Algorithmen zur Visualisierung von Graphen

Rückblick und gitterbasiertes Graphenzeichnen mit ILP

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Tamara Mchedlidze · **Martin Nöllenburg**

11.02.2015

Rückblick der VL-Themen

1. Einführung
2. Baumlayouts
3. Serien-parallele Graphen
4. Geradlinige planare Zeichnungen: Shift- und Realizer-Methode
5. Datenstrukturen für planare Einbettungen
6. Kontaktrepräsentationen planarer Graphen
7. Lagenlayouts (Sugiyama-Framework)
8. Metro Map Layout
9. Knickminimierung und Kompaktierung in orthogonalen Layouts (TSM-Framework)
10. Aufwärtsplanarität
11. Kräftebasierte Algorithmen
12. Inkrementelle orthogonale Zeichnungen

Rückblick der VL-Themen

1. Einführung
2. Baumlayouts
3. Serien-parallele Graphen
4. Geradlinige planare Zeichnungen:
   Shift- und Realizer-Methode
5. Datenstrukturen für planare Einbettungen
6. Kontaktrepräsentationen planarer Graphen
7. Lagenlayouts (Sugiyama-Framework)
8. Metro Map Layout
9. Knickminimierung und Kompaktierung in orthogonalen
   Layouts (TSM-Framework)
10. Aufwärtsplanarität
11. Kräftebasierte Algorithmen
12. Inkrementelle orthogonale Zeichnungen

Welche Aspekte haben wir jeweils behandelt? Welche Querbezüge gibt es?

Prüfungsmodalitäten (Master)

Mündliche Prüfung (20 Minuten)

- Termin 1: 25.2.2015 ab 16:15 Uhr
- Termin 2: 25.3.2015 ab 16:15 Uhr
- Termin 3: 15.4.2015 ab 16:15 Uhr
- Anmeldung im Sekretariat (lilian.beckert@kit.edu)

Inhalt: Stoff aus Vorlesung und Übung (Skript, Literatur & Folien/Mitschrieb nutzen – nur was besprochen wurde)

Ziele:

- vorgestellte Algorithmen und Layoutprobleme kennen, erklären und analysieren
- Problemmodellierung/Ästhetikkriterien, Einordnung, Vor-/Nachteile, Beweisideen, ...

Prüfer: beide Dozenten gemeinsam, Tamara: Englisch

Prüfungsmodalitäten (Master)

Mündliche Prüfung (20 Minuten)

- Termin 1: 25.2.2015 ab 16:15 Uhr
- Termin 2: 25.3.2015 ab 16:15 Uhr
- Termin 3: 15.4.2015 ab 16:15 Uhr
- Anmeldung im Sekretariat (lilian.beckert@kit.edu)

Inhalt: Stoff aus Vorlesung und Übung (Skript, Literatur & Folien/Mitschrieb nutzen – nur was besprochen wurde)

außer Blatt 6

Ziele:

- vorgestellte Algorithmen und Layoutprobleme kennen, erklären und analysieren
- Problemmodellierung/Ästhetikkriterien, Einordnung, Vor-/Nachteile, Beweisideen, ...

Prüfer: beide Dozenten gemeinsam, Tamara: Englisch

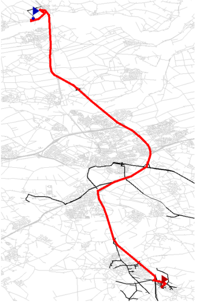
Werbung für Sommersemester 2015

- Praktikum Graphenvisualisierung (Master)

  aufbauend auf diese VL
  Teilnahme am Graph Drawing Contest 2015

- Algorithmische Kartografie (Master)

  vermutlich dienstags 9:45 Uhr



- Algorithmen für Routenplanung (Master)

