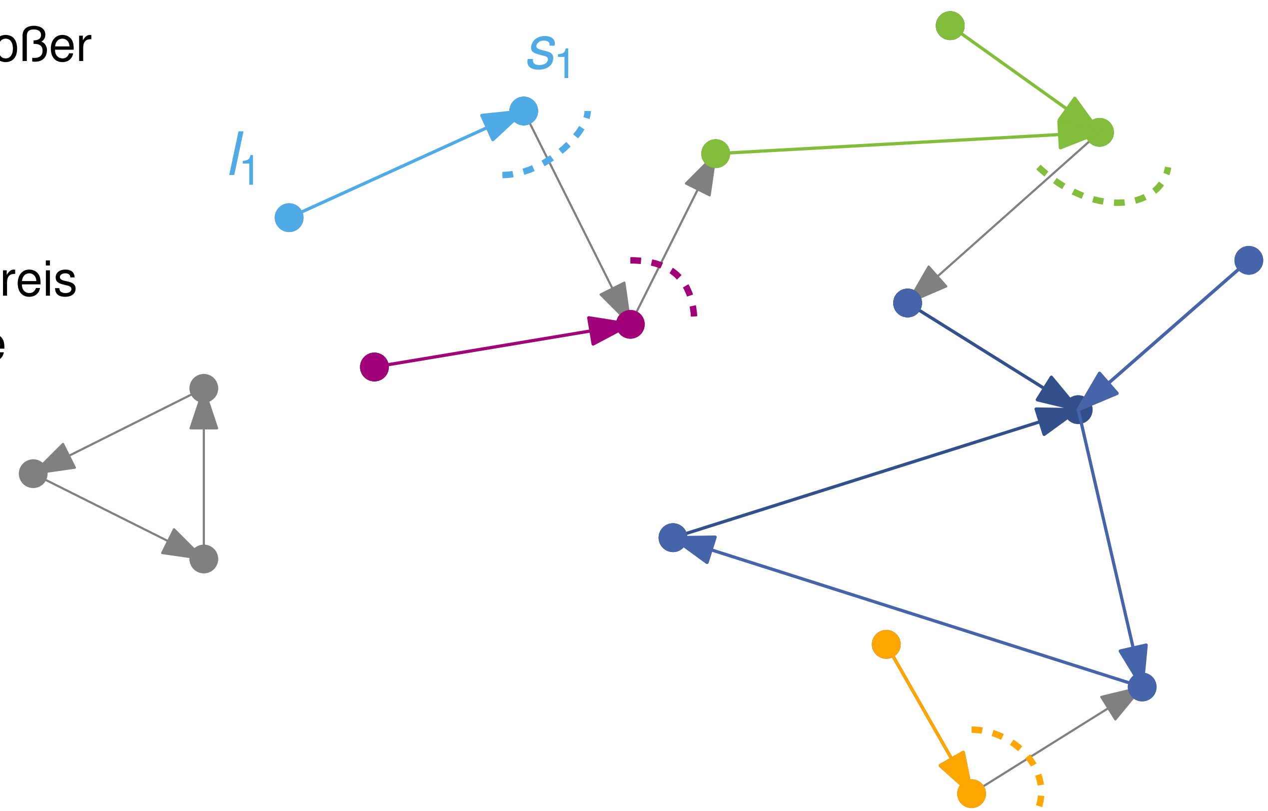
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1



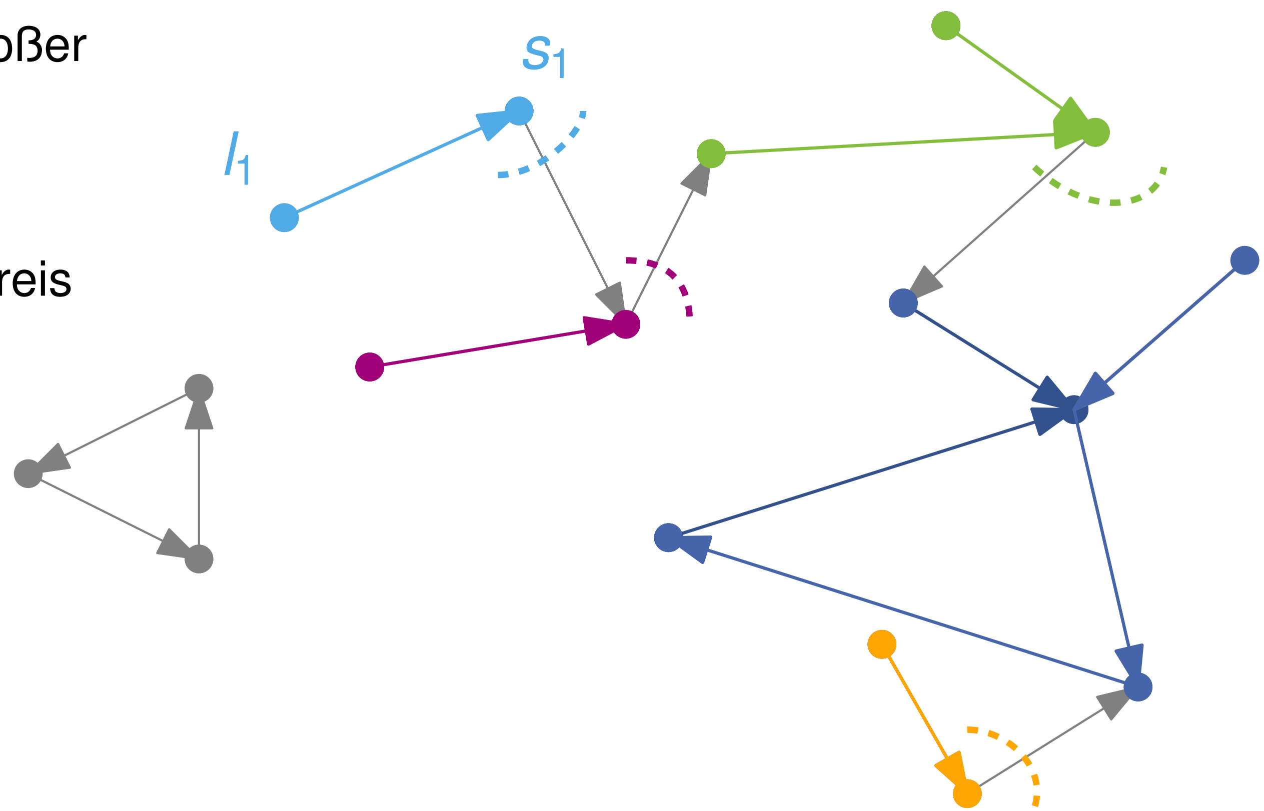
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1
- Kreise werden in Pfade der Länge 1 und 2 zerlegt



Institute of Theoretical Informatics
Algorithmics Group

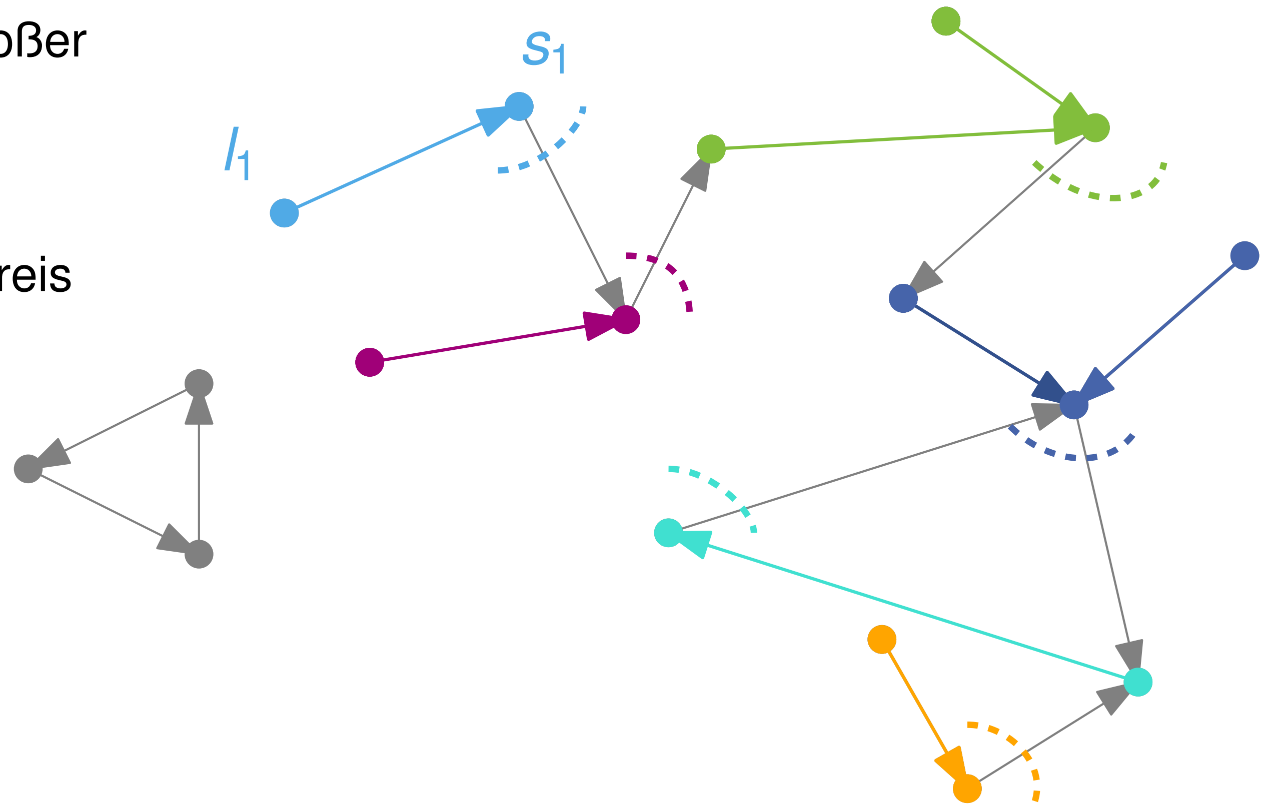
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1
- Kreise werden in Pfade der Länge 1 und 2 zerlegt



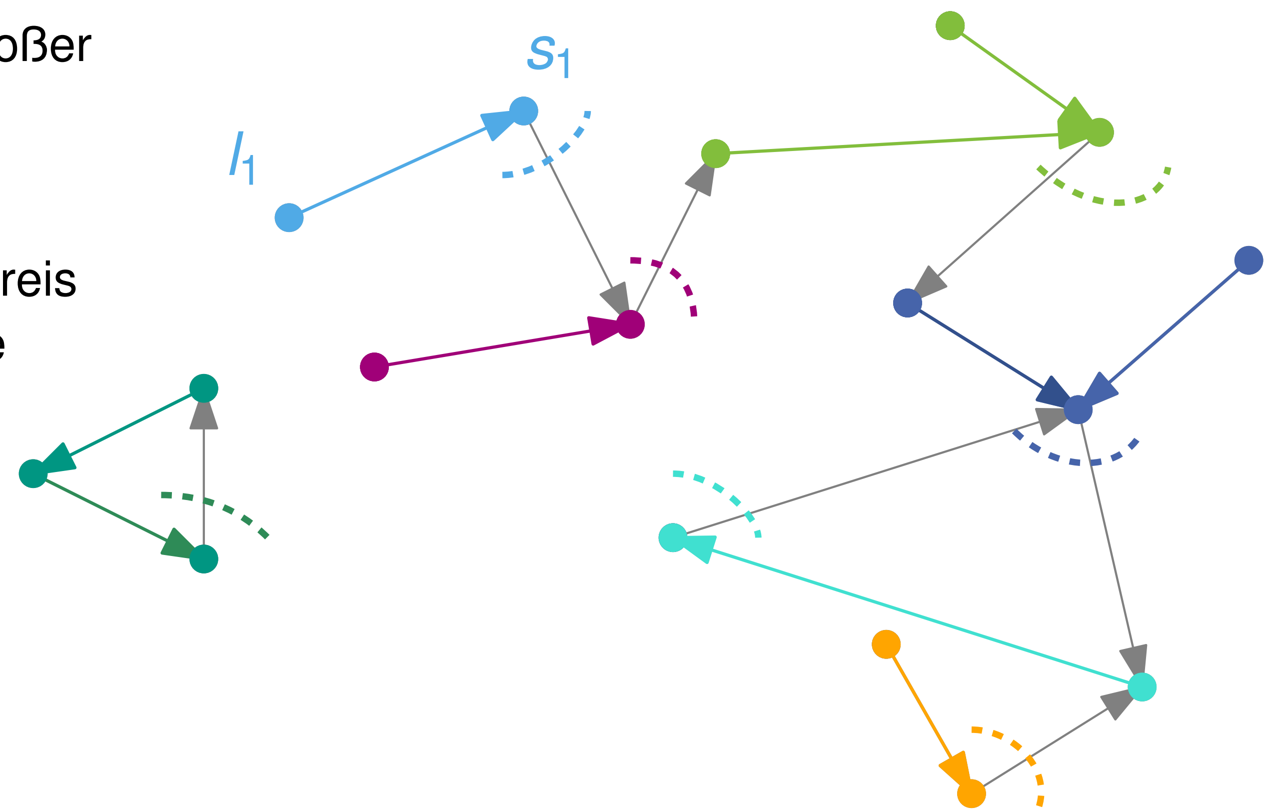
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

- Wähle Blatt mit möglichst großer Distanz zu anderem Knoten
- Nachfolger liegt auf einem Kreis oder ist Wurzel eines In-Tree der Höhe 1
- Kreise werden in Pfade der Länge 1 und 2 zerlegt



APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

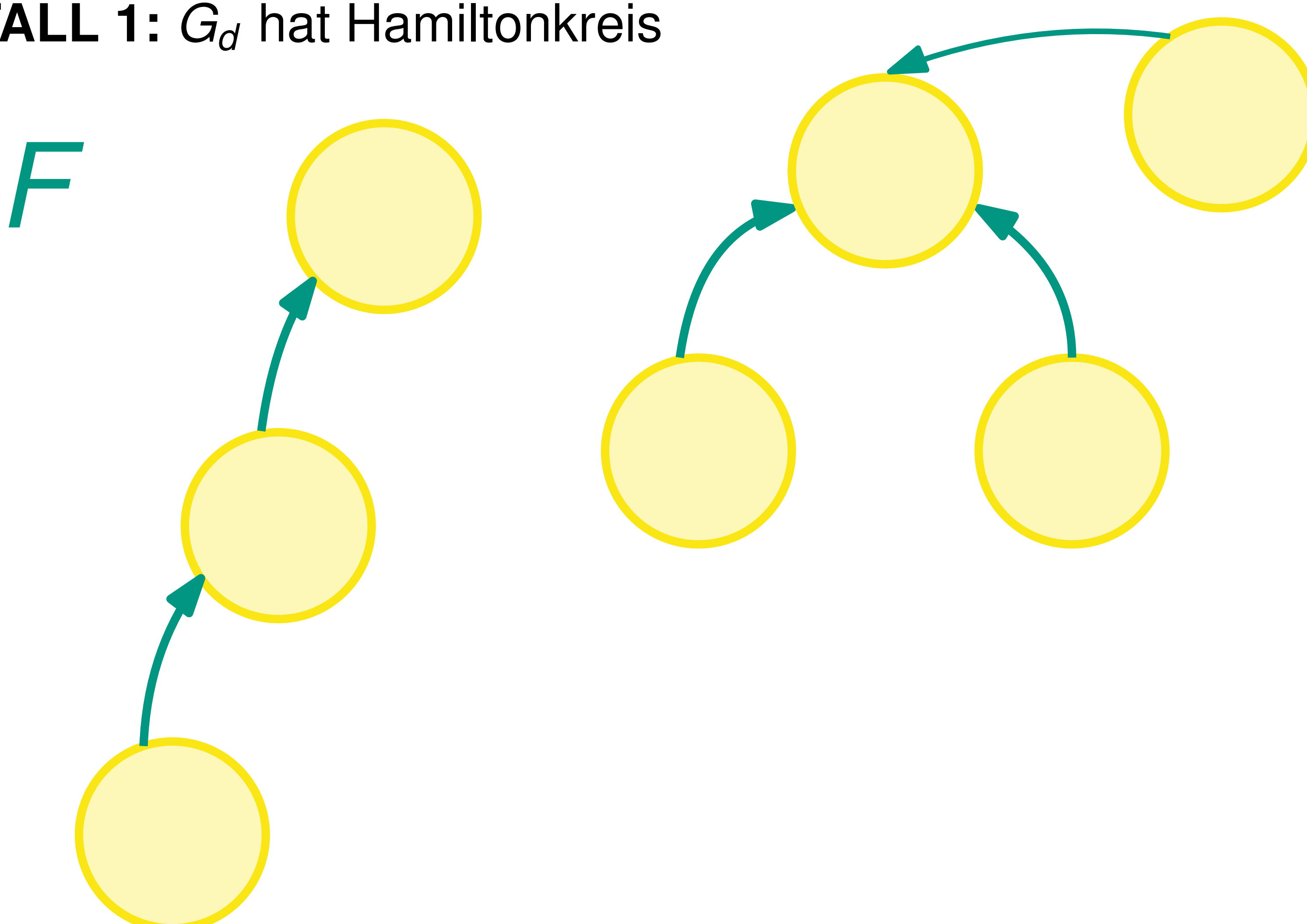
Lemma:

Jede Funktion F hat einen aufspannenden Teilgraph, der aus In-Trees der Höhe 1 und Pfaden der Länge 2 besteht.

Institute of Theoretical Informatics
Algorithmics Group

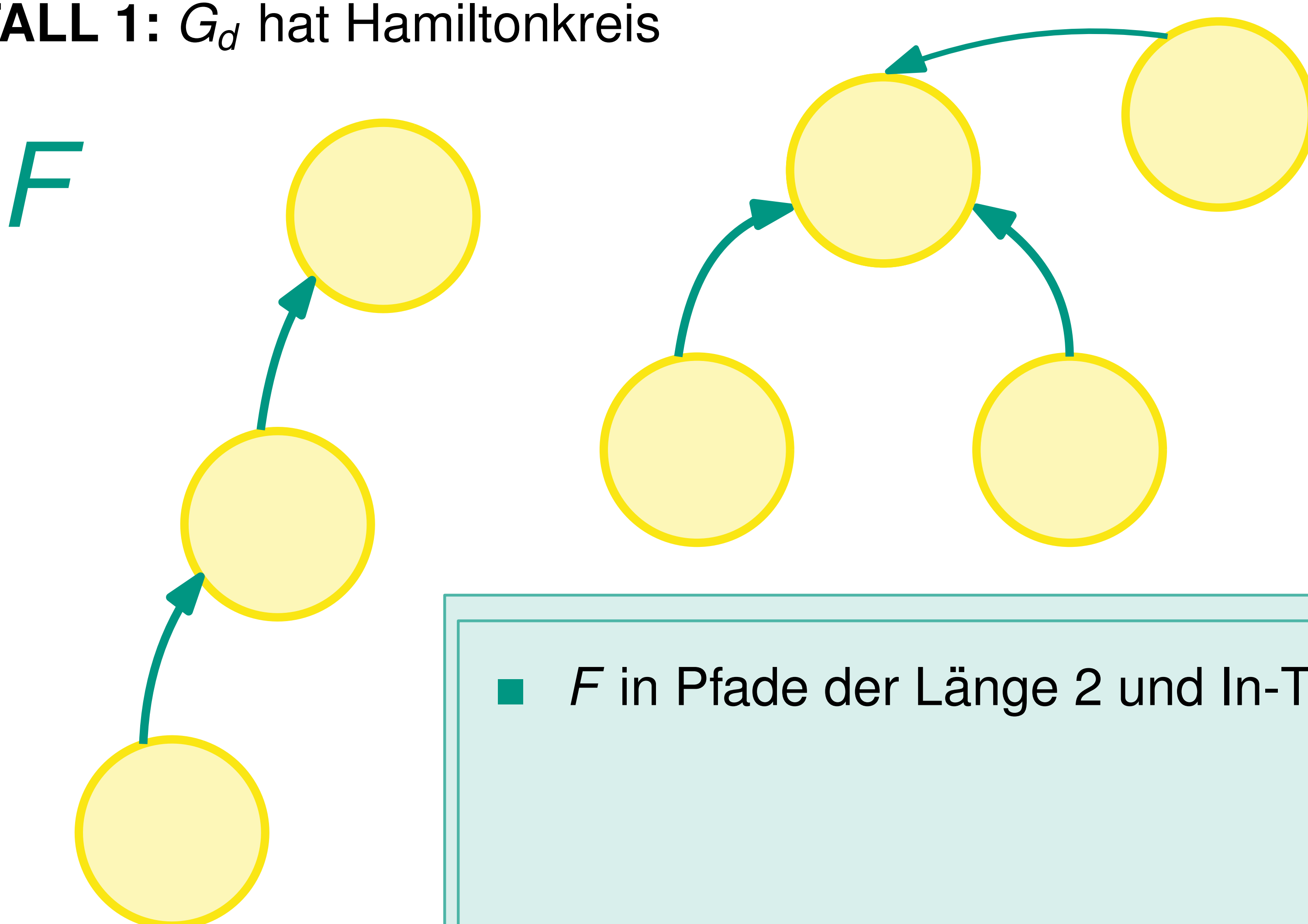
APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis



APPROXIMATIONSALGORITHMUS

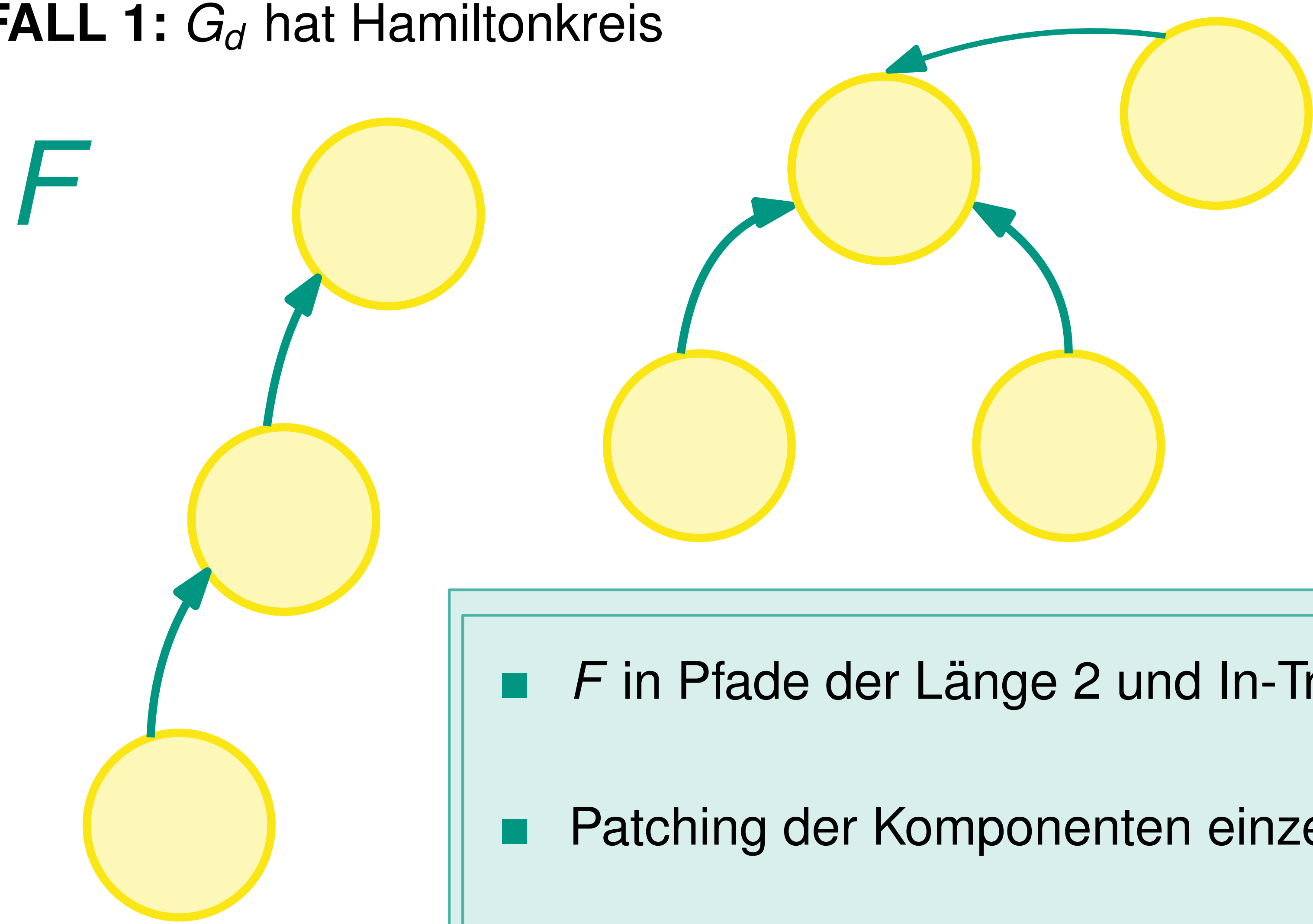
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen

APPROXIMATIONSALGORITHMUS

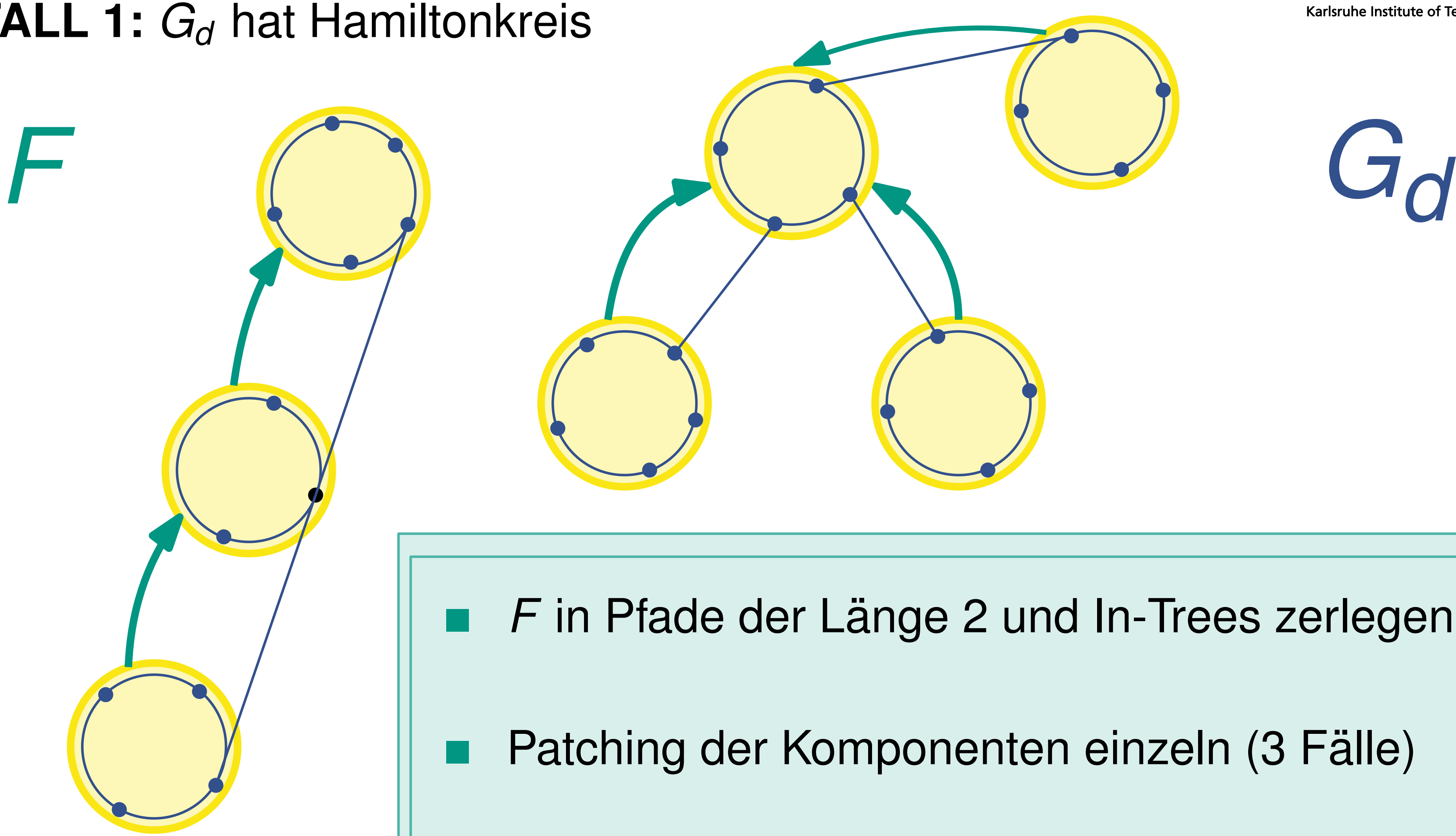
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

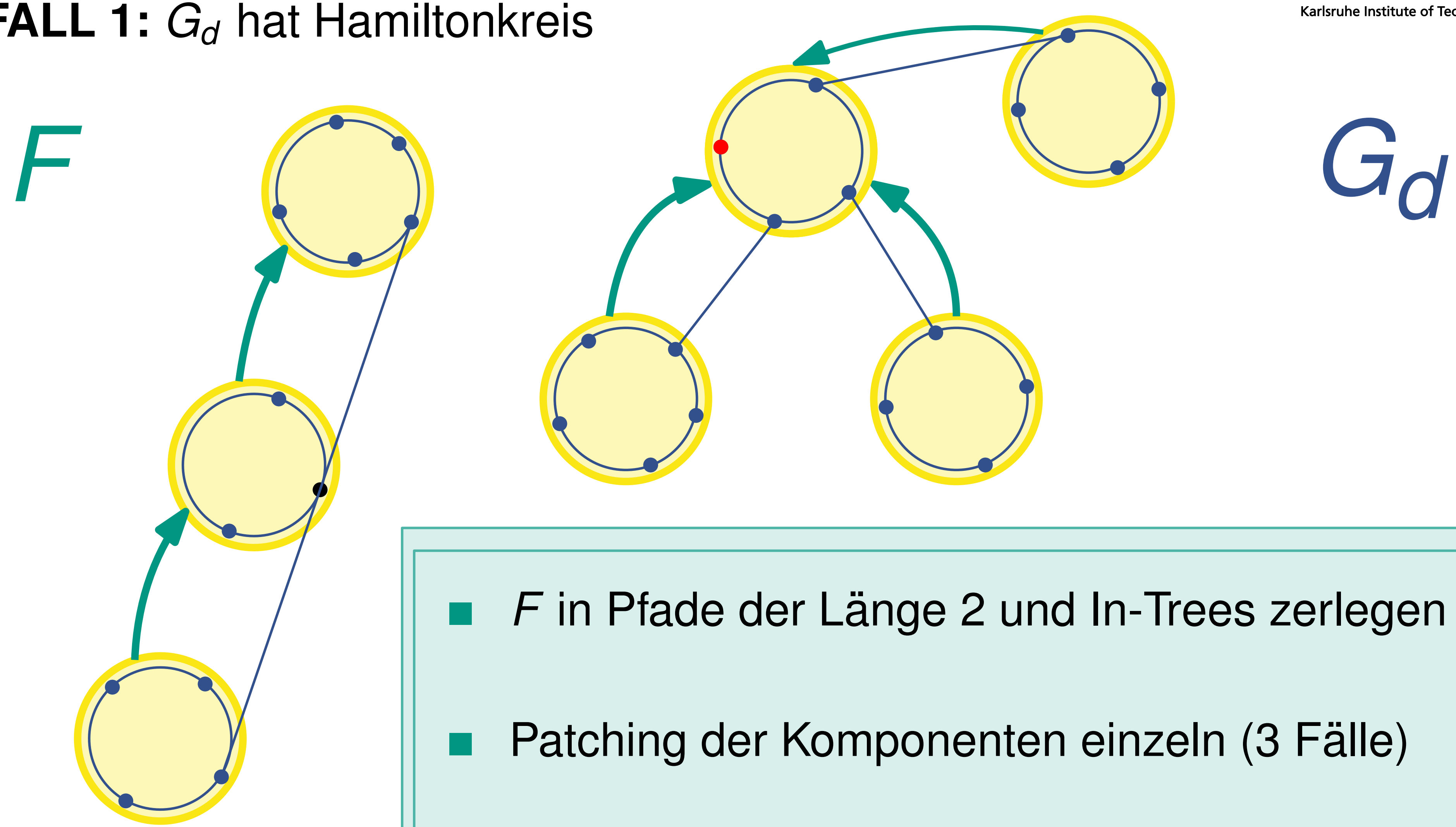


- *F* in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

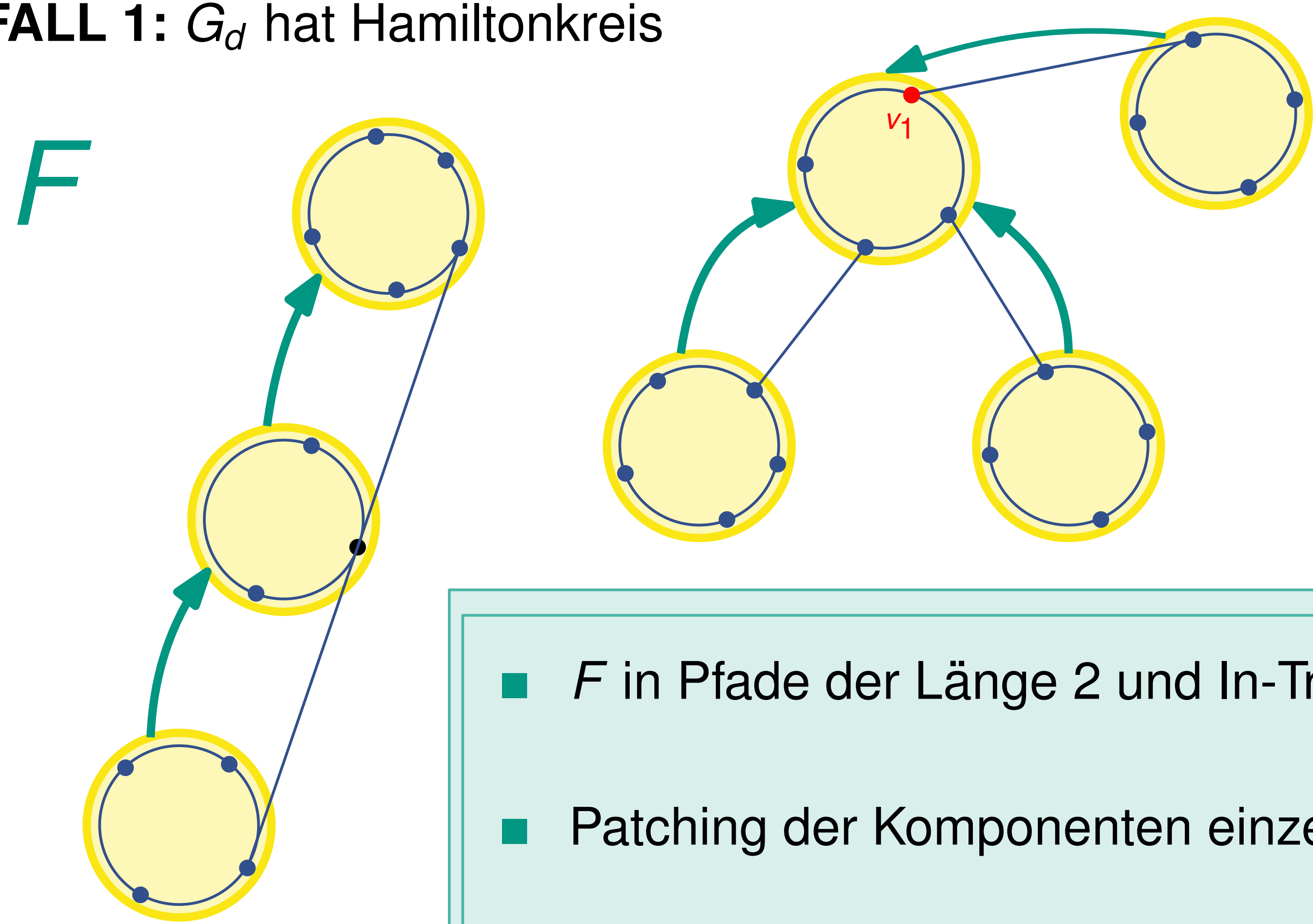


- *F* in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

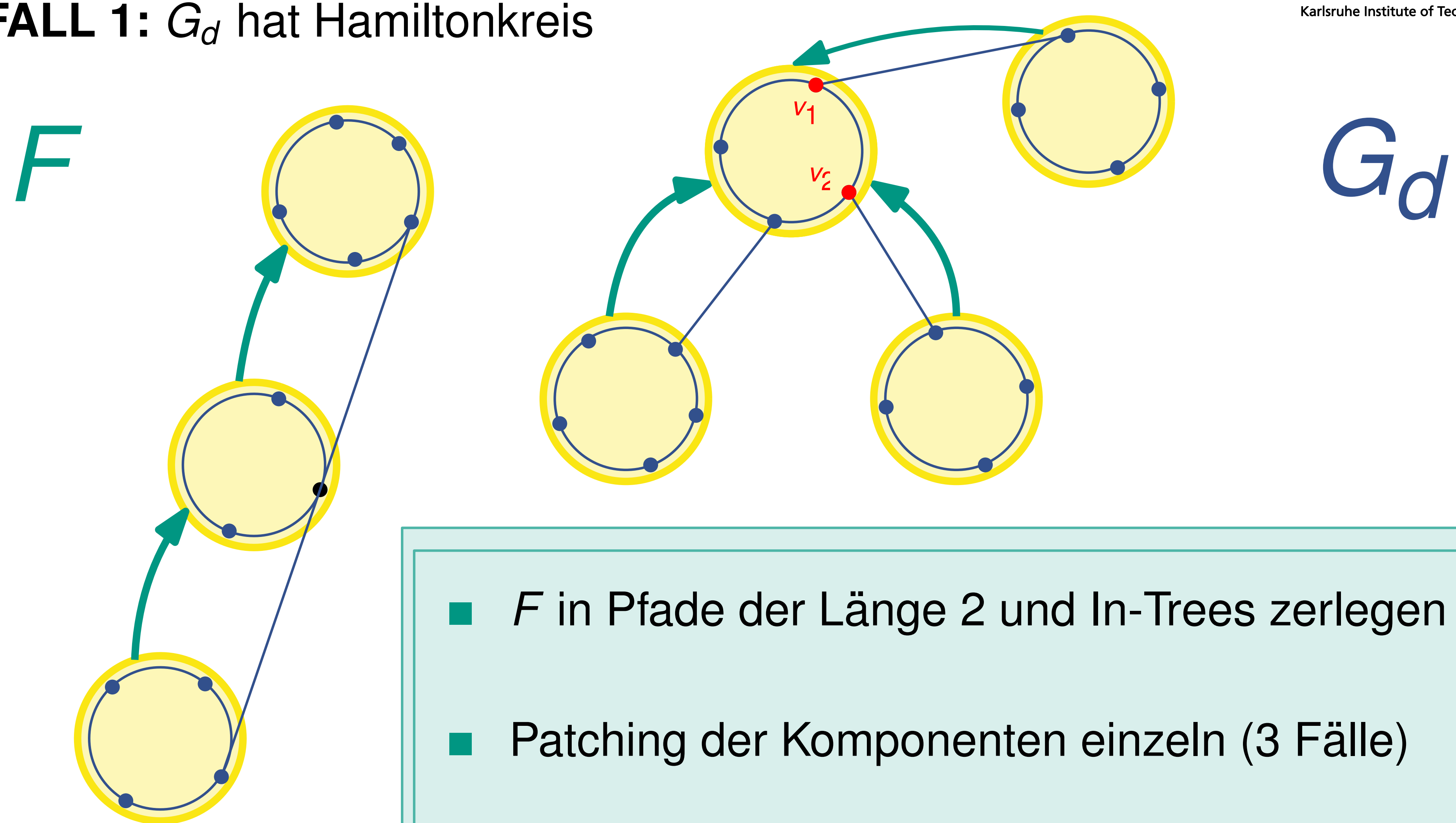
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

APPROXIMATIONSALGORITHMUS

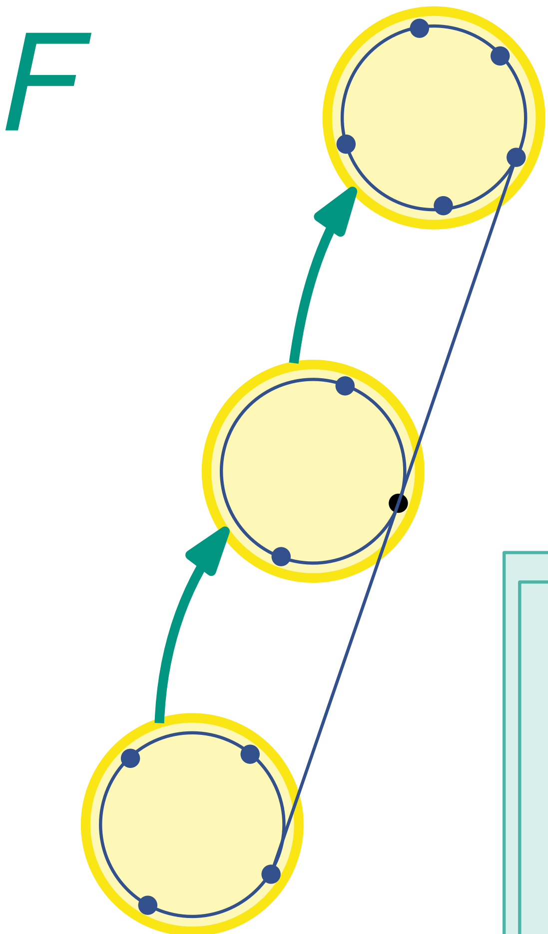
FALL 1: G_d hat Hamiltonkreis

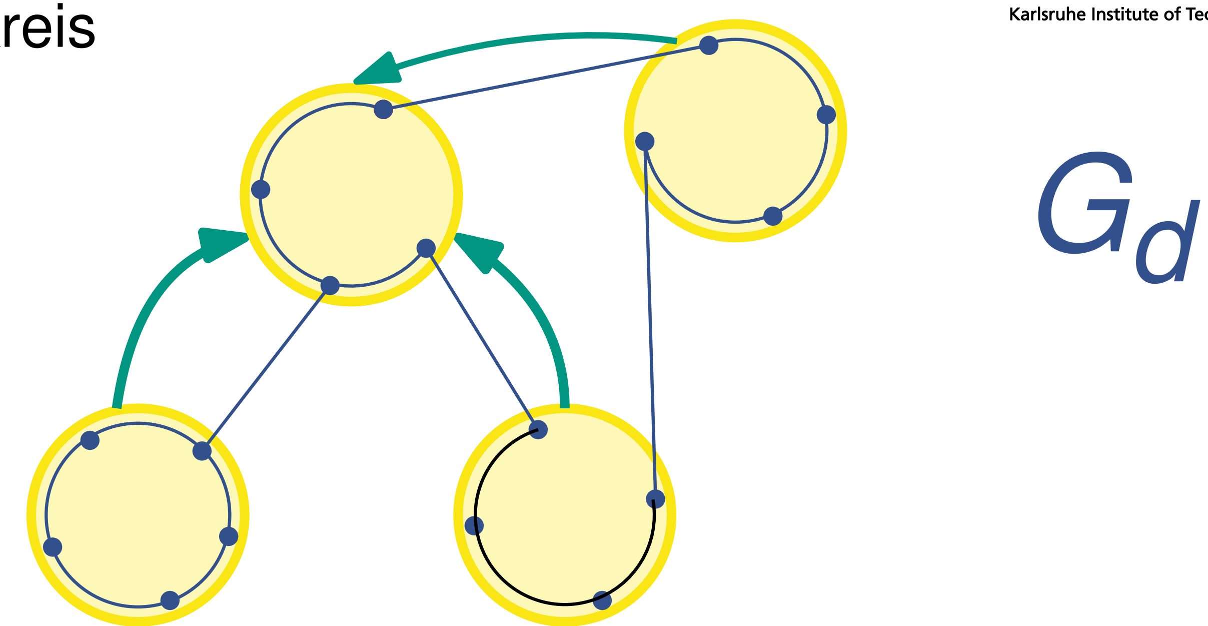


- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis





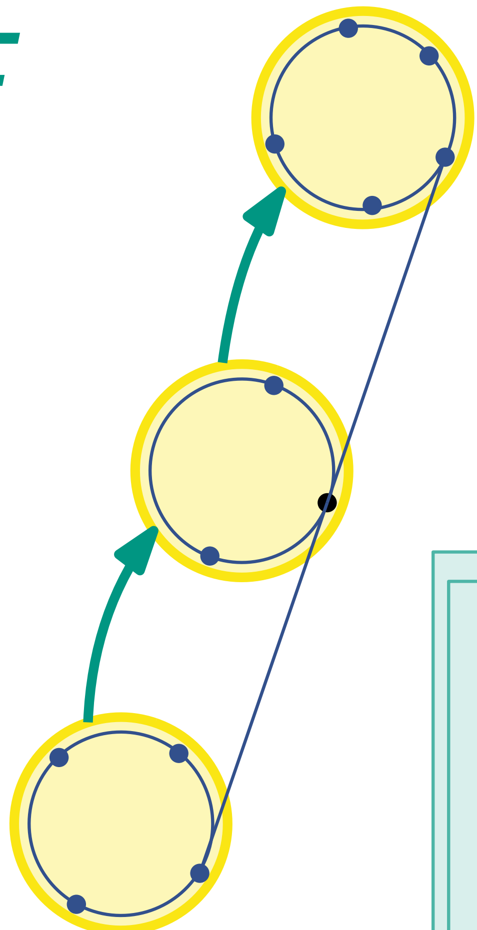
- *F* in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

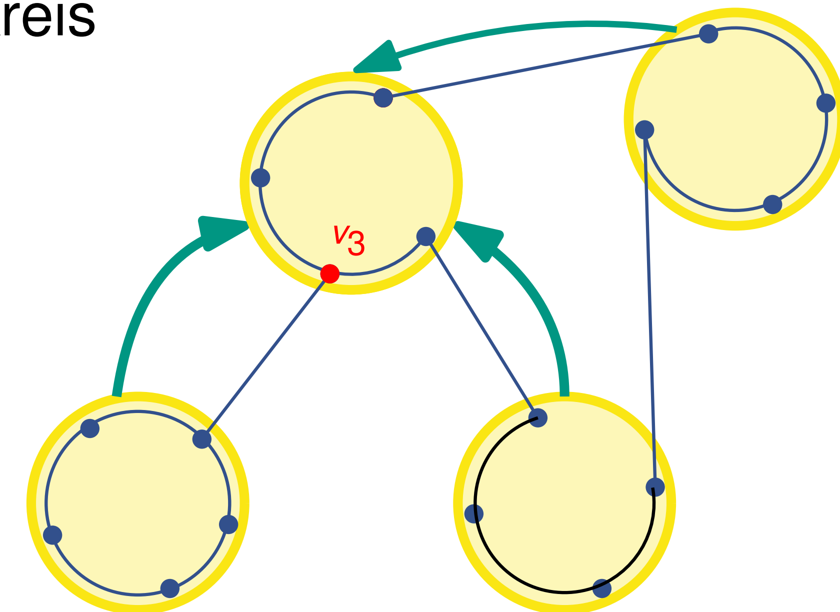
Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