  Montag 14:00 – 15:30
  Mittwoch 11:30 – 13:00

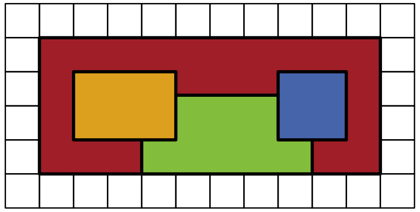
- Masterarbeiten und HiWi-Jobs

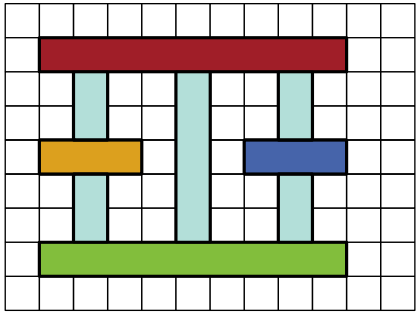
  Themen in allen Forschungsfeldern des Lehrstuhls
  einfach Mitarbeiter ansprechen

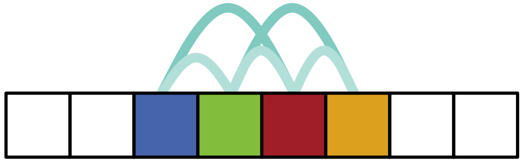
`i11www.iti.kit.edu`

Gitterbasiertes Graphenzeichnen mit ILP

basiert auf [Biedl et al. ’14]

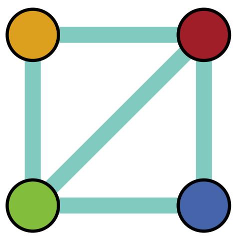


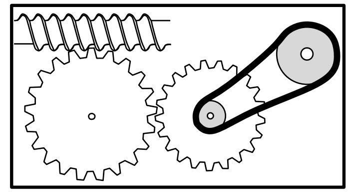




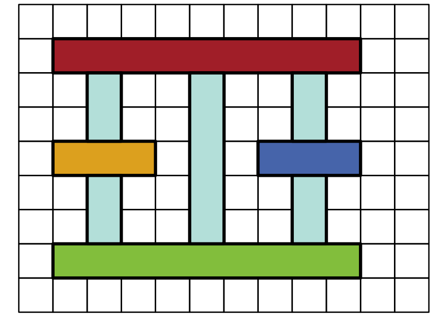
Motivation

Solve grid-based graph drawing problems in a unified framework:

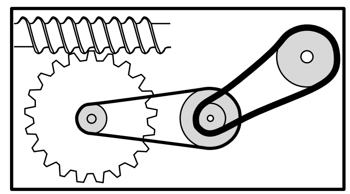




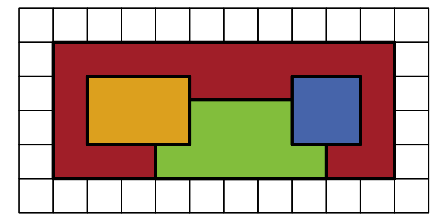
visibility representation



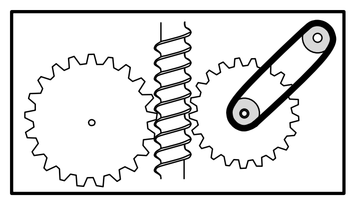
minimize
width
NP-hard



boxicity



recognition
NP-hard



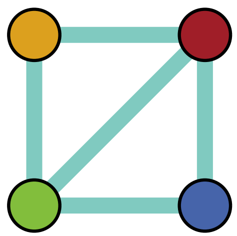
st-orientation

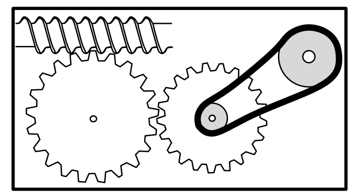
st-orientation
minimize longest *st*-path
NP-hard

pathwidth,
bandwidth,
...