F





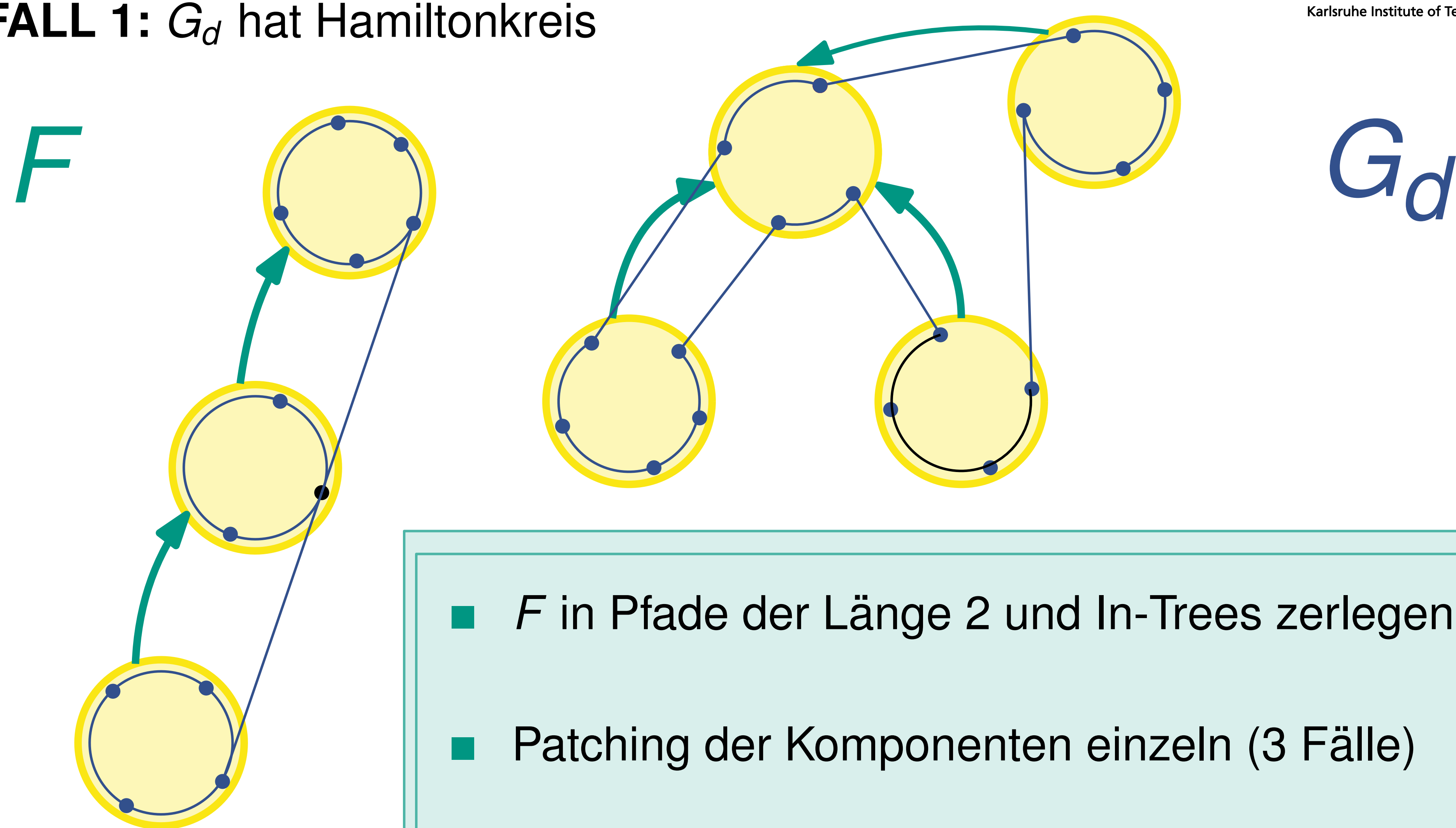
G_d

- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

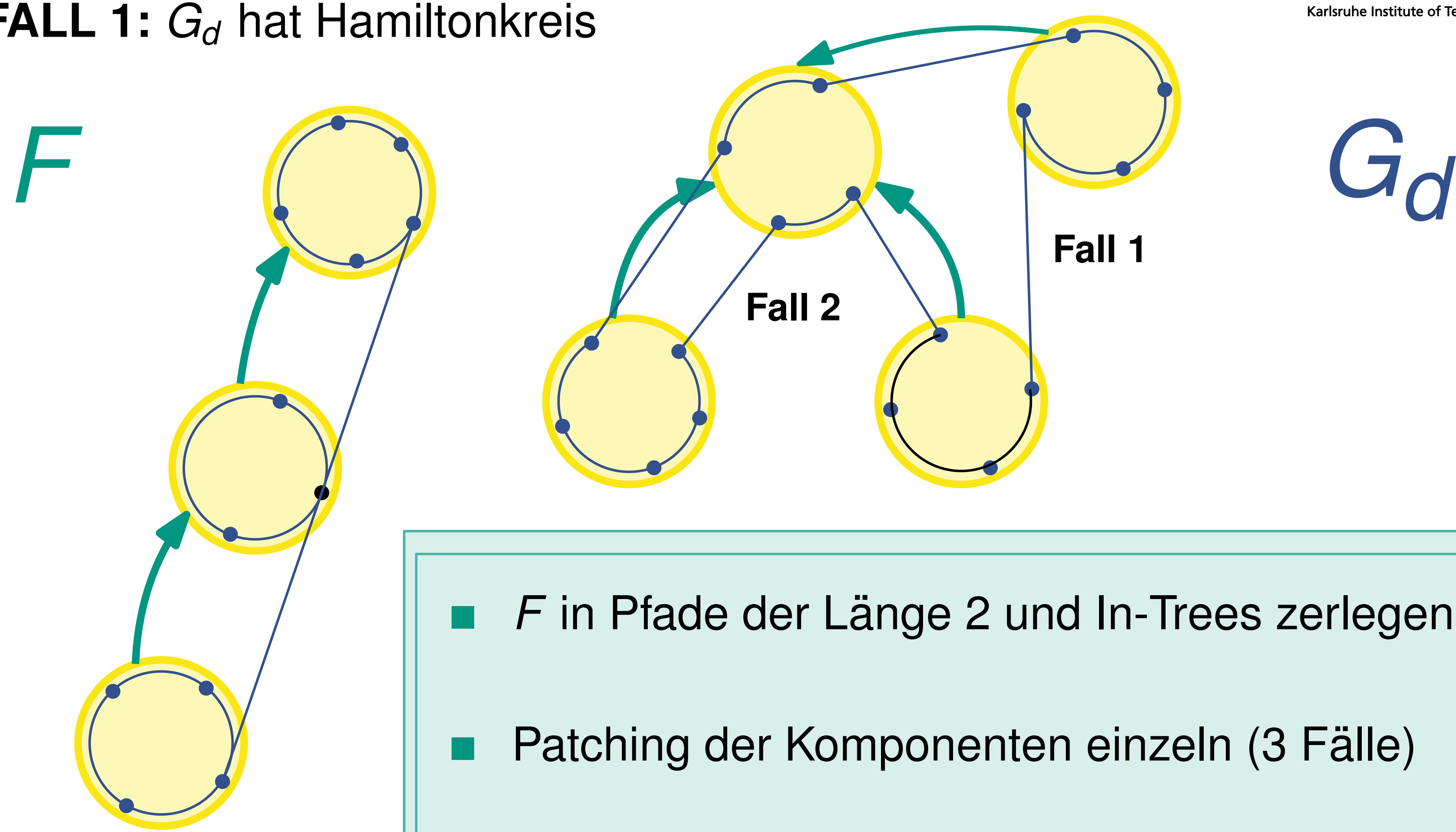
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

APPROXIMATIONSALGORITHMUS

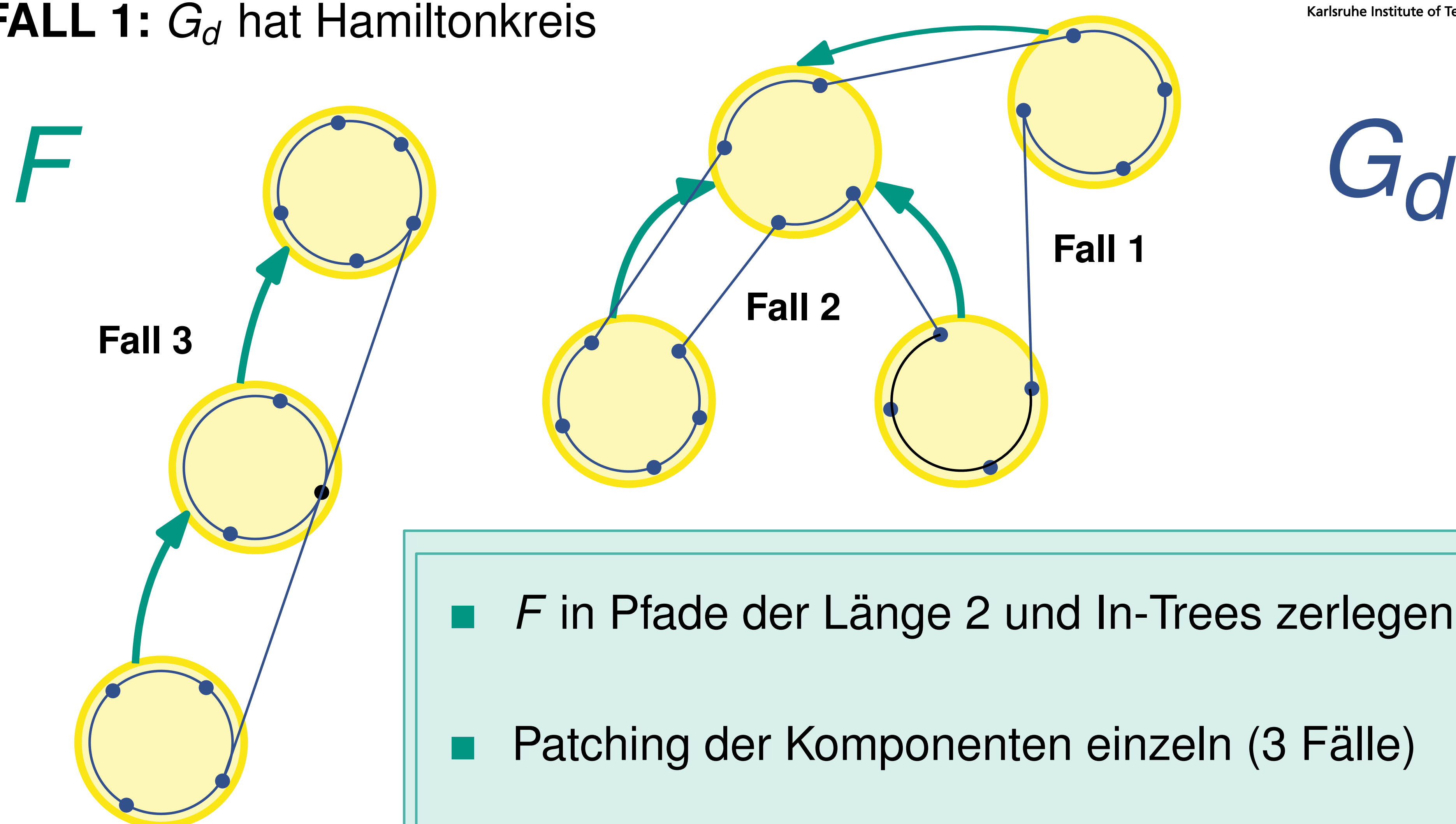
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

APPROXIMATIONSALGORITHMUS

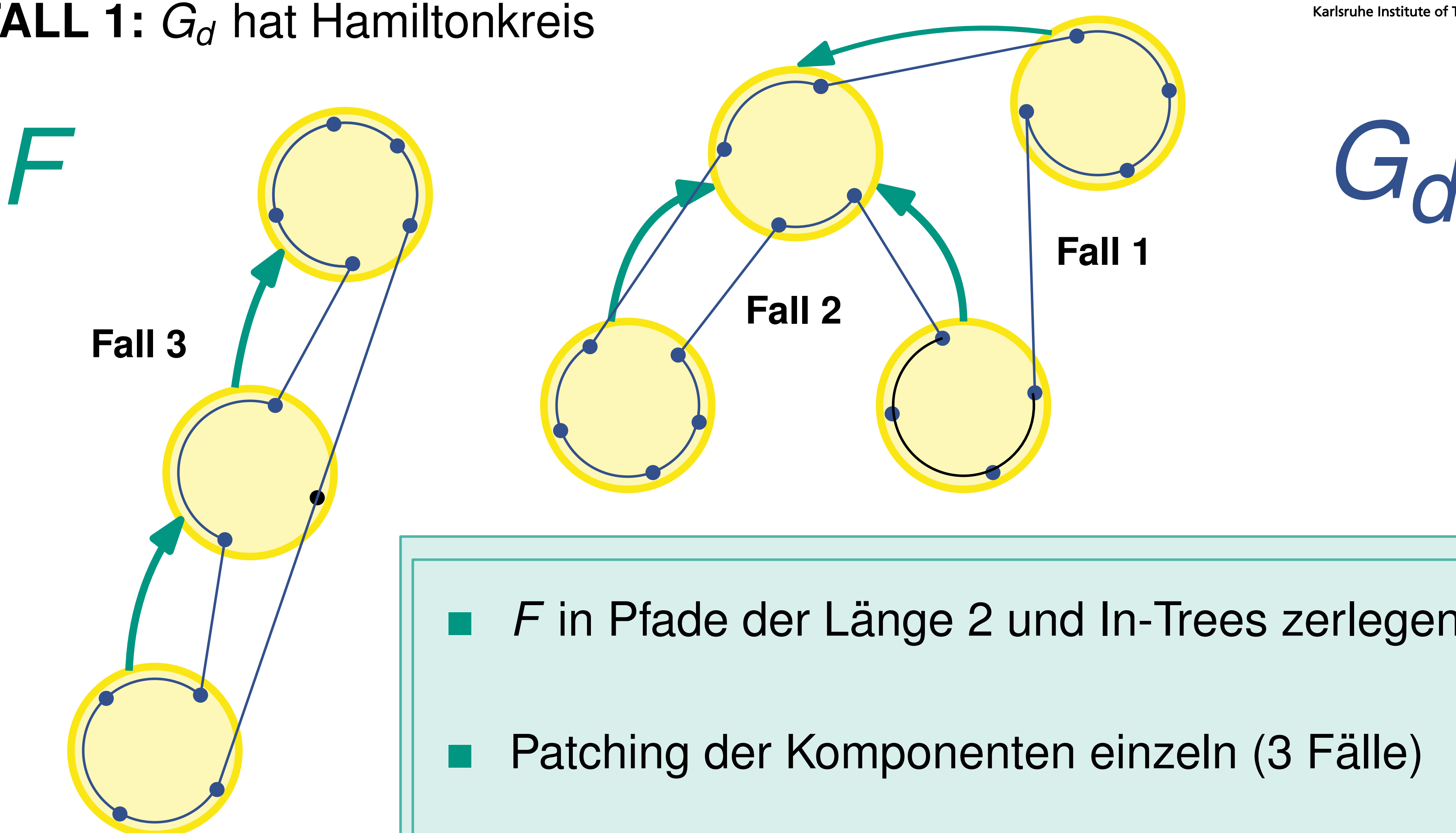
FALL 1: G_d hat Hamiltonkreis



- *F* in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

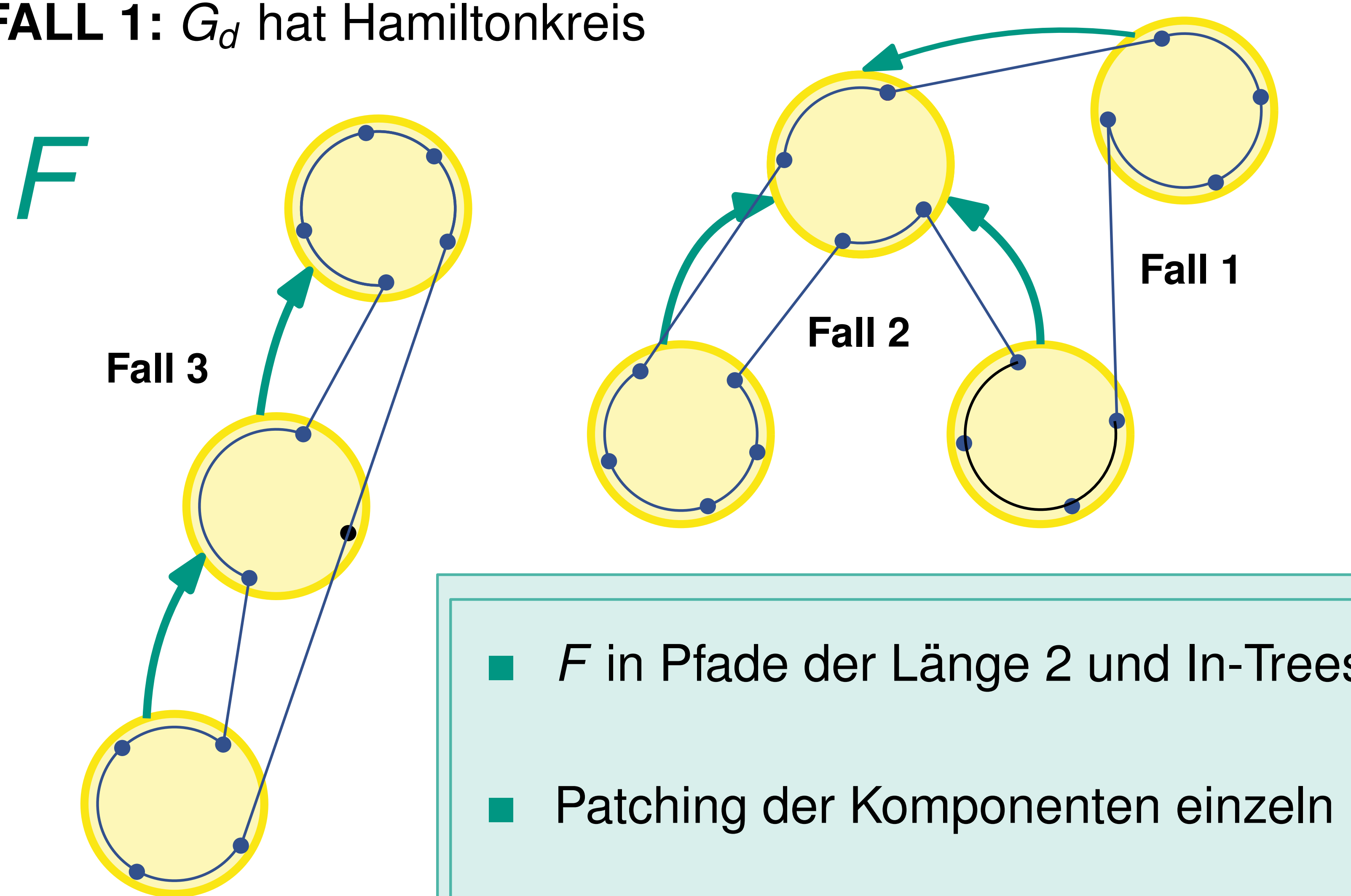


- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

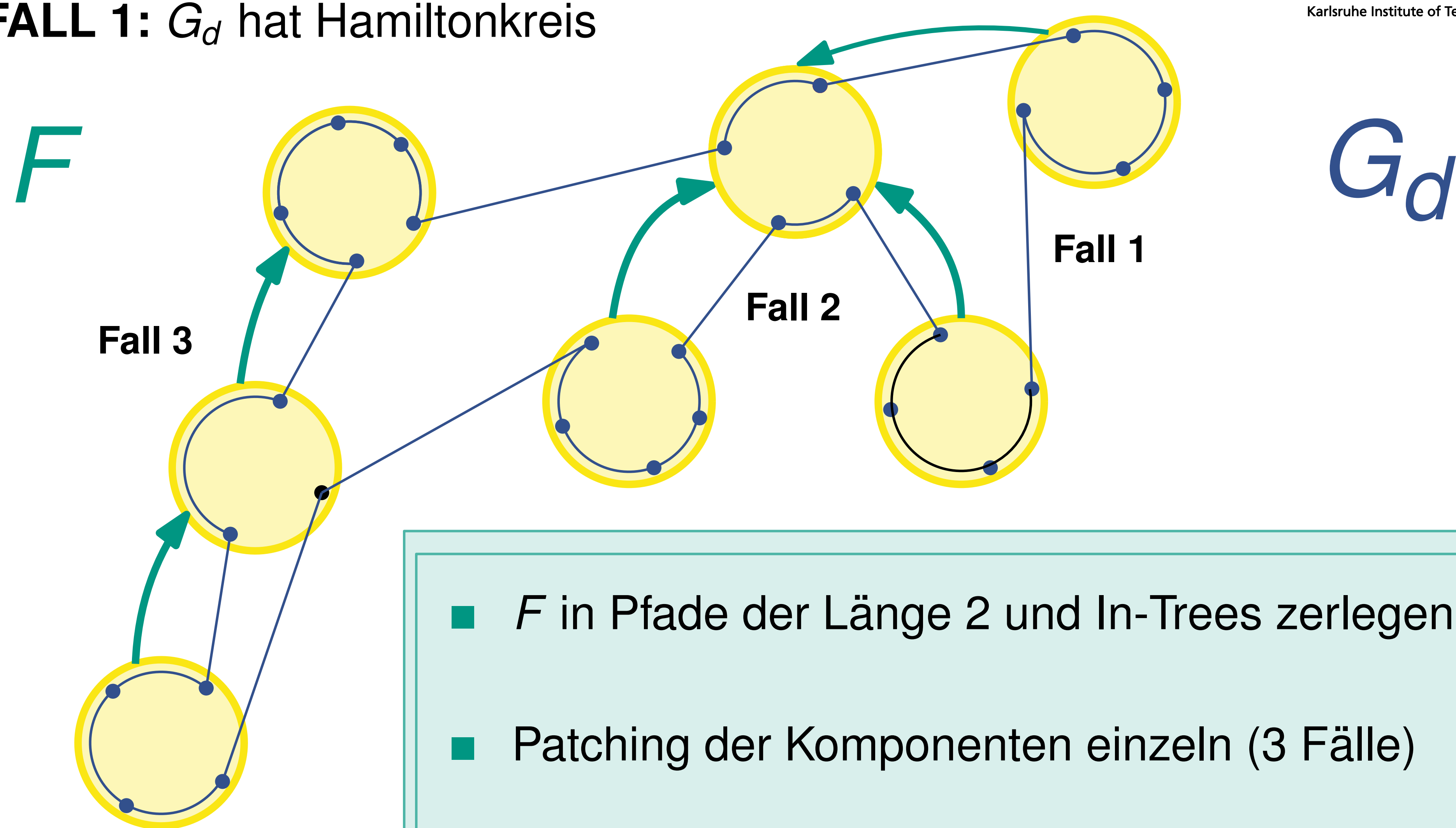
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)
- Patching der Komponenten miteinander

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

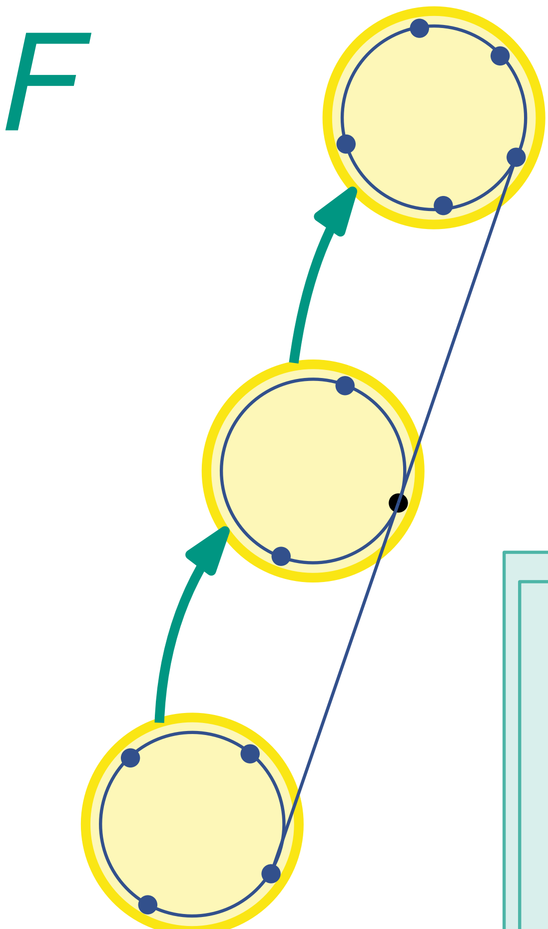


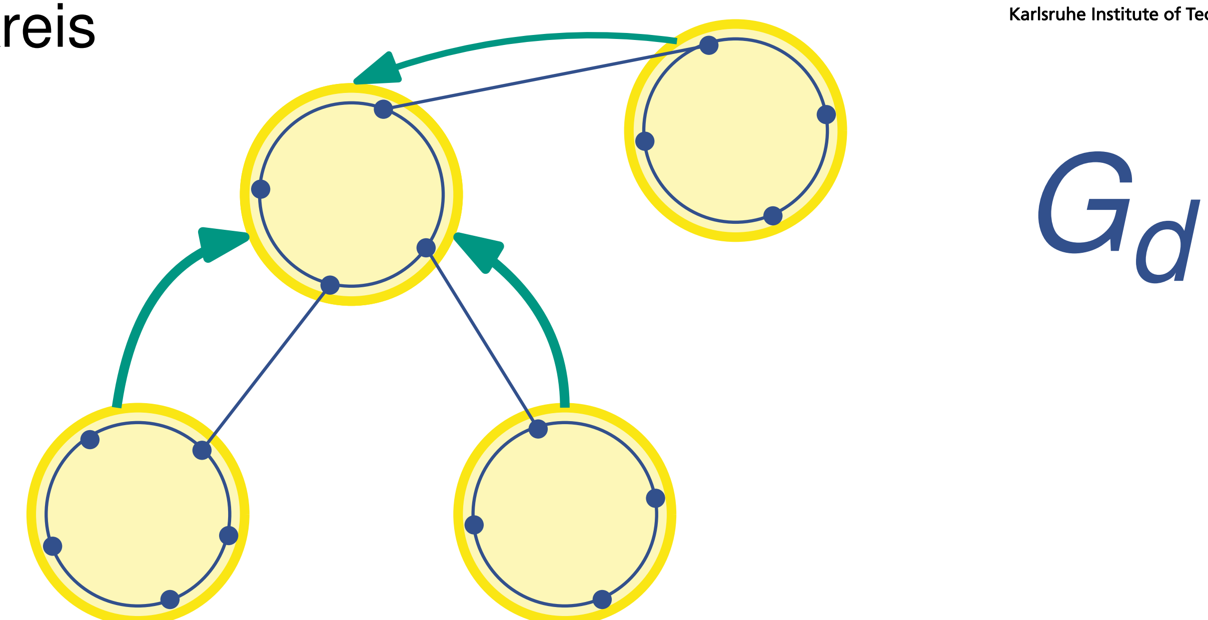
- F in Pfade der Länge 2 und In-Trees zerlegen
- Patching der Komponenten einzeln (3 Fälle)
- Patching der Komponenten miteinander