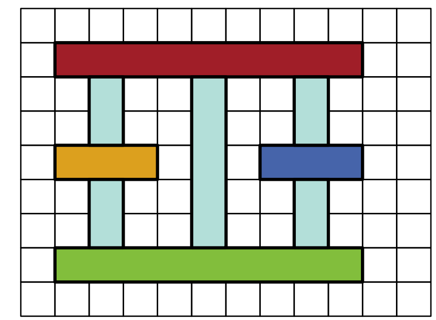
Motivation

Solve grid-based graph drawing problems in a unified framework:

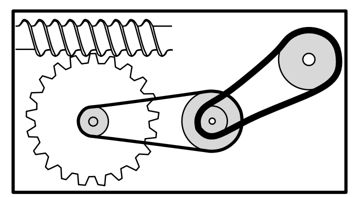




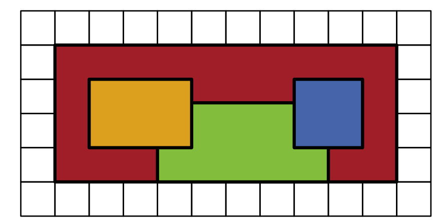
visibility representation



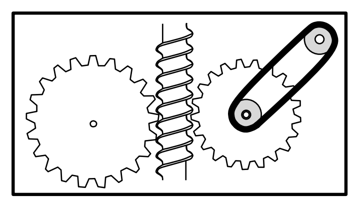
minimize width NP-hard



boxicity



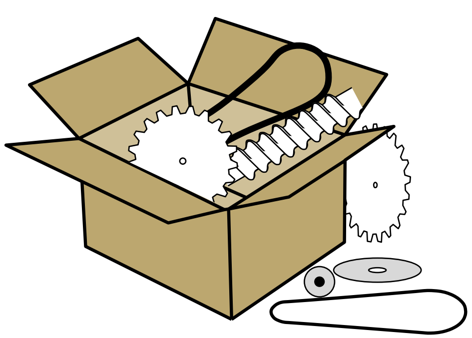
recognition NP-hard



st-orientation

st-orientation
minimize longest *st*-path
NP-hard

pathwidth,
bandwidth,
. . .



Apply Integer Linear Programming (ILP):

- Collection of general constraints.
- Solving problem = assemble constraints.

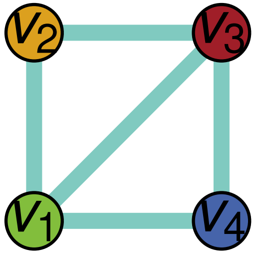
Approach

Given: Grid R, objects, constraints

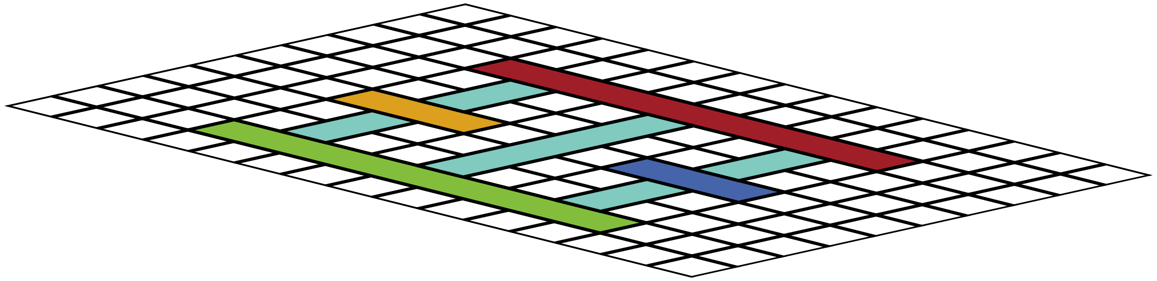
Find: Box on R for each object such that constraints are satisfied.

Example:

Visibility Representation



Vertices = horizontal boxes.
Edges = vertical boxes.



Approach

Given: Grid R, objects, constraints

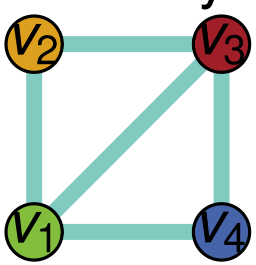
Find: Box on R for each object such that constraints are satisfied.