APPROXIMATIONSALGORITHMUS

KOSTEN

FALL 1: G_d hat Hamiltonkreis





- F in Pfade der Länge 2 und In-Trees zerlegen

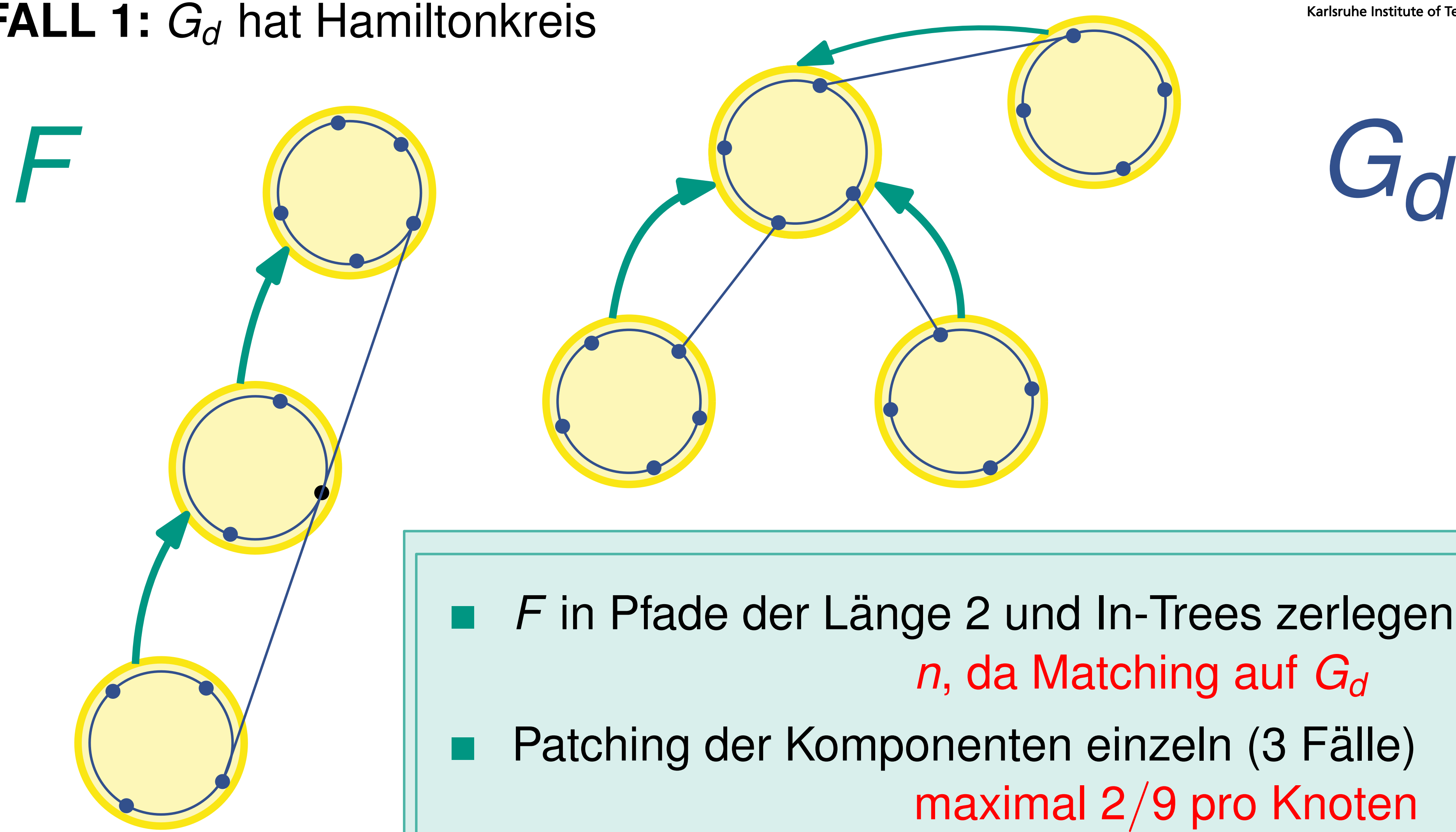
  n, da Matching auf G_d

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

KOSTEN

FALL 1: G_d hat Hamiltonkreis



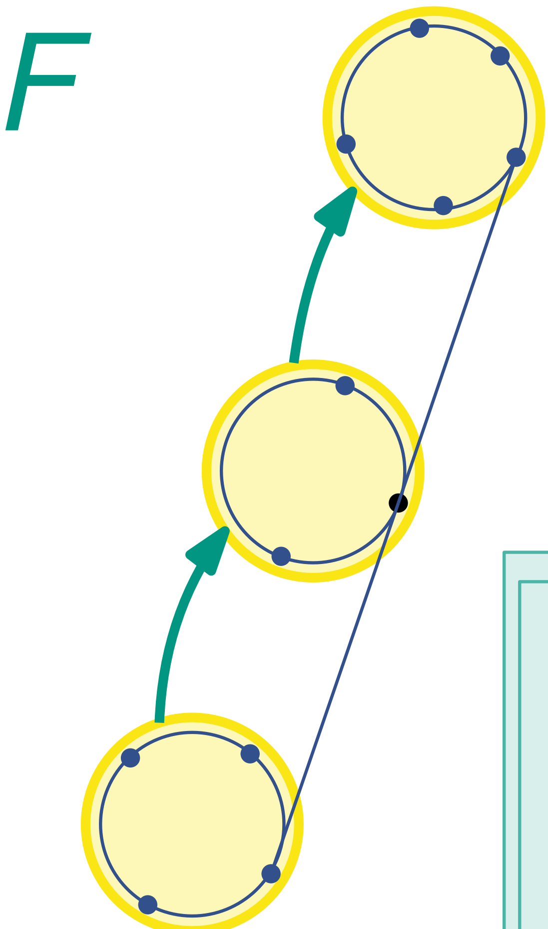
- F in Pfade der Länge 2 und In-Trees zerlegen
  n, da Matching auf G_d
- Patching der Komponenten einzeln (3 Fälle)
  maximal $2/9$ pro Knoten

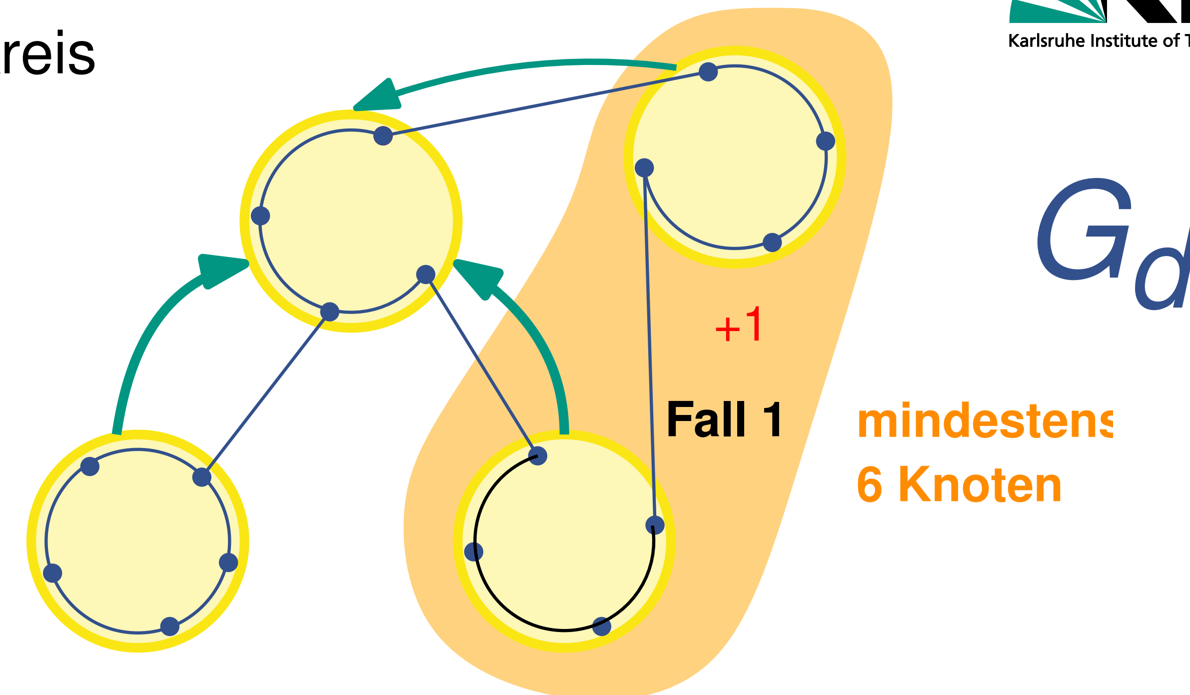
Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

KOSTEN





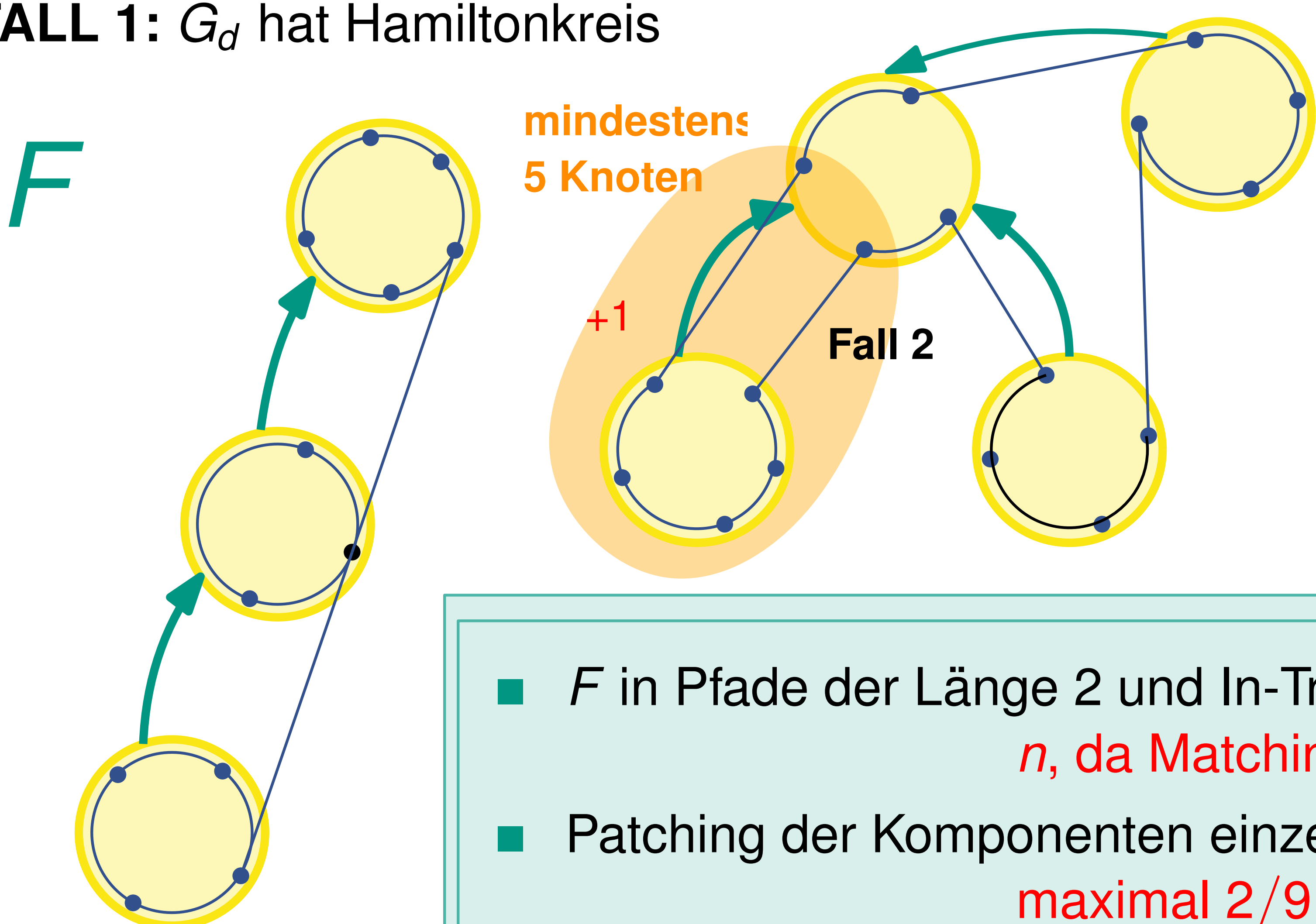
- F in Pfade der Länge 2 und In-Trees zerlegen
  n, da Matching auf G_d
- Patching der Komponenten einzeln (3 Fälle)
  maximal $2/9$ pro Knoten

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 1: G_d hat Hamiltonkreis

KOSTEN



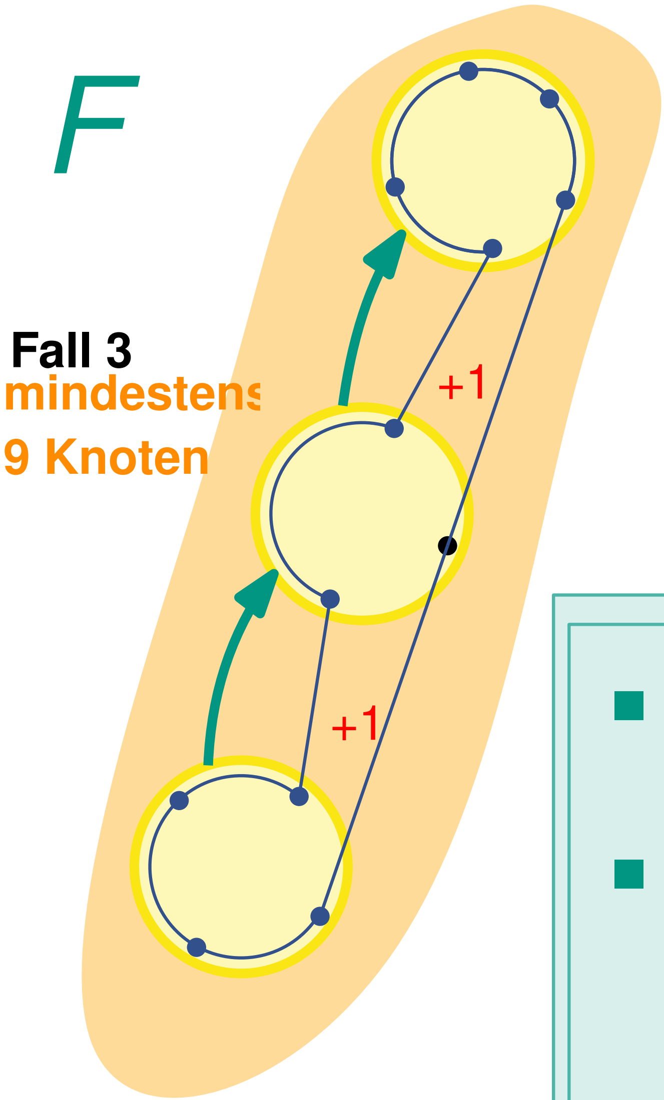
- F in Pfade der Länge 2 und In-Trees zerlegen
  n, da Matching auf G_d
- Patching der Komponenten einzeln (3 Fälle)
  maximal $2/9$ pro Knoten

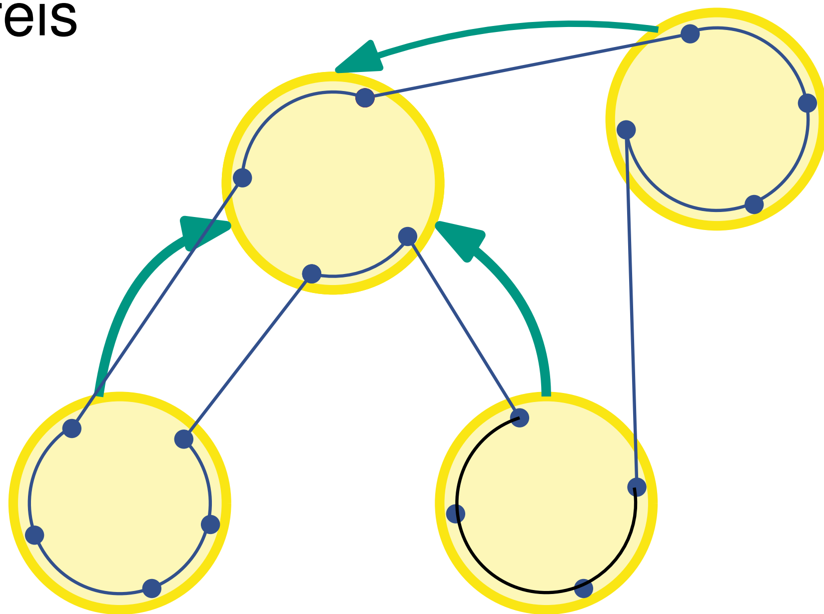
APPROXIMATIONSALGORITHMUS

KOSTEN

FALL 1: G_d hat Hamiltonkreis

F





G_d

- F in Pfade der Länge 2 und In-Trees zerlegen

  n, da Matching auf G_d

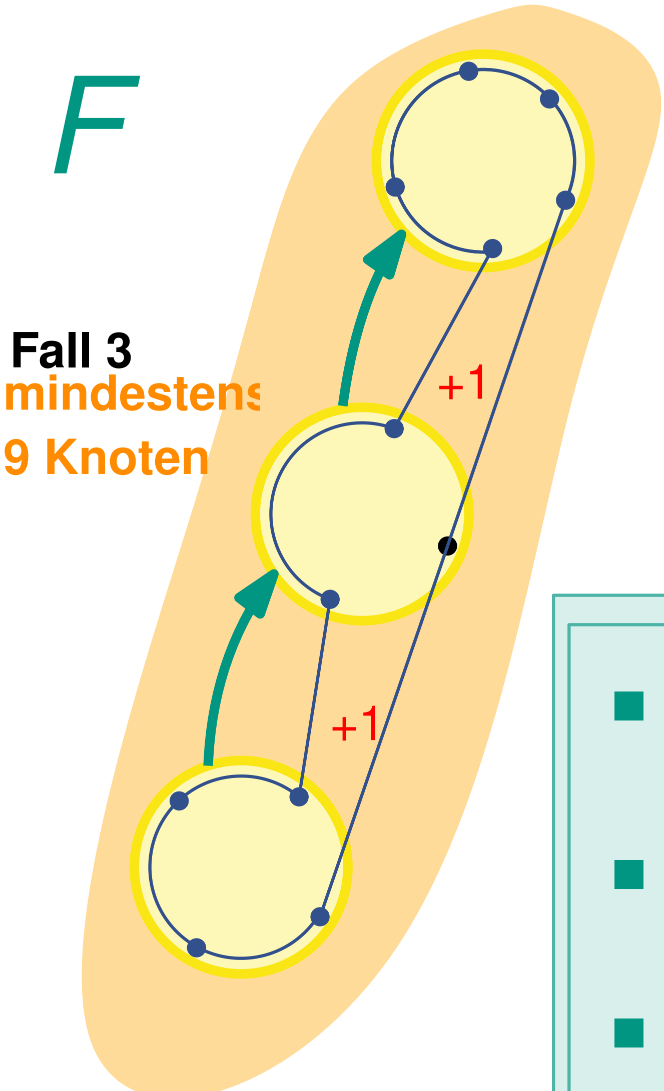
- Patching der Komponenten einzeln (3 Fälle)

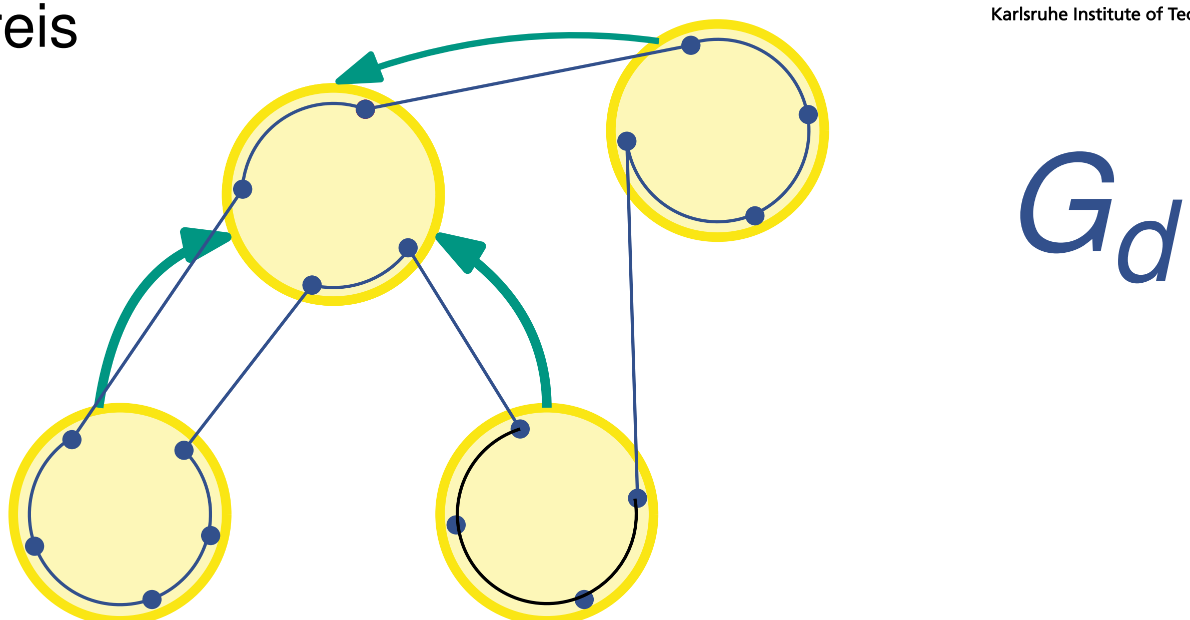
  maximal $2/9$ pro Knoten

APPROXIMATIONSALGORITHMUS

KOSTEN

FALL 1: G_d hat Hamiltonkreis



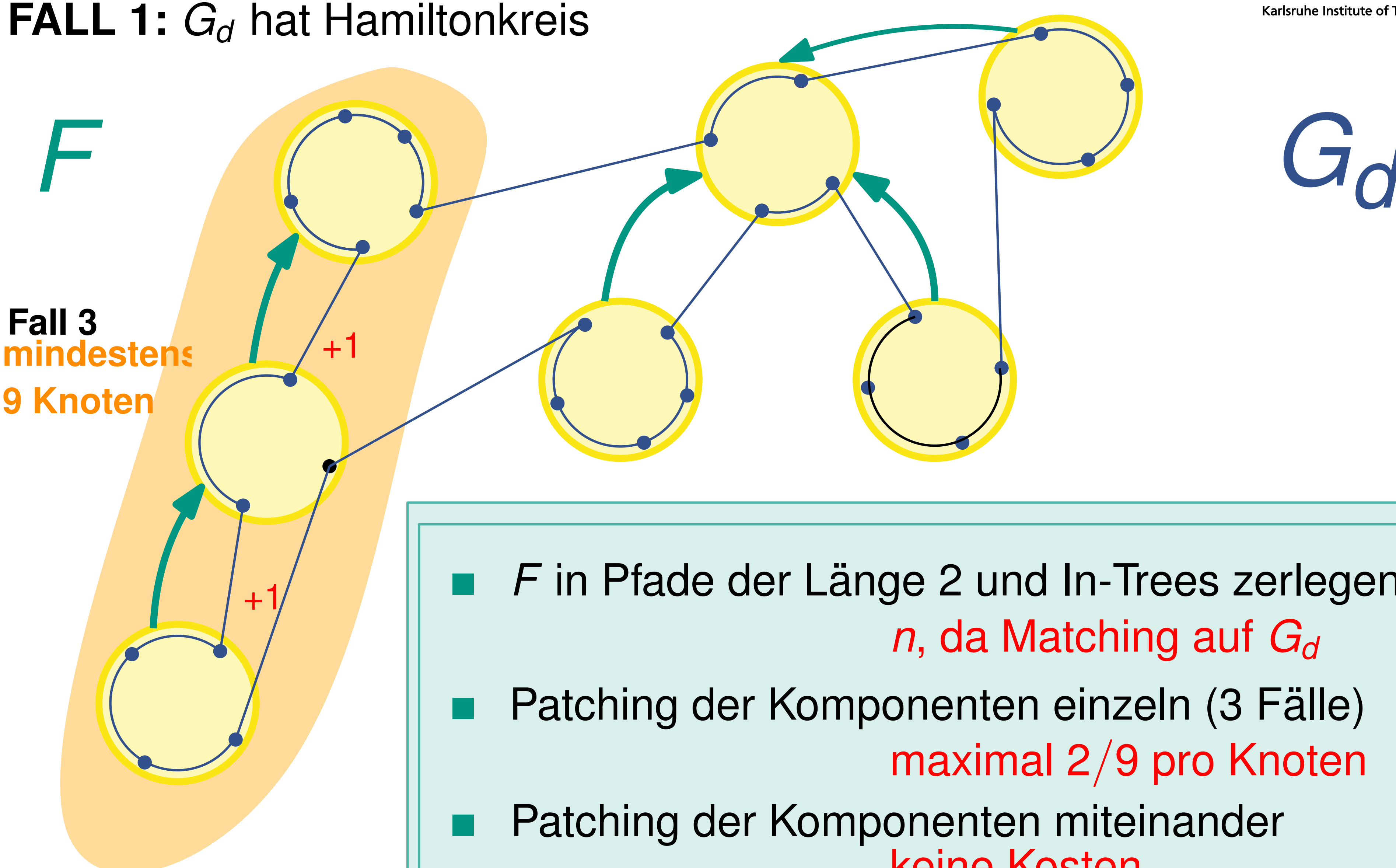


- F in Pfade der Länge 2 und In-Trees zerlegen
  n, da Matching auf G_d
- Patching der Komponenten einzeln (3 Fälle)
  maximal $2/9$ pro Knoten
- Patching der Komponenten miteinander
  keine Kosten