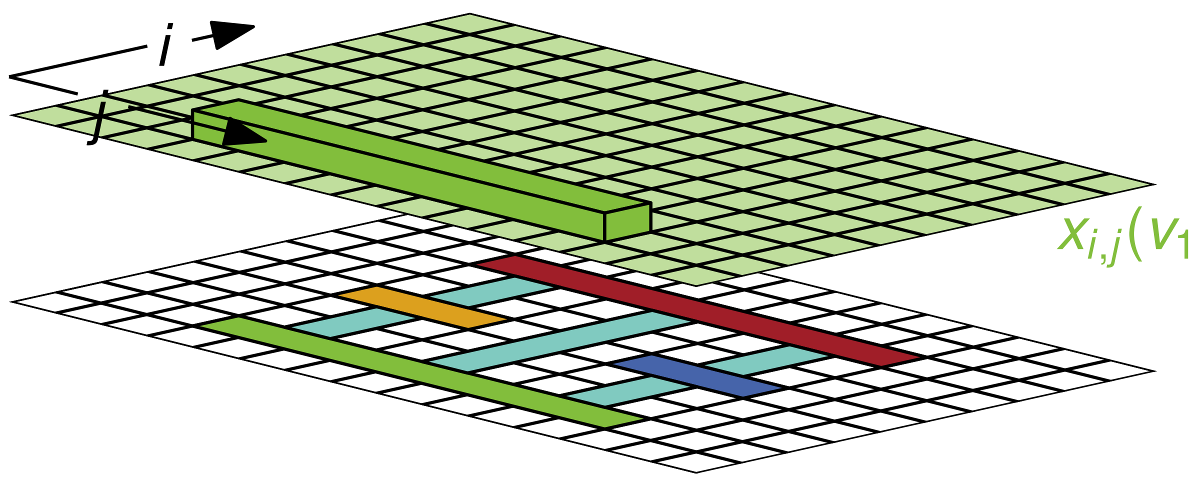
Representation of single box B:

- Grid of binary variables $x(B)_{i,j}$
- Meaning: $x(B)_{i,j} = 1$ iff grid point (i,j) belongs to B.

Example:

Visibility Representation





Vertices = horizontal boxes.
Edges = vertical boxes.

Approach

Given: Grid R, objects, constraints

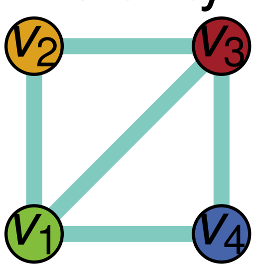
Find: Box on R for each object such that constraints are satisfied.

Representation of single box B:

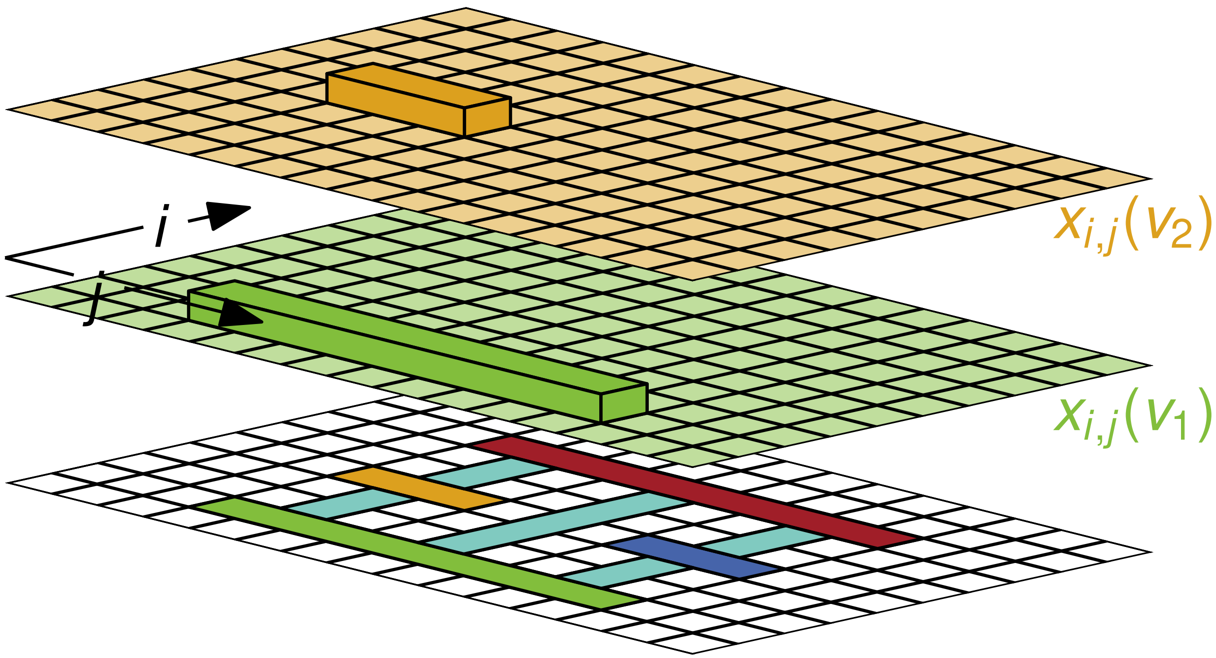
- Grid of binary variables $x(B)_{i,j}$
- Meaning: $x(B)_{i,j} = 1$ iff grid point (i,j) belongs to B.