APPROXIMATIONSALGORITHMUS

KOSTEN

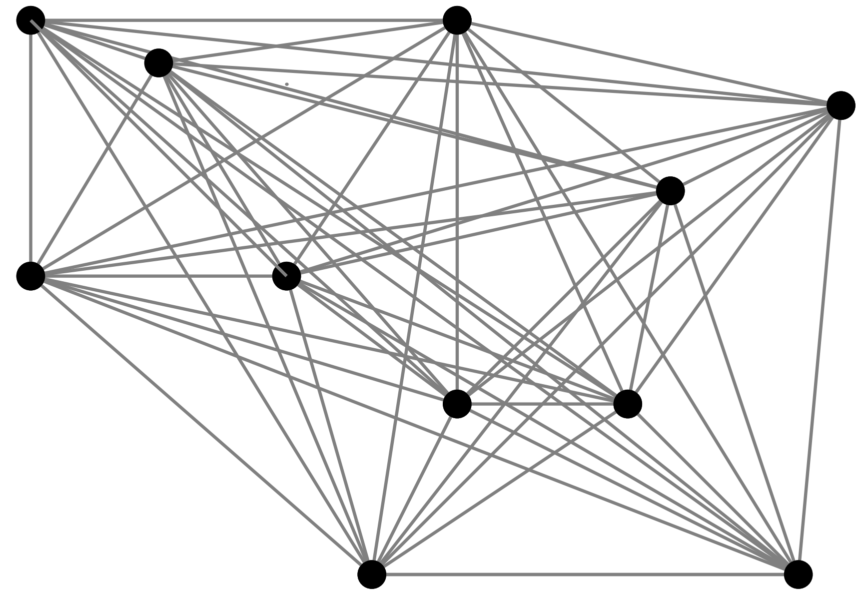
FALL 1: G_d hat Hamiltonkreis



- F in Pfade der Länge 2 und In-Trees zerlegen
  n, da Matching auf G_d
- Patching der Komponenten einzeln (3 Fälle)
  maximal $2/9$ pro Knoten
- Patching der Komponenten miteinander
  keine Kosten

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

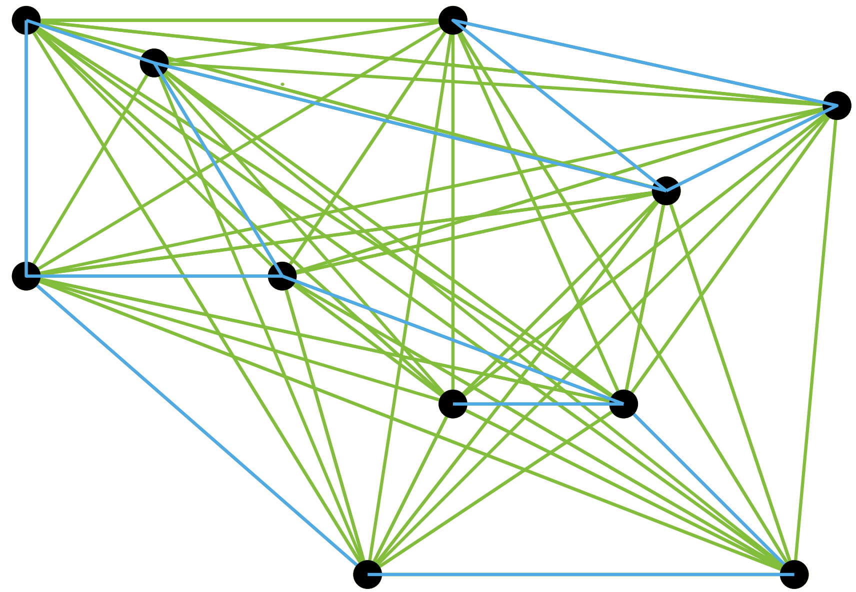


$$G = (V, E) \quad w : E \to \{1, 2\}$$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis



$$G = (V, E) \quad w : E \to \{1, 2\}$$

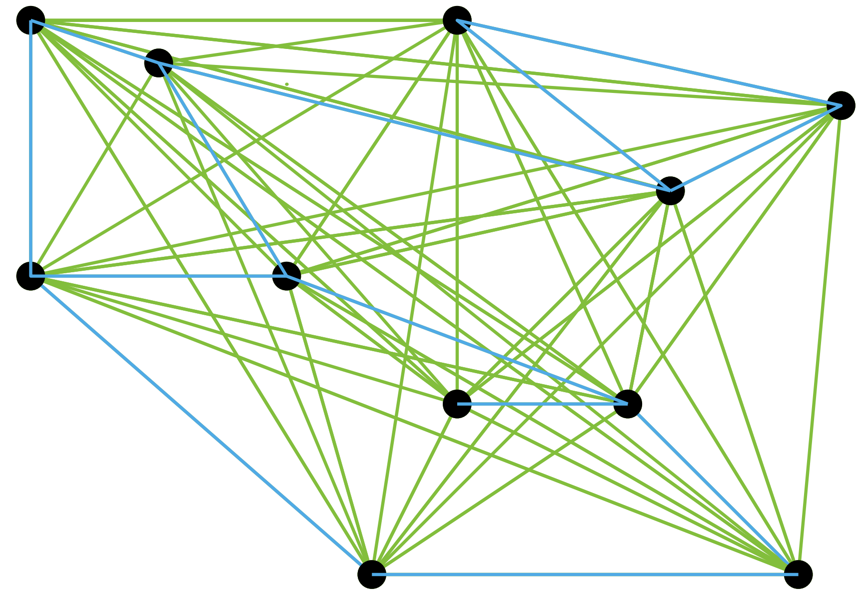
$$e \in E' \leftrightarrow e \in E, w(e) = 1$$

$$e \in E'' \leftrightarrow e \in E, w(e) = 2$$

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G sodass:
  - → maximal ein nicht-reiner Kreis
  - → darin: keine 2-Kante inzident zu 1-Kante, die Kreis verlässt



$$G = (V, E) \quad w : E \to \{1, 2\}$$

$$e \in E' \leftrightarrow e \in E, w(e) = 1$$

$$e \in E'' \leftrightarrow e \in E, w(e) = 2$$

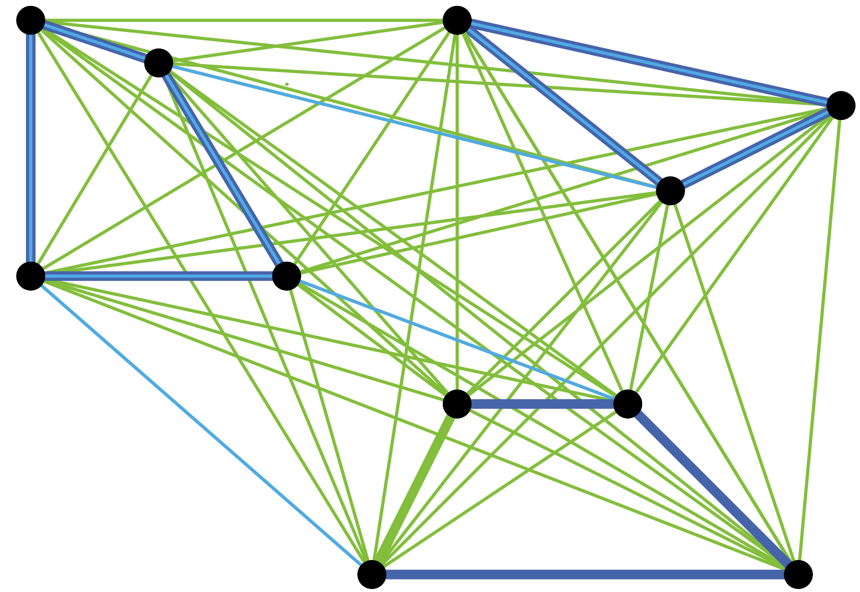
Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS
FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G sodass:

→ maximal ein nicht-reiner Kreis

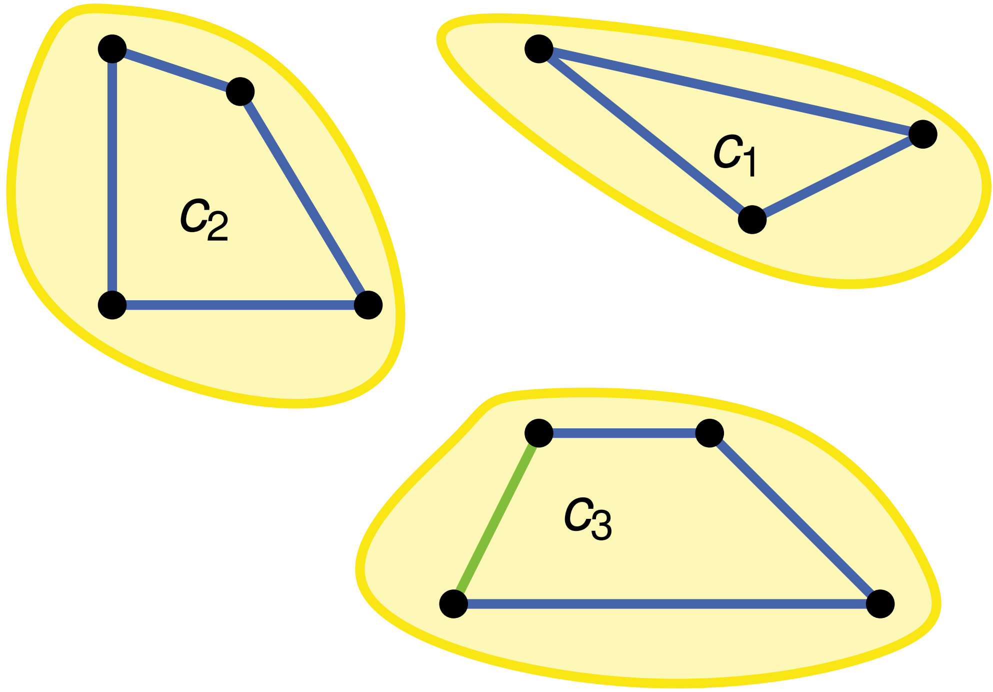
→ darin: keine 2-Kante inzident zu 1-Kante, die Kreis verlässt



APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

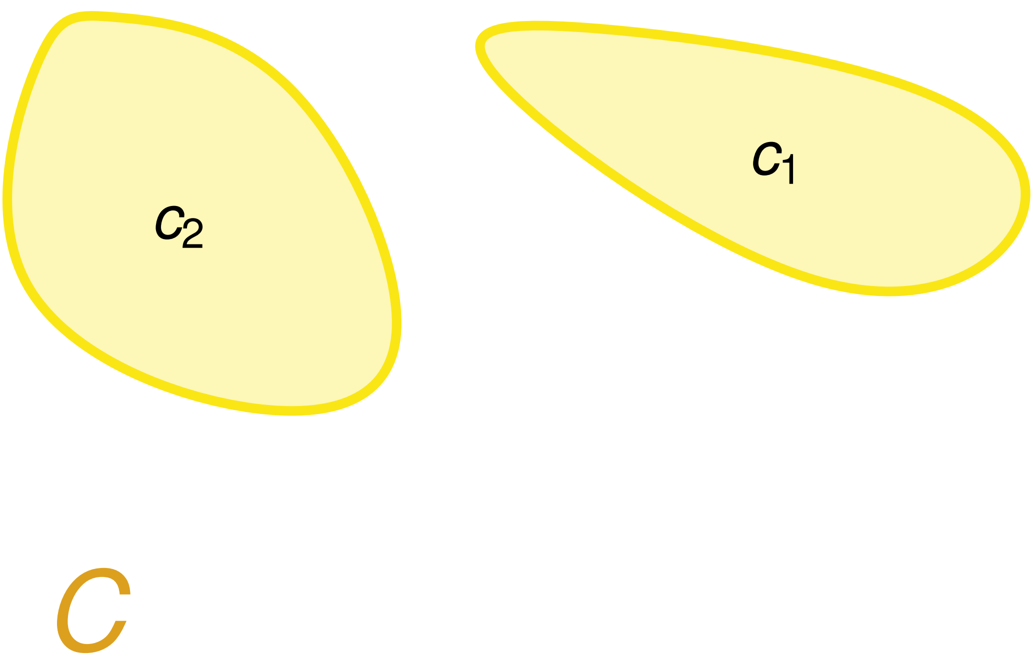
- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise

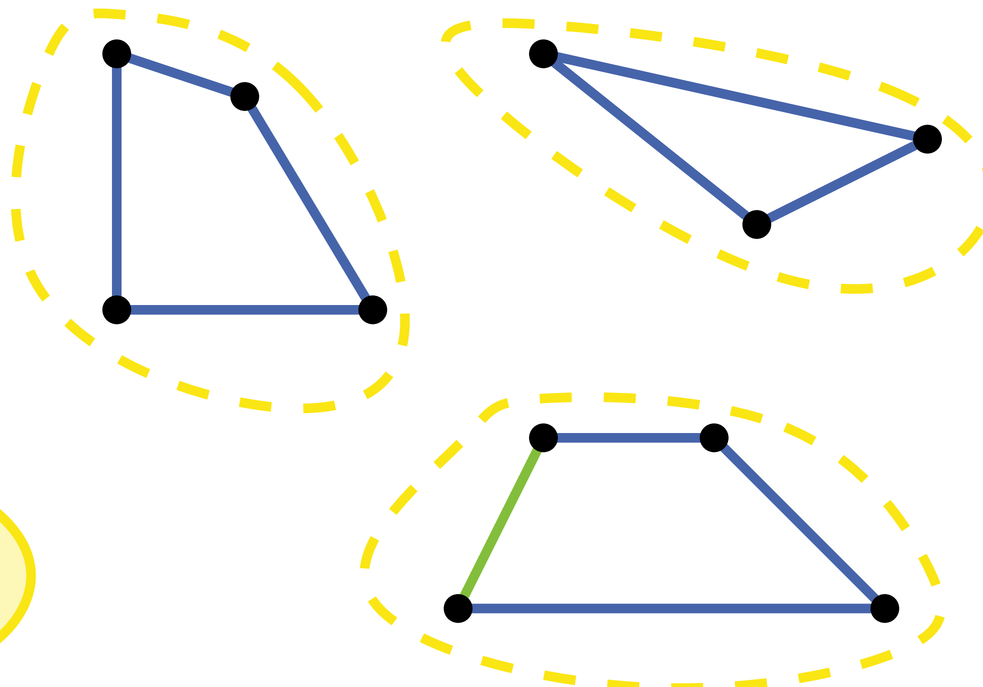


APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise

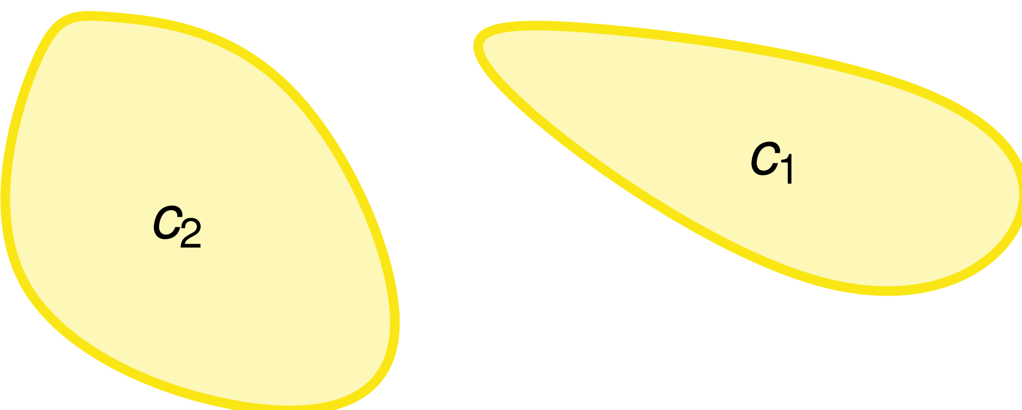


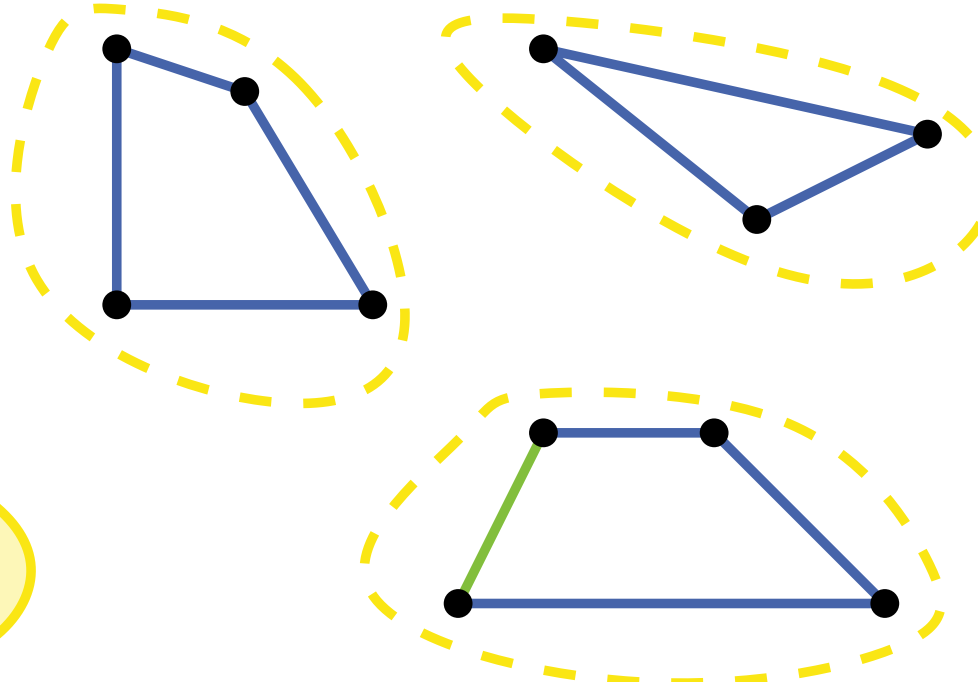


APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise





C

$$B = (V \cup C, D)$$

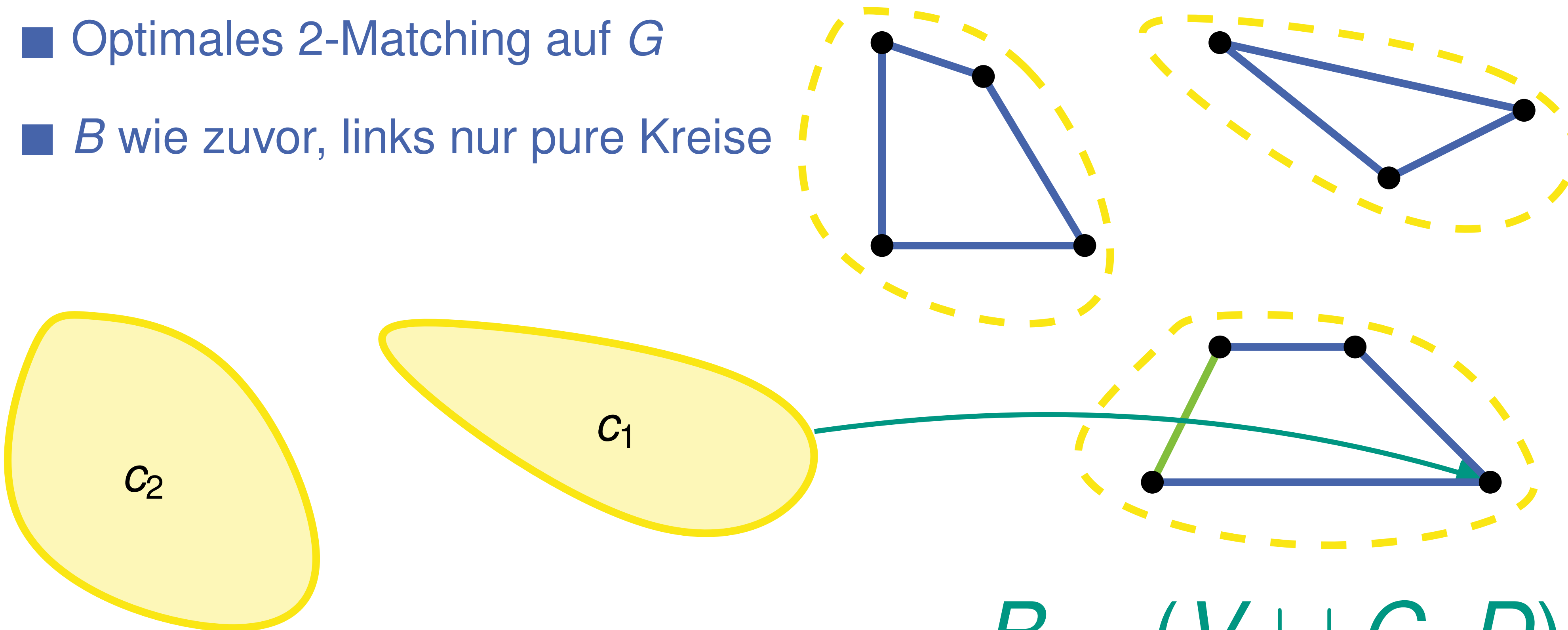
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise



C

$$B = (V \cup C, D)$$

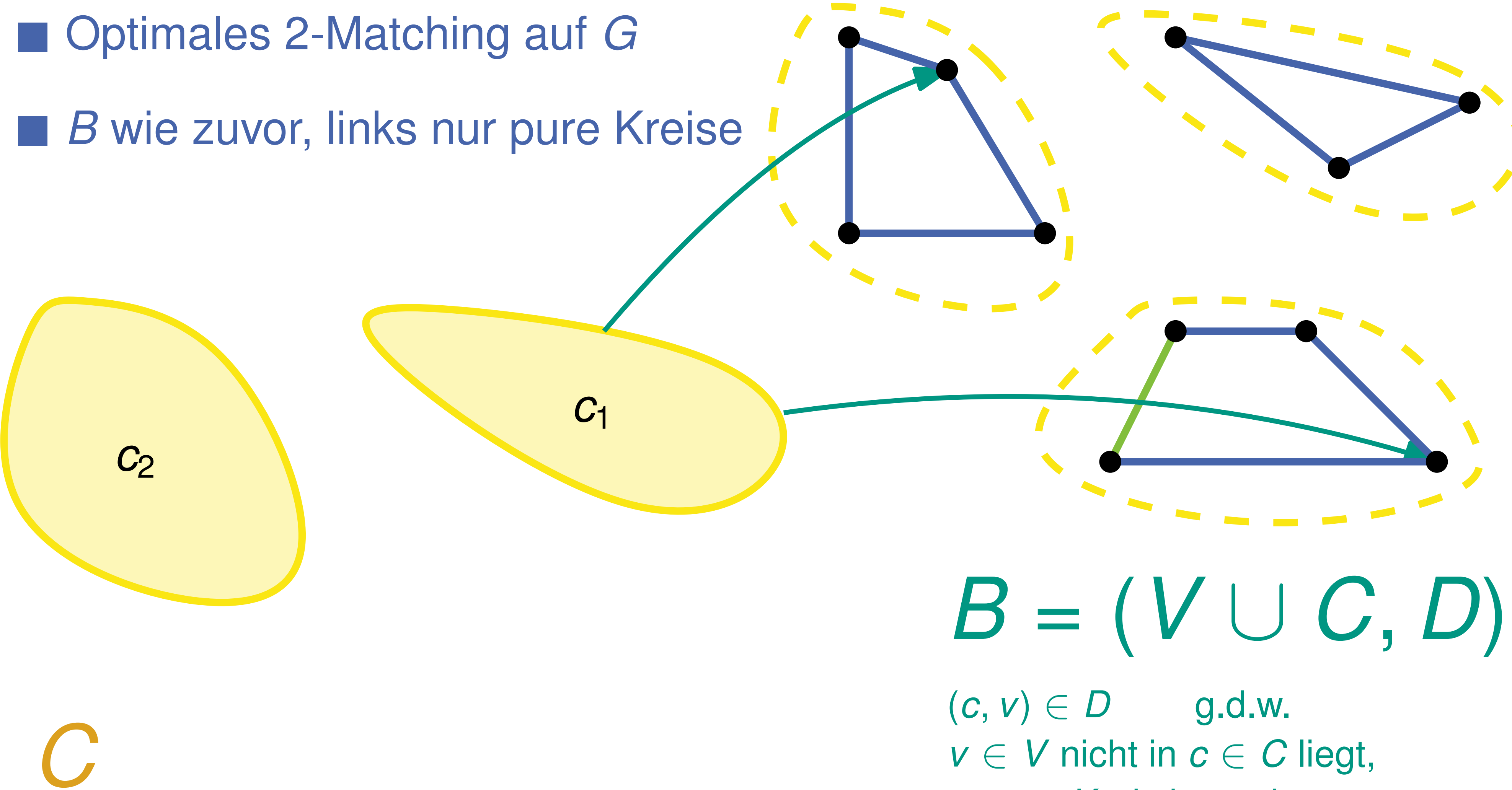
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise



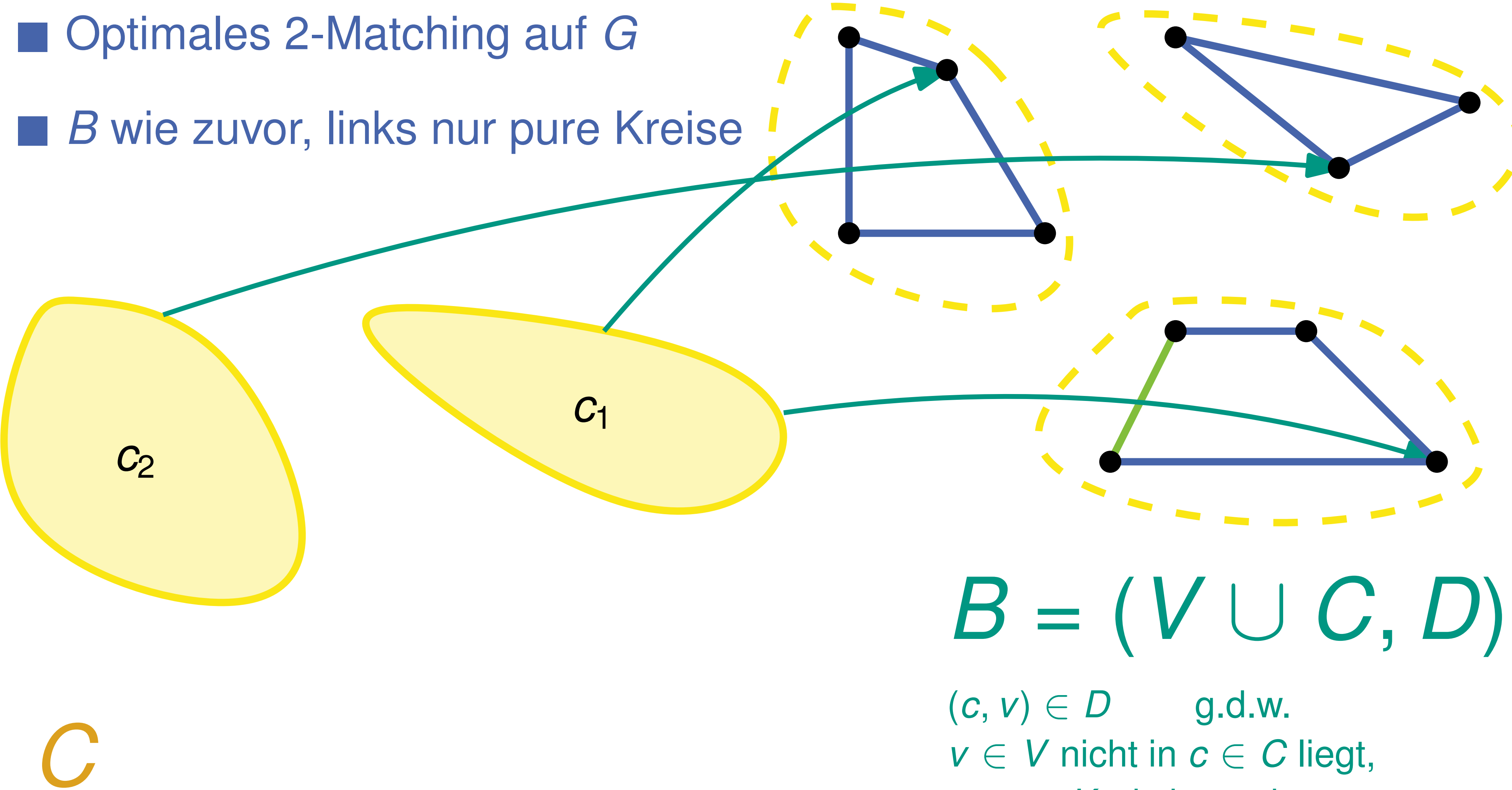
$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise



$$B = (V \cup C, D)$$

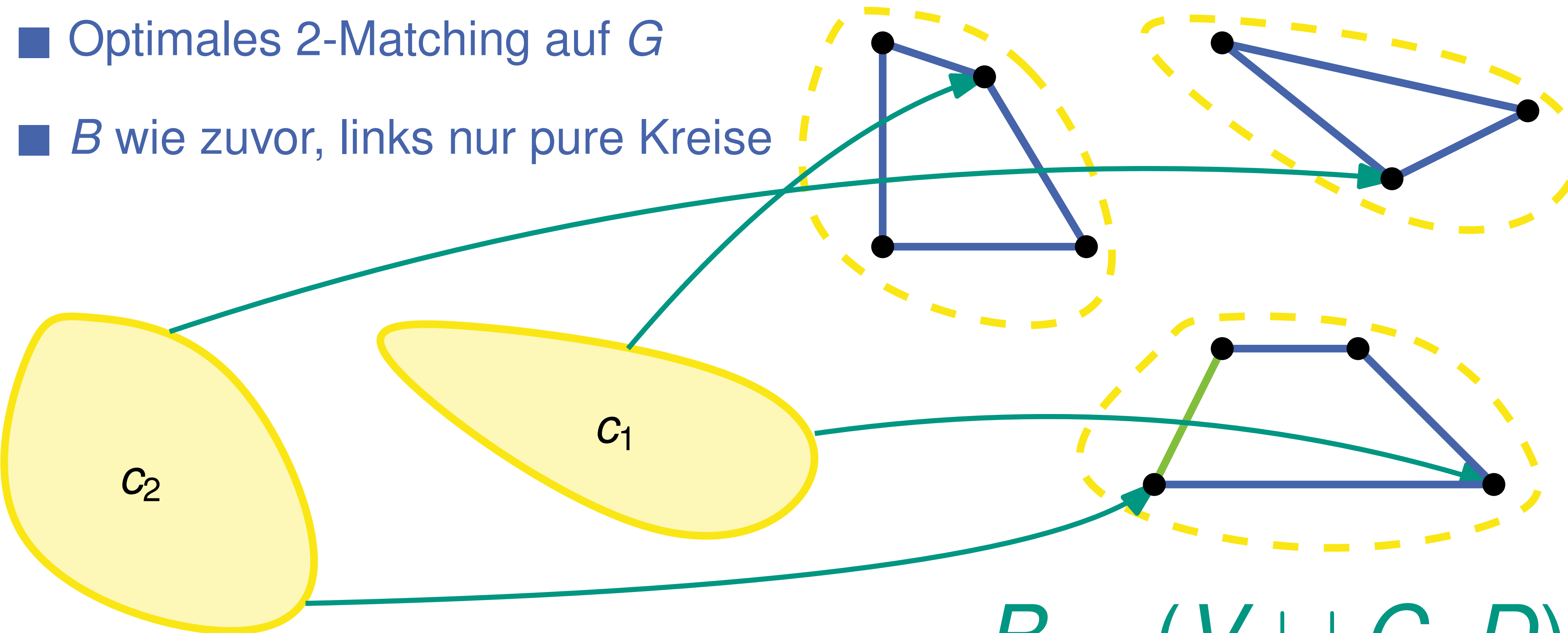
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise



$$B = (V \cup C, D)$$

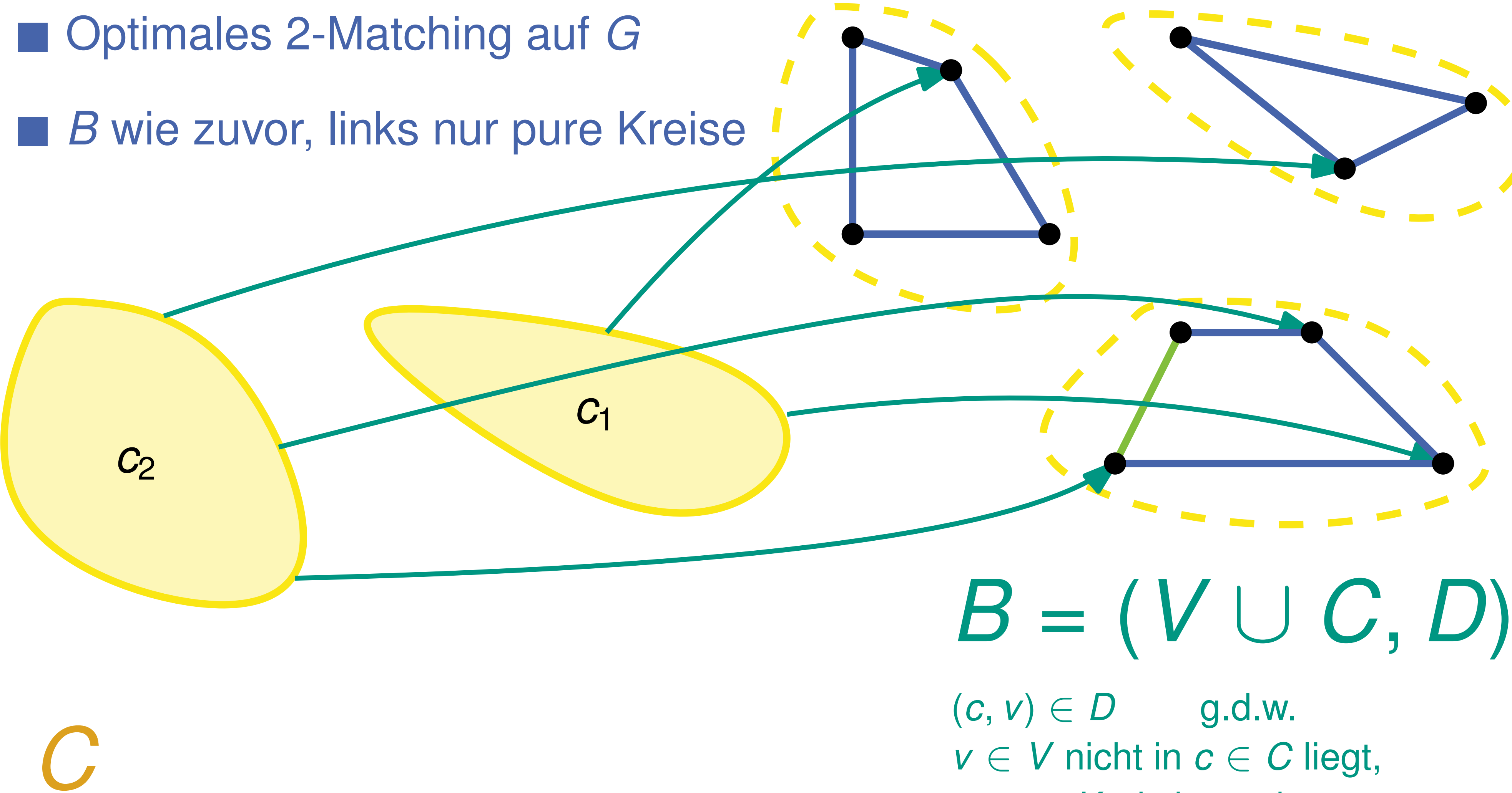
$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

C

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Optimales 2-Matching auf G
- B wie zuvor, links nur pure Kreise



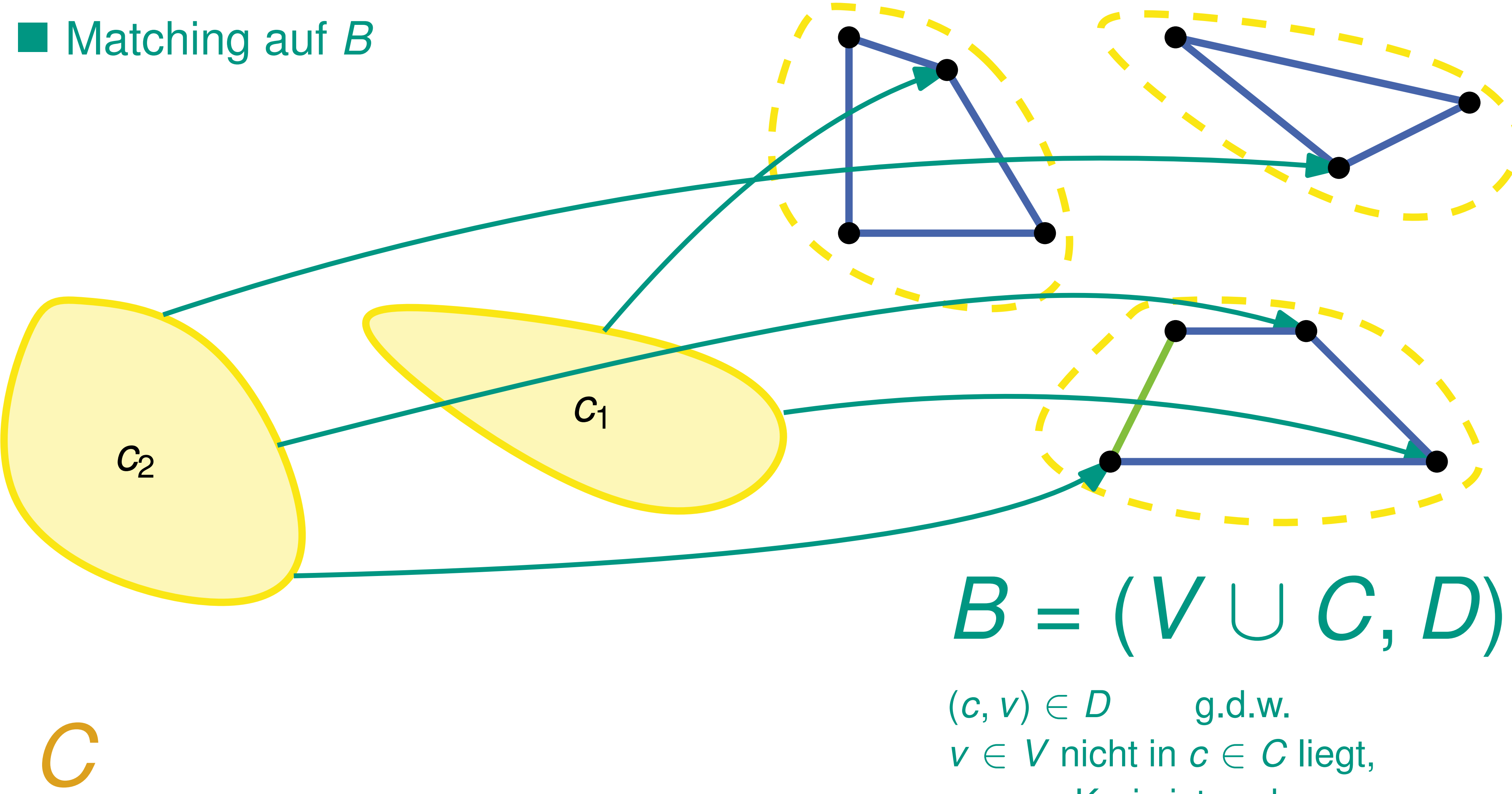
$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B



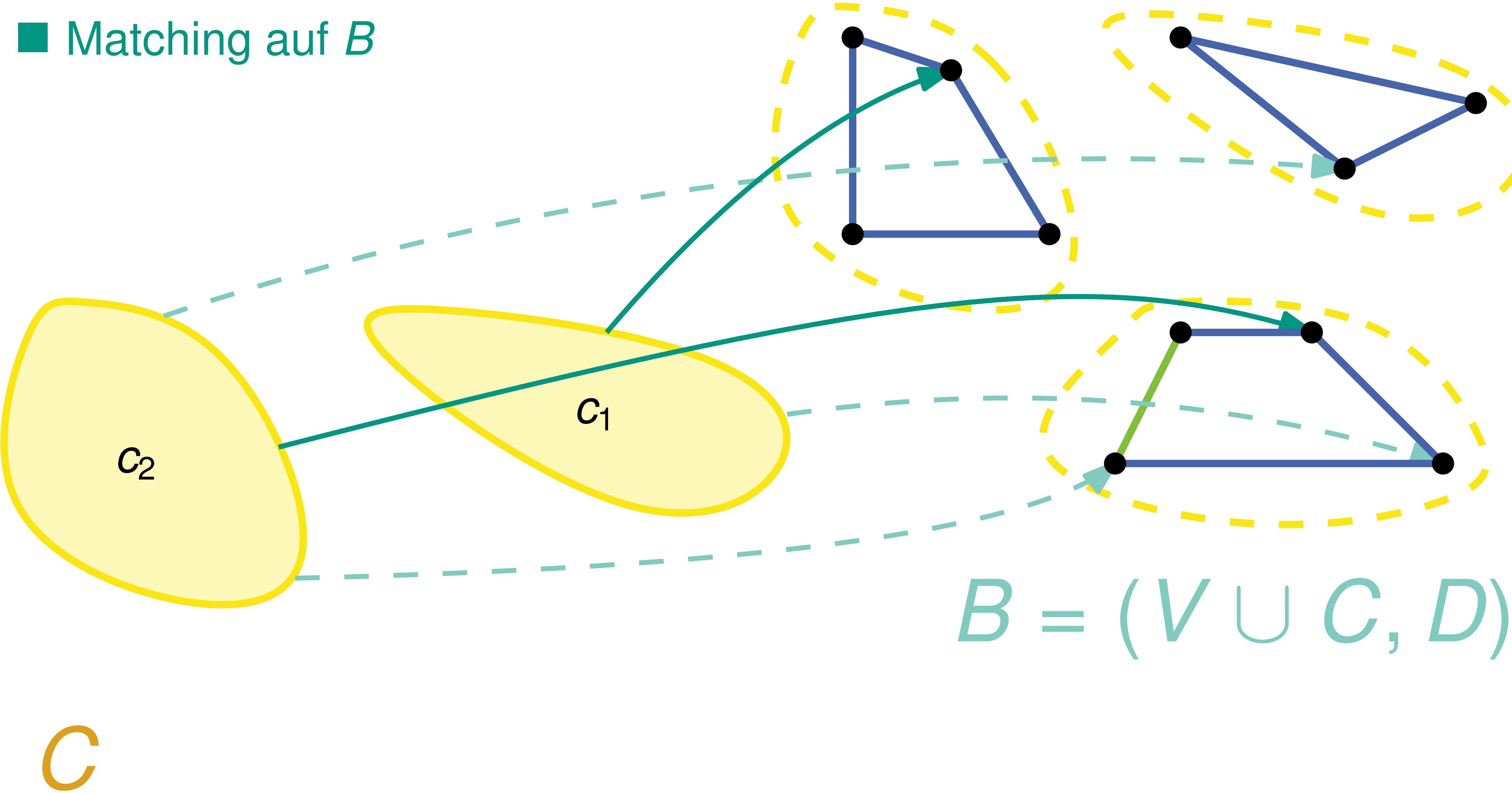
$$B = (V \cup C, D)$$

$(c, v) \in D$ g.d.w.
$v \in V$ nicht in $c \in C$ liegt,
c purer Kreis ist und
$u \in V$ in c existiert mit $\{u, v\} \in E'$