Example:

Visibility Representation



Vertices = horizontal boxes.
Edges = vertical boxes.



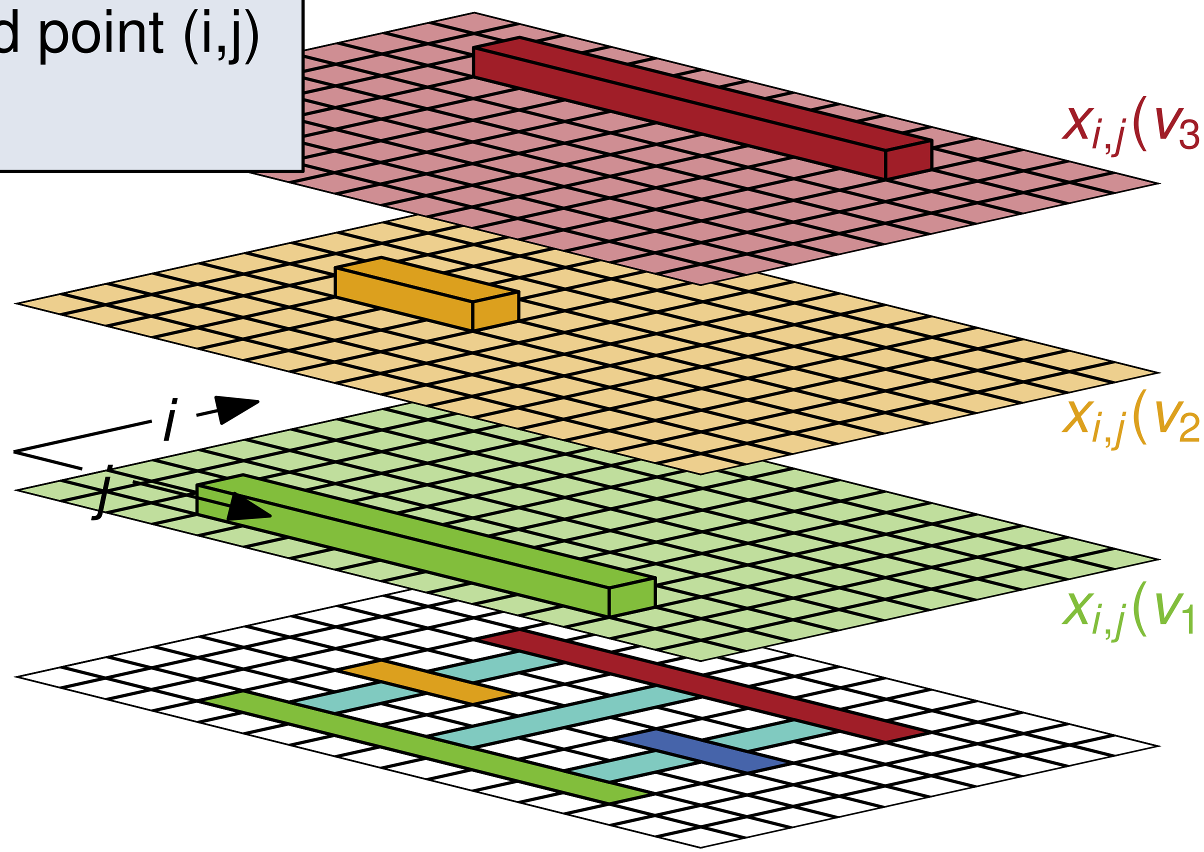
Approach

Given: Grid R, objects, constraints

Find: Box on R for each object such that constraints are satisfied.

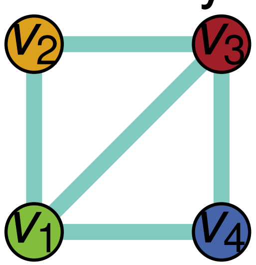
Representation of single box B:

- Grid of binary variables $x(B)_{i,j}$
- Meaning: $x(B)_{i,j} = 1$ iff grid point (i,j) belongs to B.



Example:

Visibility Representation



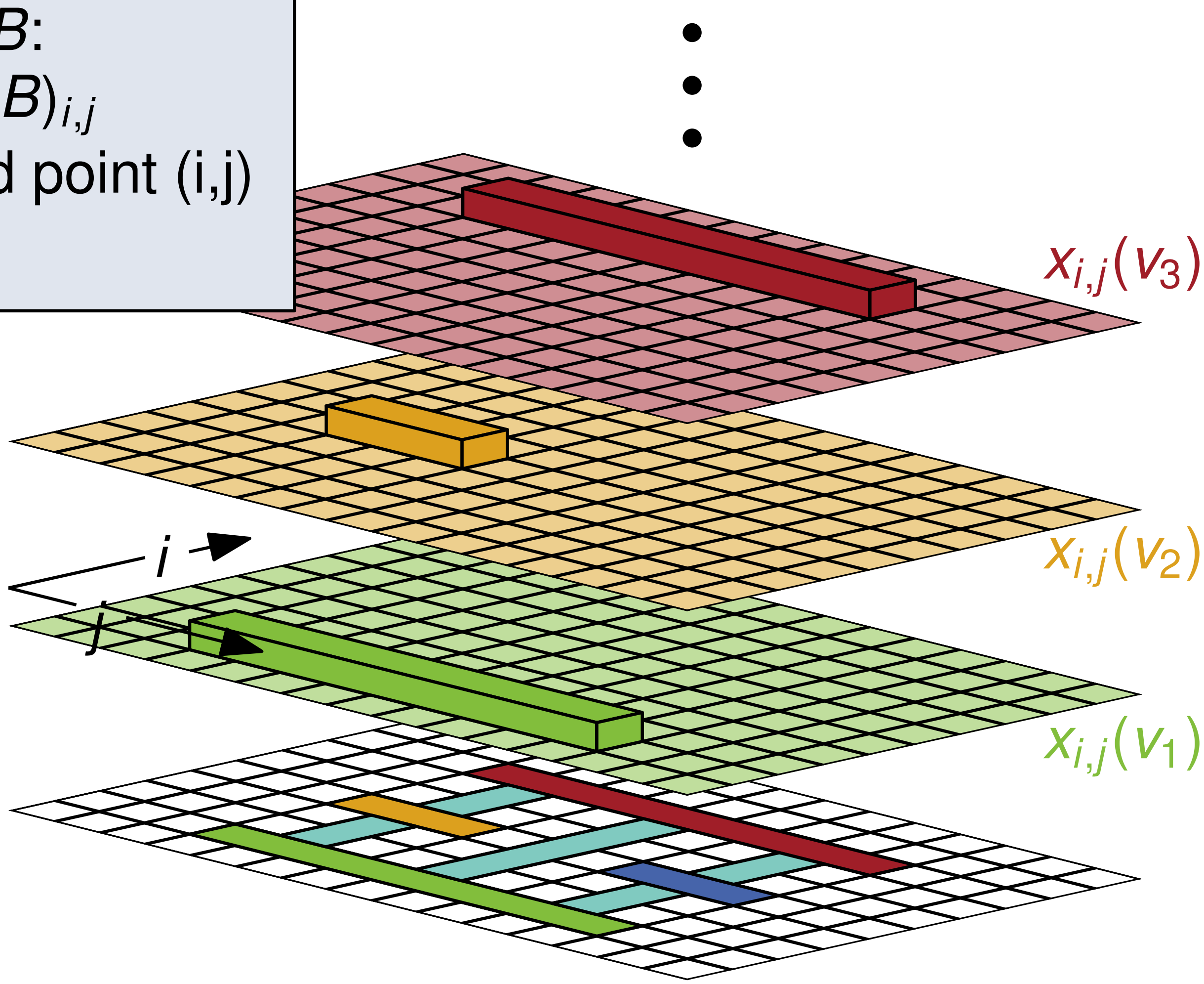
Vertices = horizontal boxes.
Edges = vertical boxes.