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B

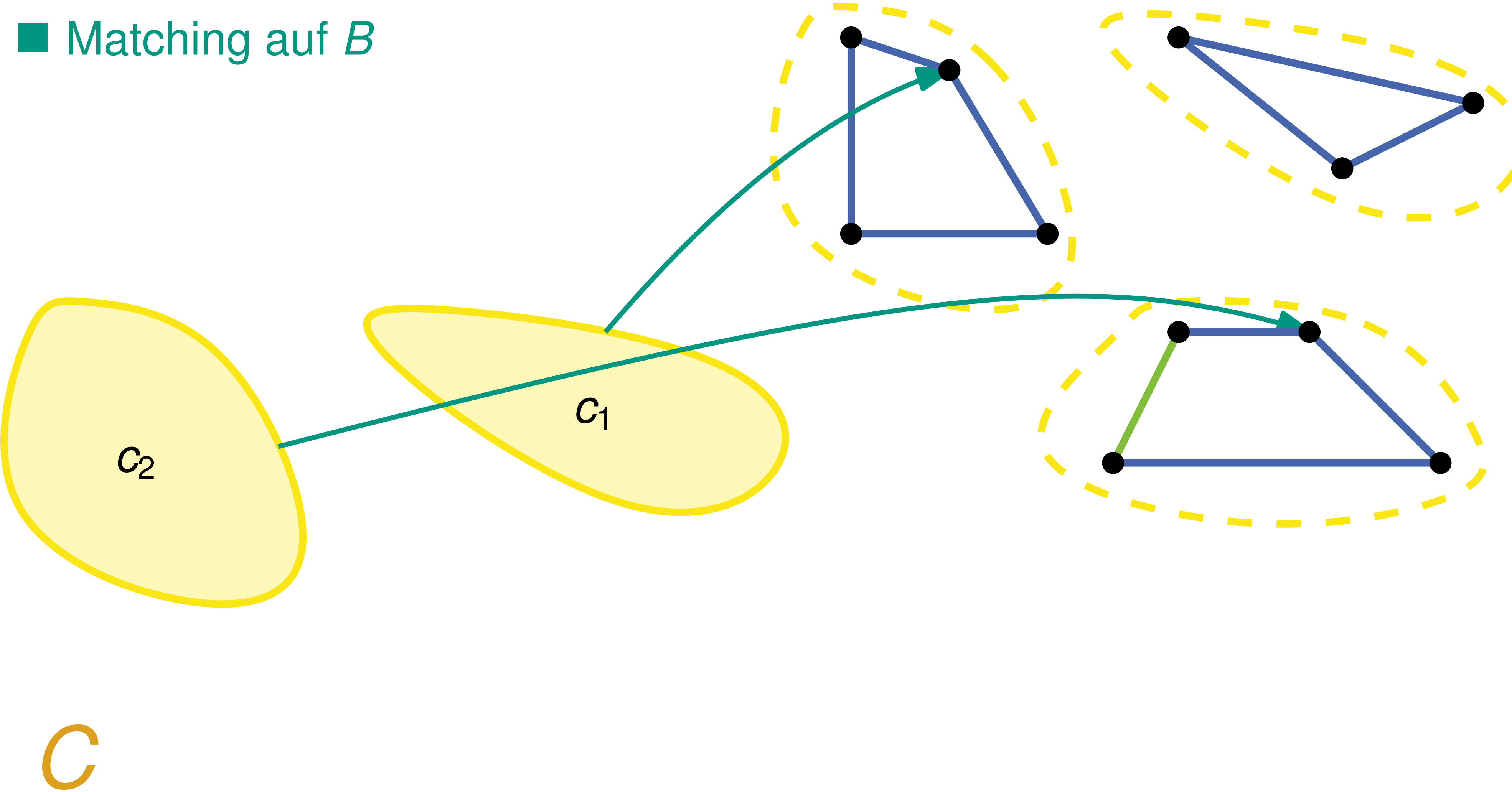


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B

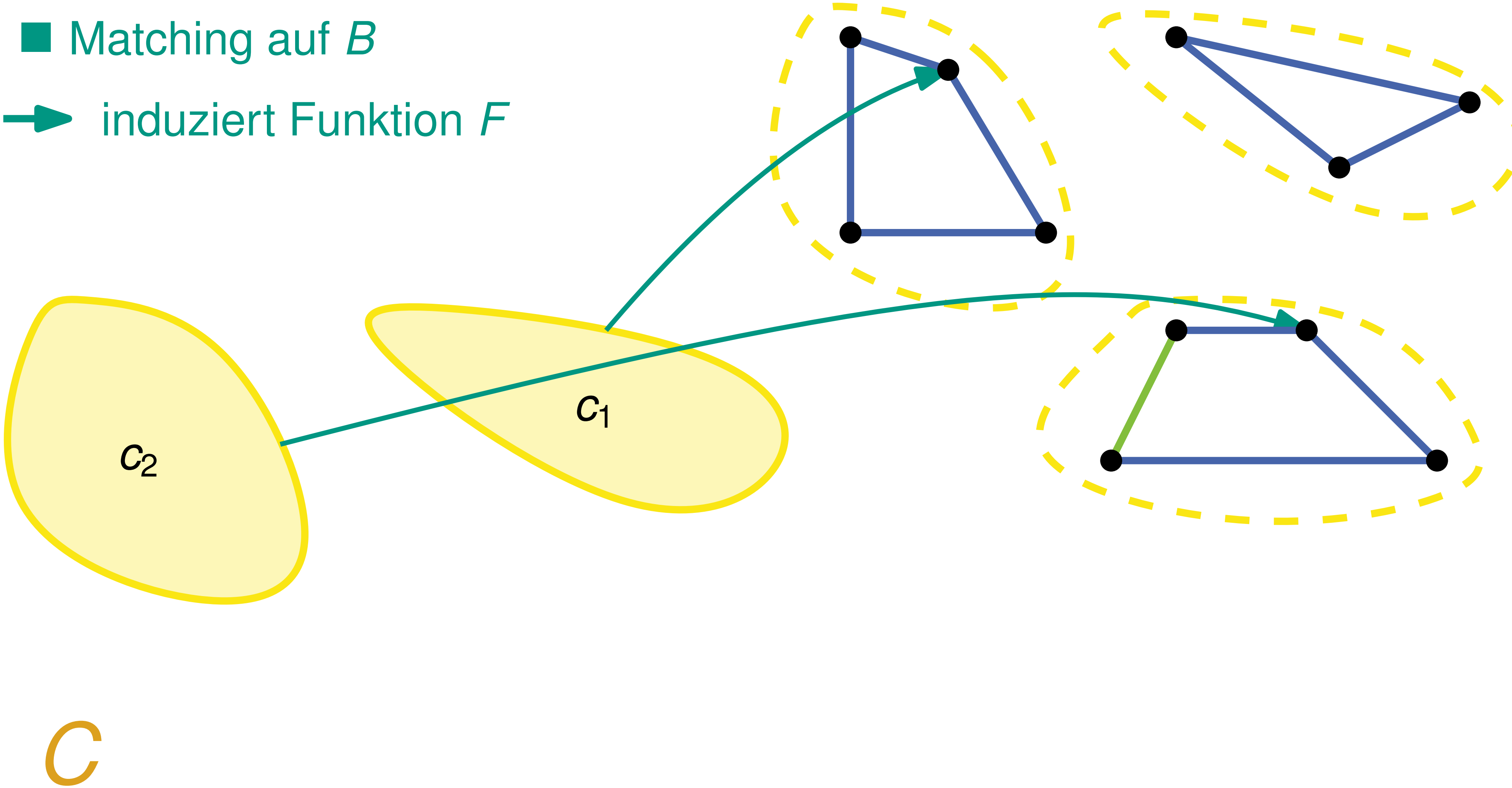


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B
- → induziert Funktion F

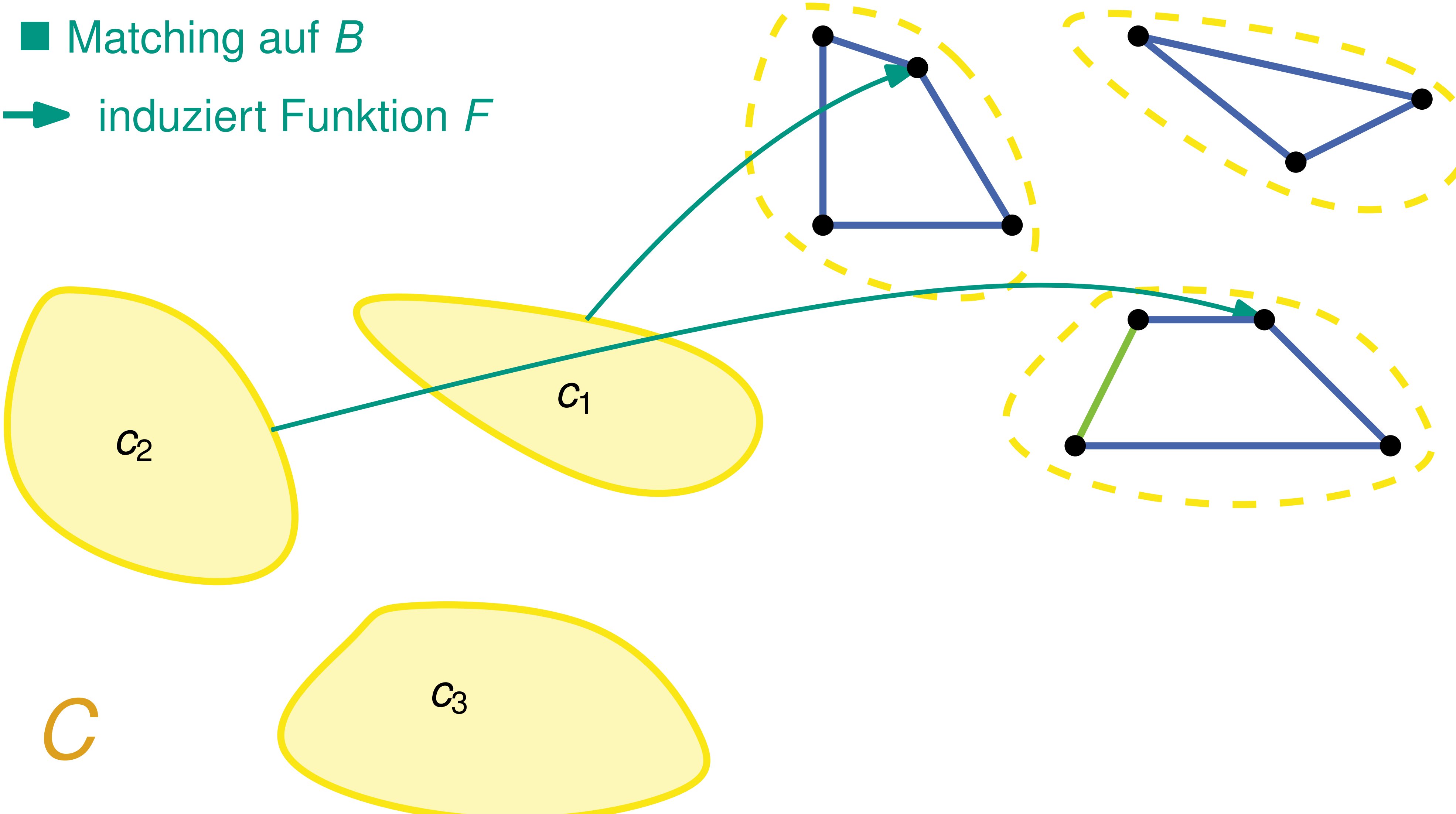


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B
- → induziert Funktion F

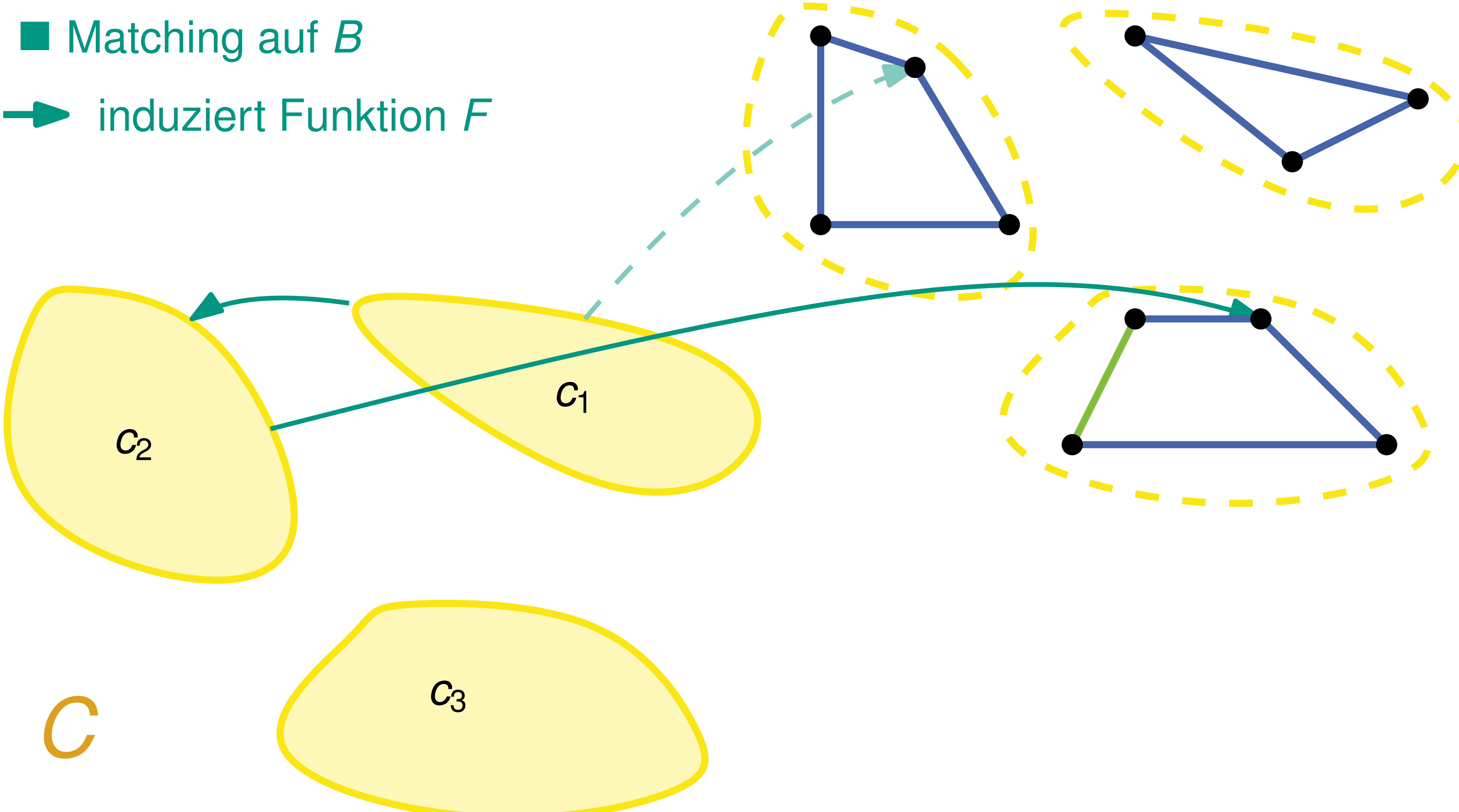


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B
- → induziert Funktion F

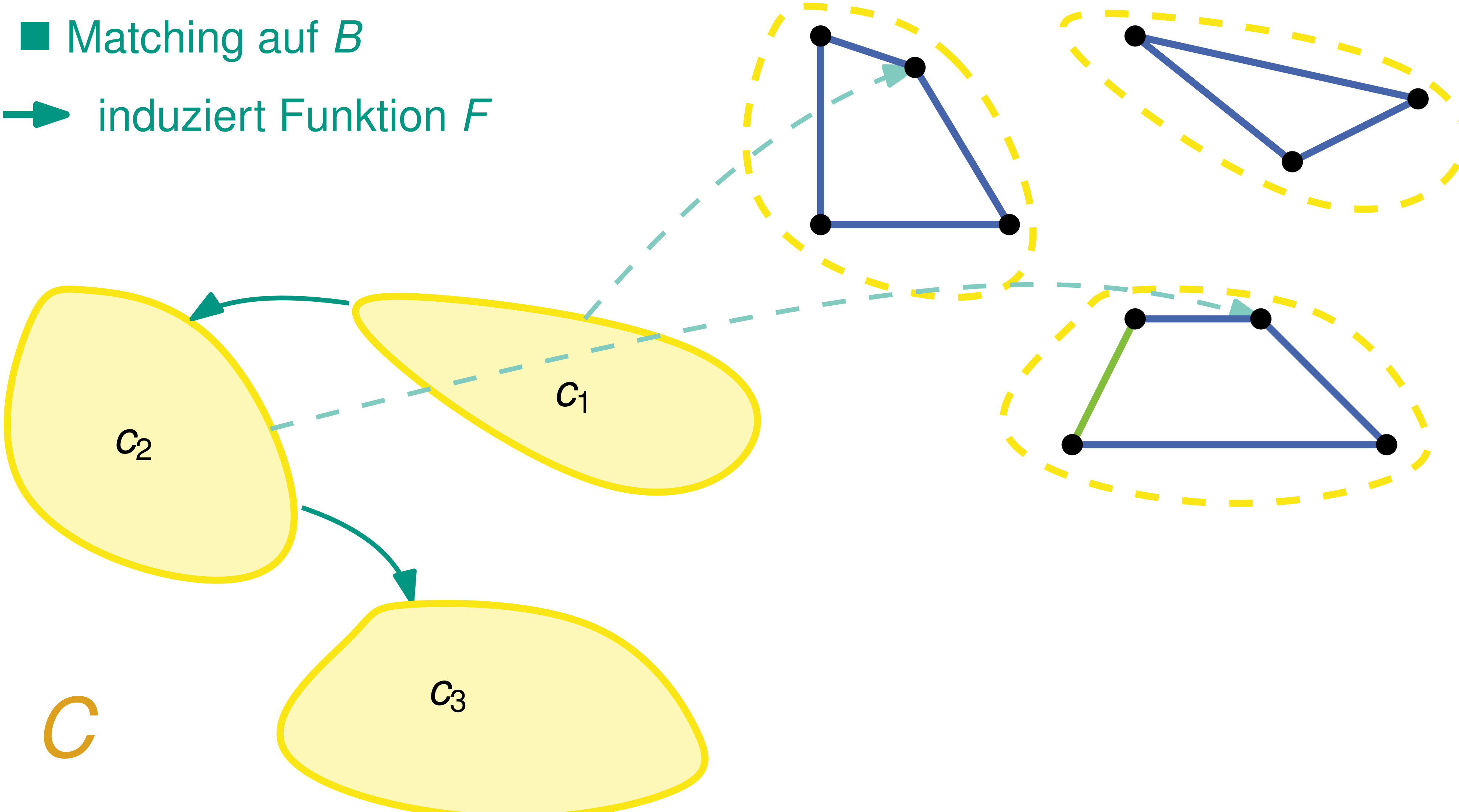


Institute of Theoretical Informatics
Algorithmics Group

APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B
- → induziert Funktion F

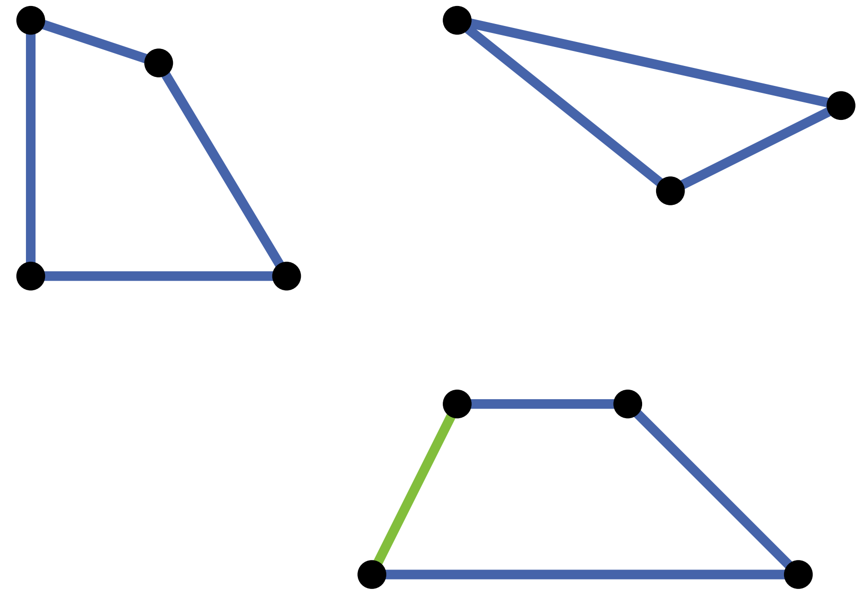


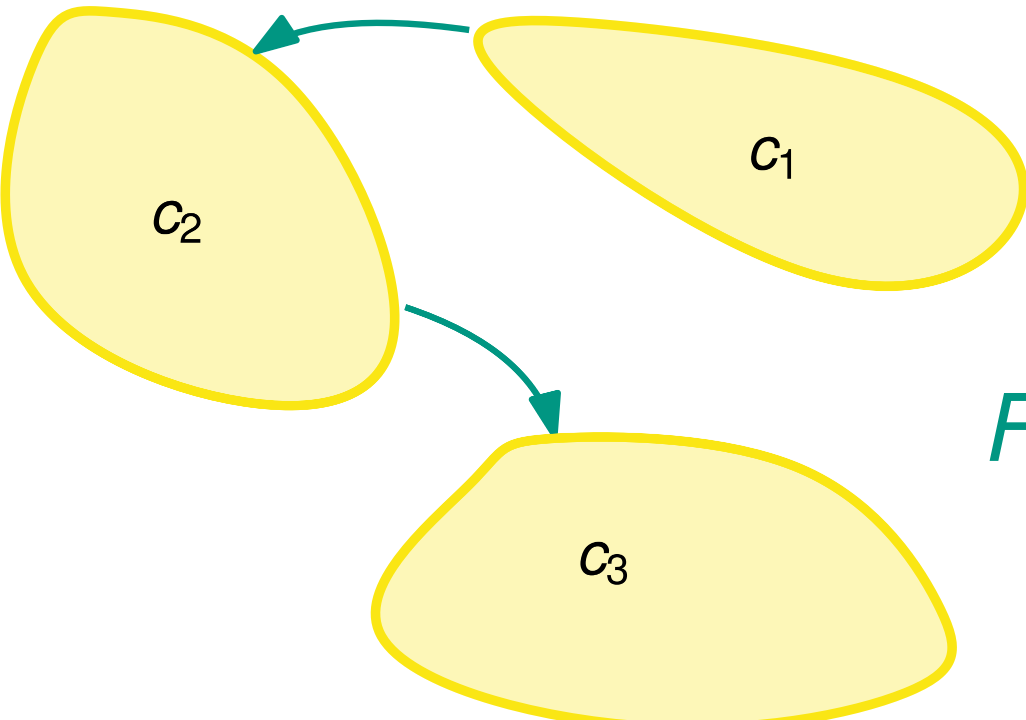
APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis

- Matching auf B

→ induziert Funktion F



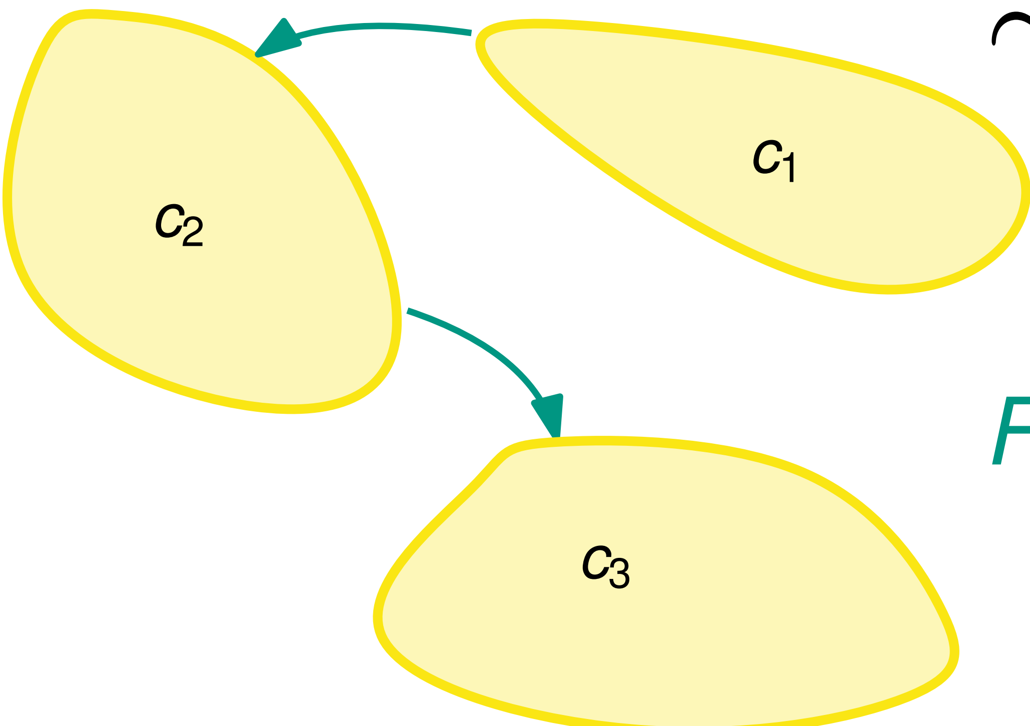


$$F = (C, A)$$

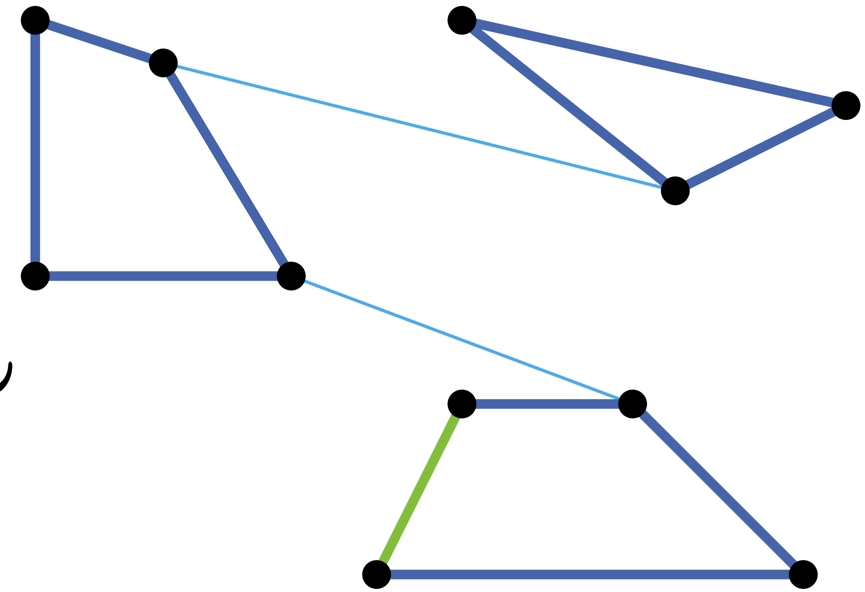
APPROXIMATIONSALGORITHMUS
FALL 2: G_d hat keinen Hamiltonkreis

Jede Kante von F ist mit einer Kante von G_d assoziiert!

- Finde aufspannenden Teilgraphen wie zuvor!