Approach

Given: Grid R, objects, constraints

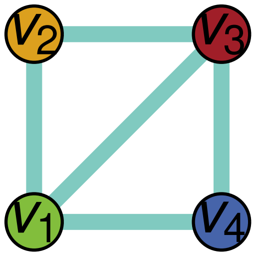
Find: Box on R for each object such that constraints are satisfied.

Representation of single box B:
- Grid of binary variables $x(B)_{i,j}$
- Meaning: $x(B)_{i,j} = 1$ iff grid point (i,j) belongs to B.



Example:

Visibility Representation



Vertices = horizontal boxes.
Edges = vertical boxes.

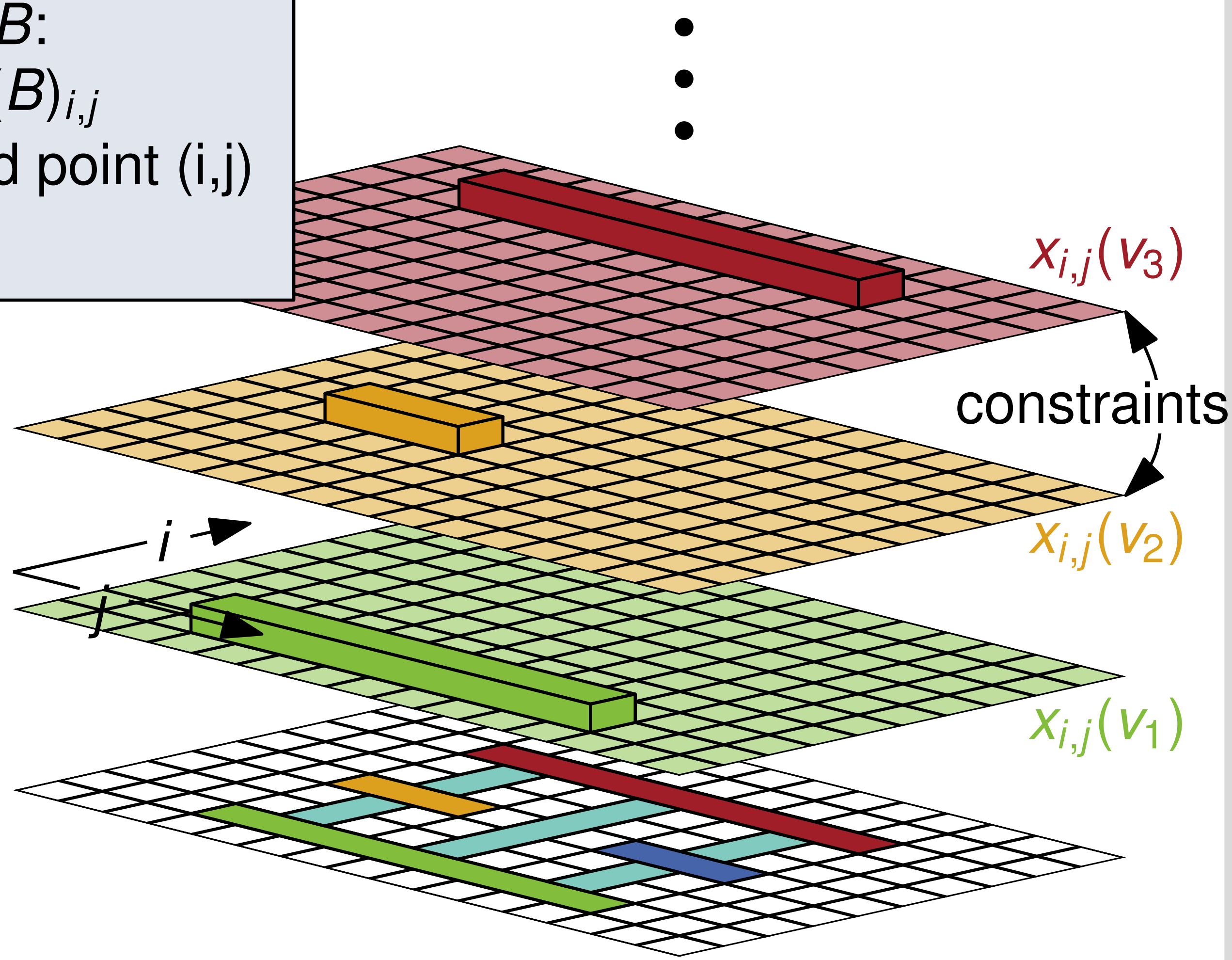
Approach

Given: Grid R, objects, constraints