$\sim$

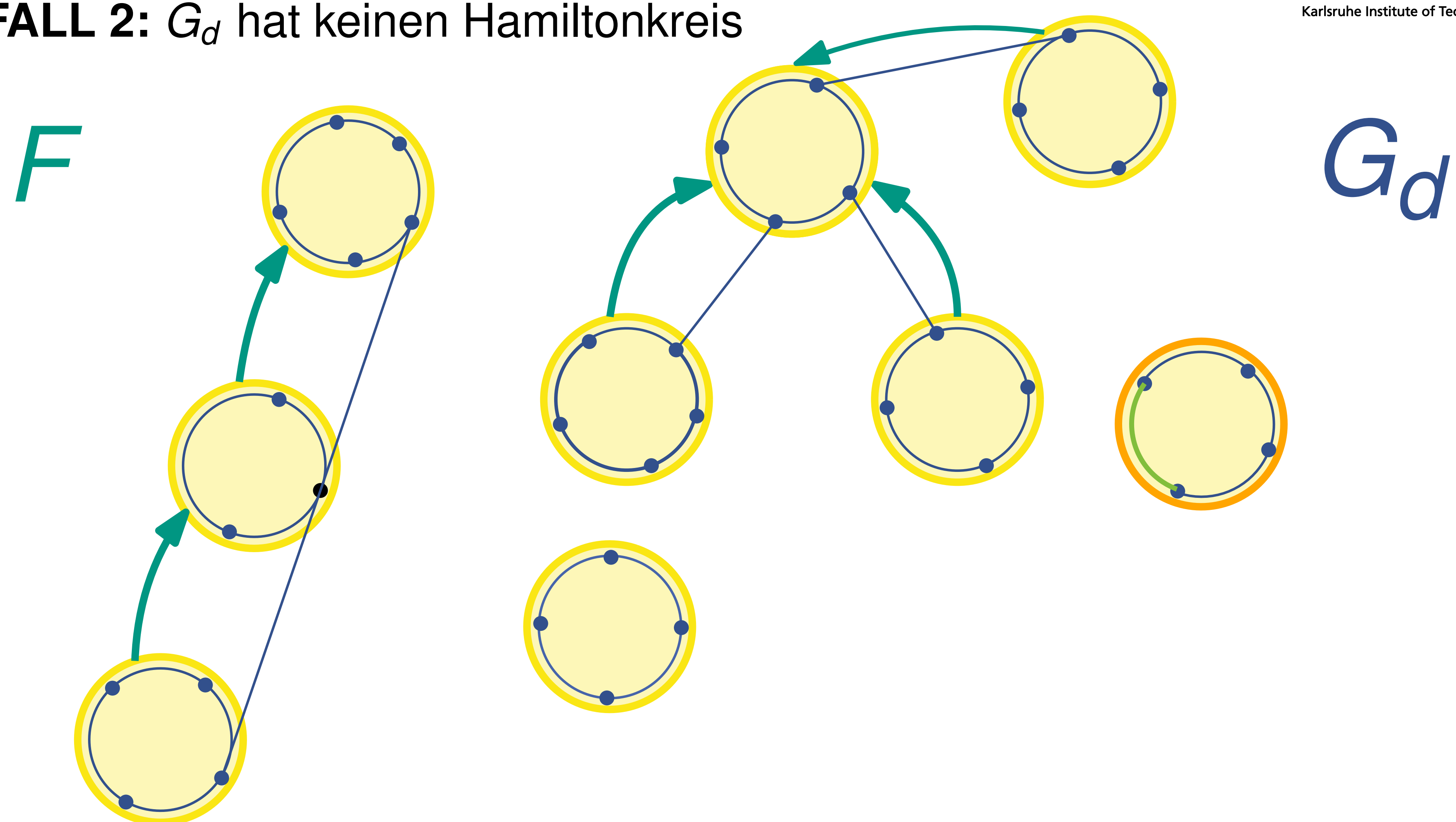


$F = (C, A)$

Institute of Theoretical Informatics
Algorithmics Group

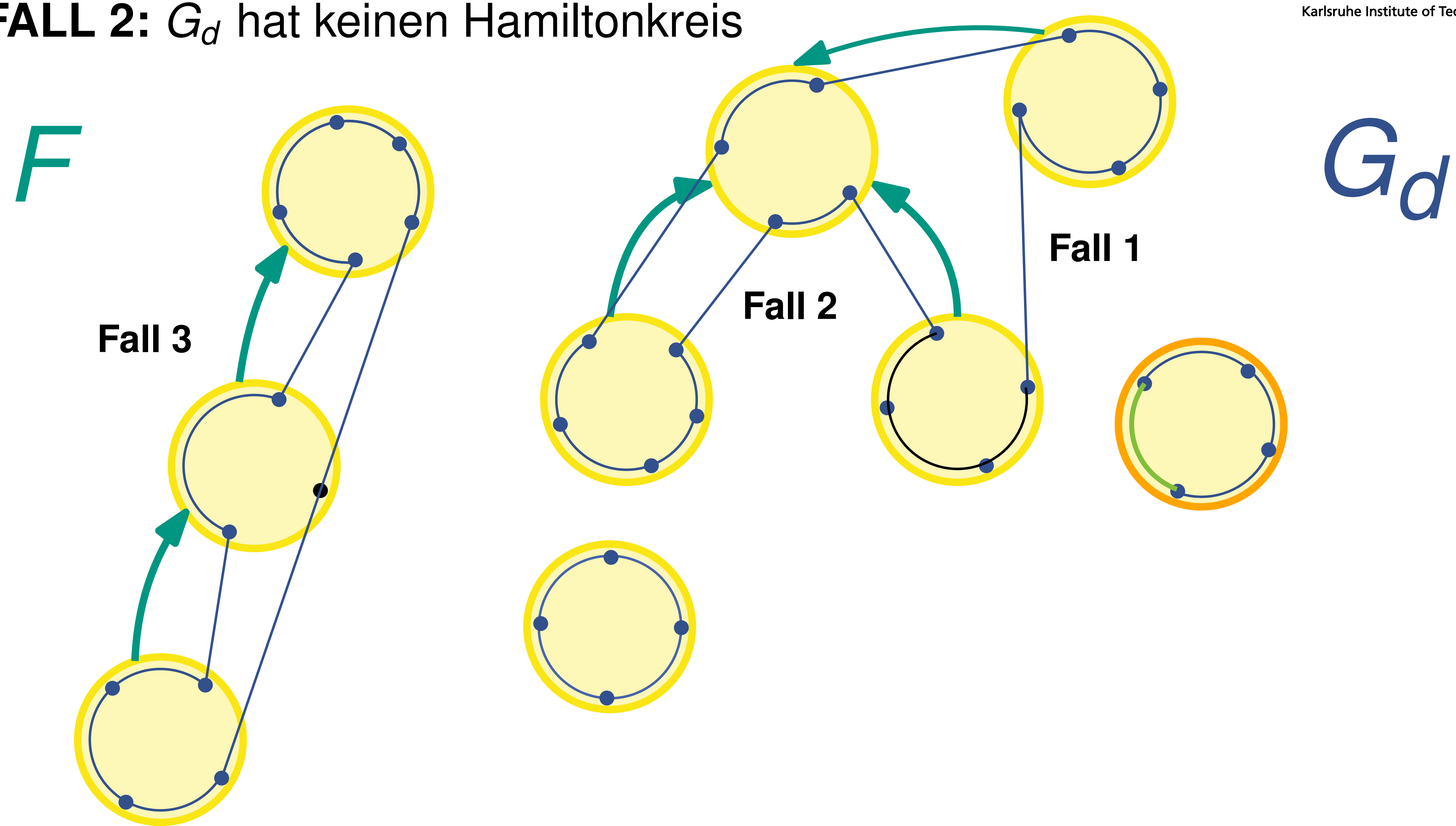
APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis



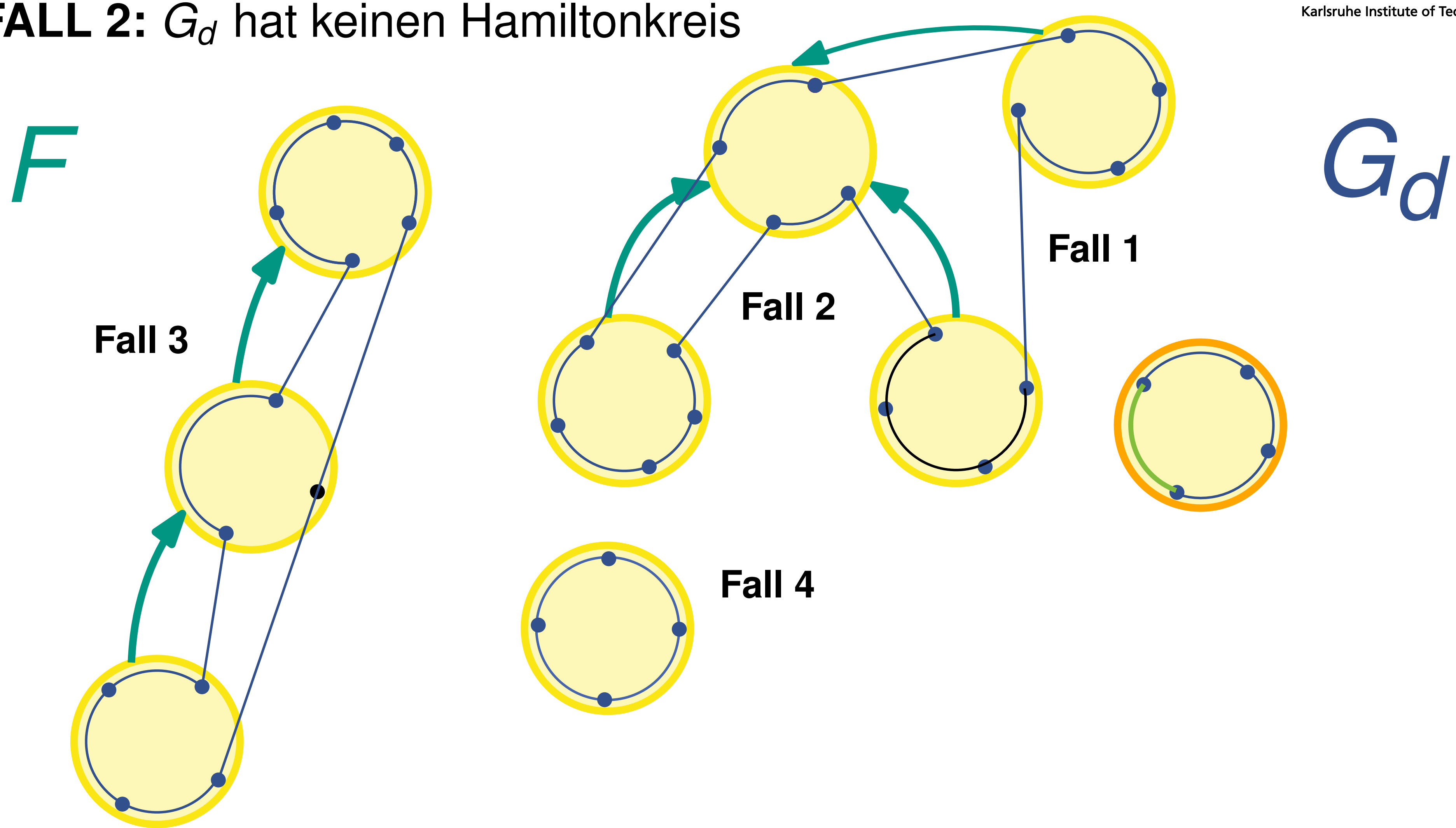
APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis



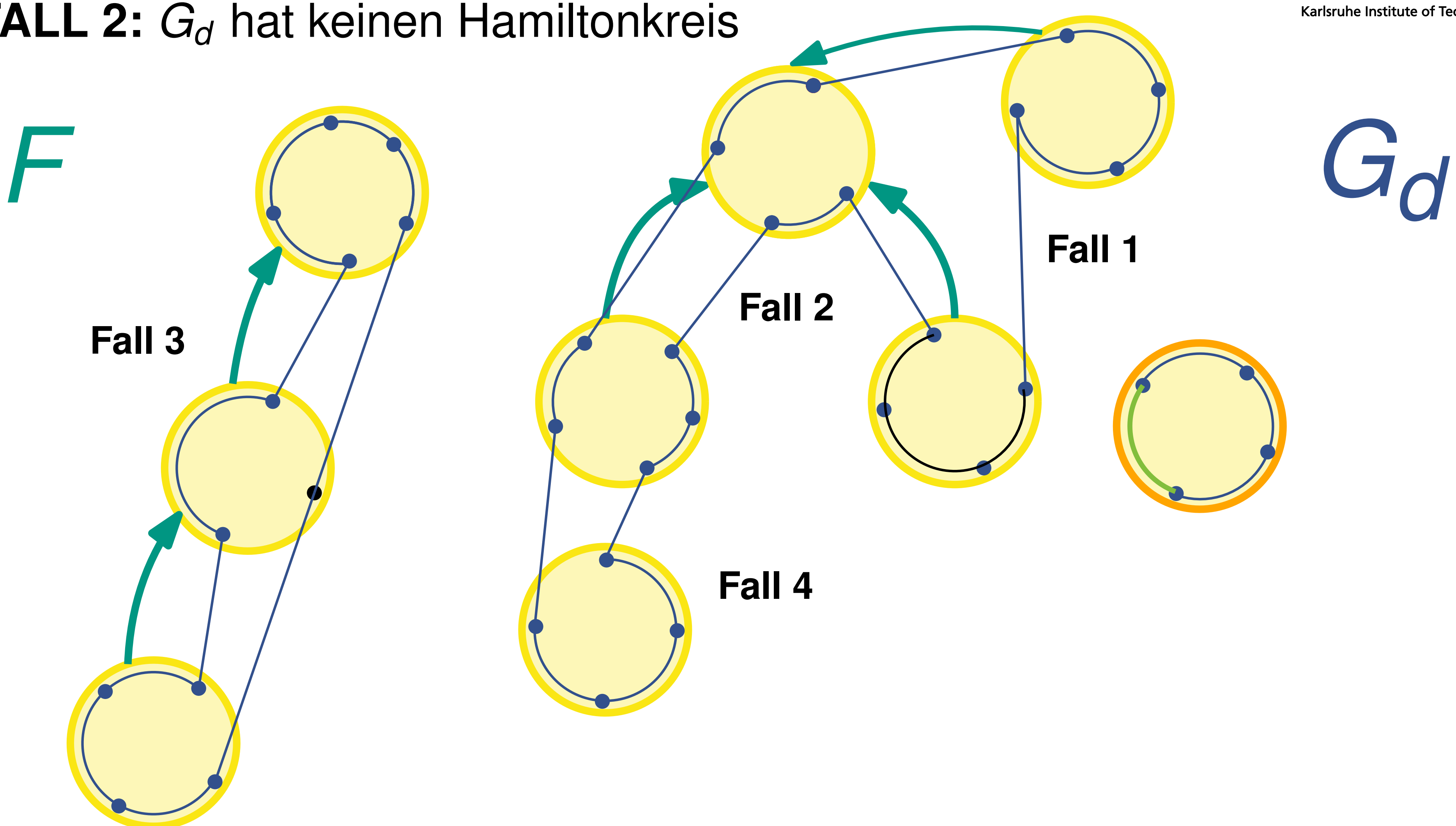
APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis



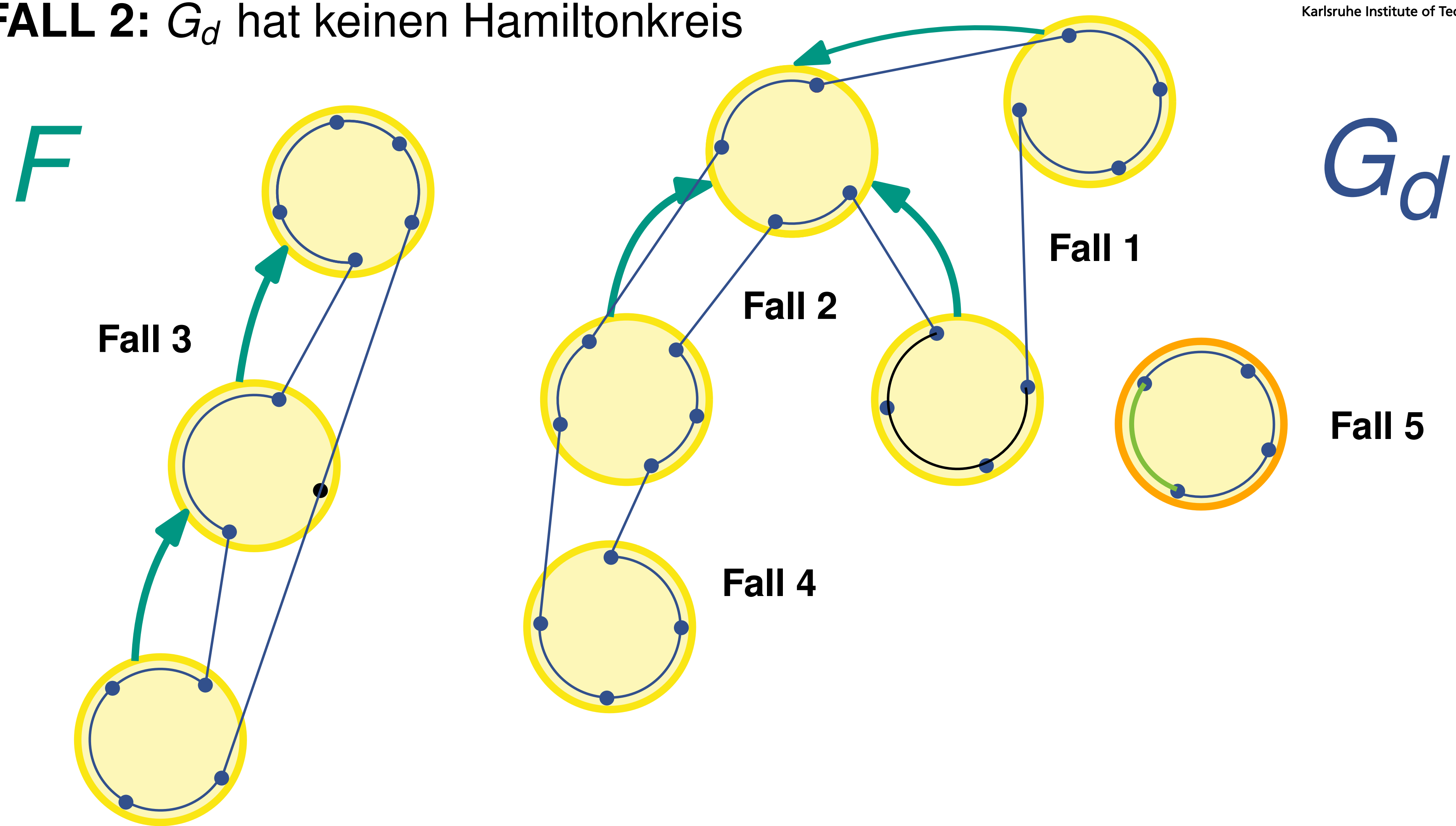
APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis



APPROXIMATIONSALGORITHMUS

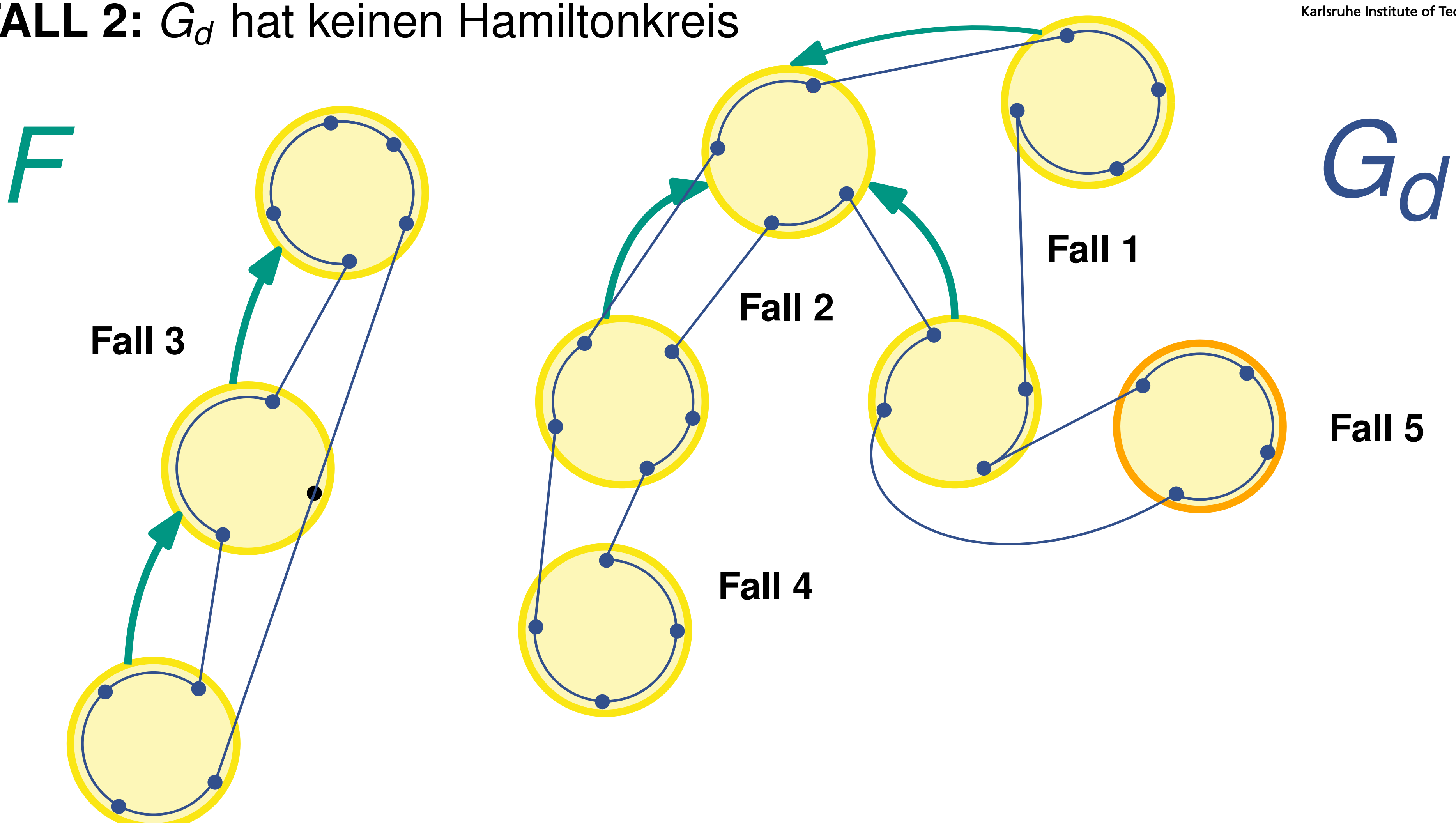
FALL 2: G_d hat keinen Hamiltonkreis



Institute of Theoretical Informatics
Algorithmics Group

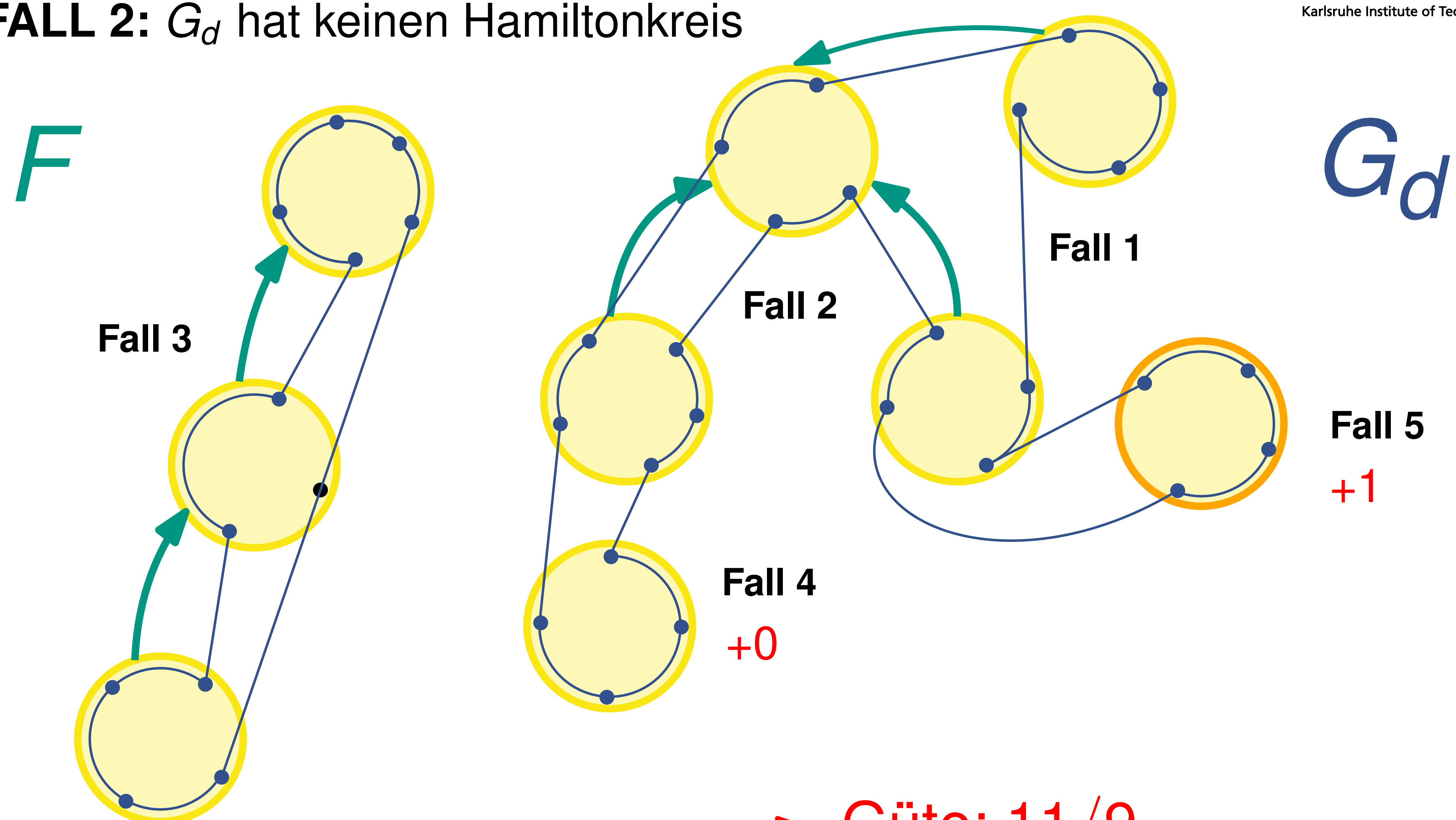
APPROXIMATIONSALGORITHMUS

FALL 2: G_d hat keinen Hamiltonkreis



APPROXIMATIONSALGORITHMUS KOSTEN

FALL 2: G_d hat keinen Hamiltonkreis



→ Güte: $11/9$

Vielen Dank für eure Aufmerksamkeit!