Find: Box on R for each object such that constraints are satisfied.

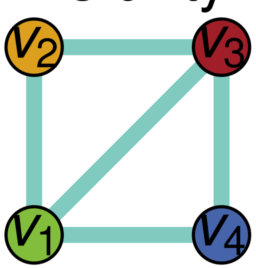
Representation of single box B:

- Grid of binary variables $x(B)_{i,j}$
- Meaning: $x(B)_{i,j} = 1$ iff grid point (i,j) belongs to B.



Example:

Visibility Representation

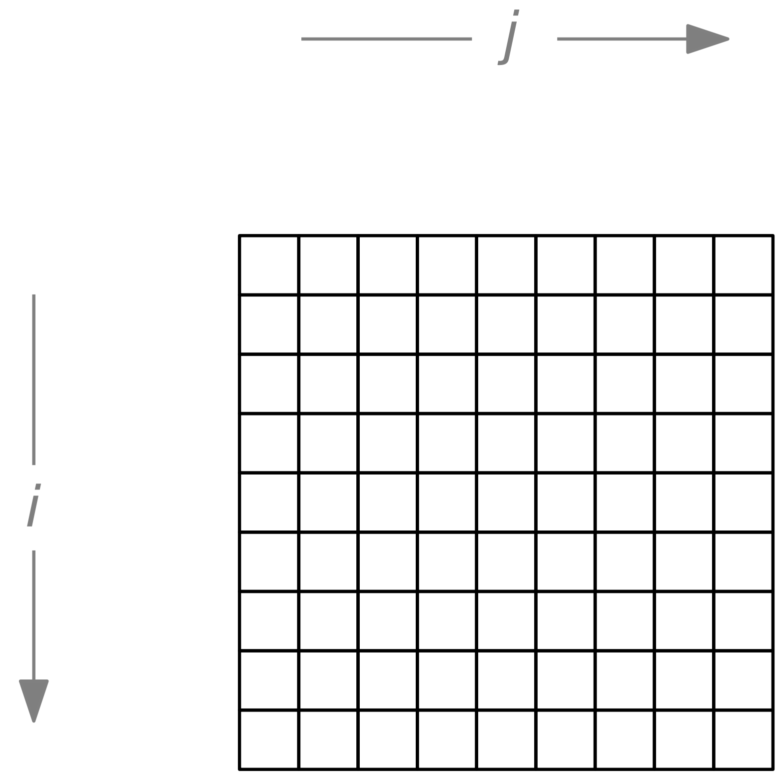


Vertices = horizontal boxes.
Edges = vertical boxes.

Representing Boxes

Represent box B by ILP constraints (here in 2D).

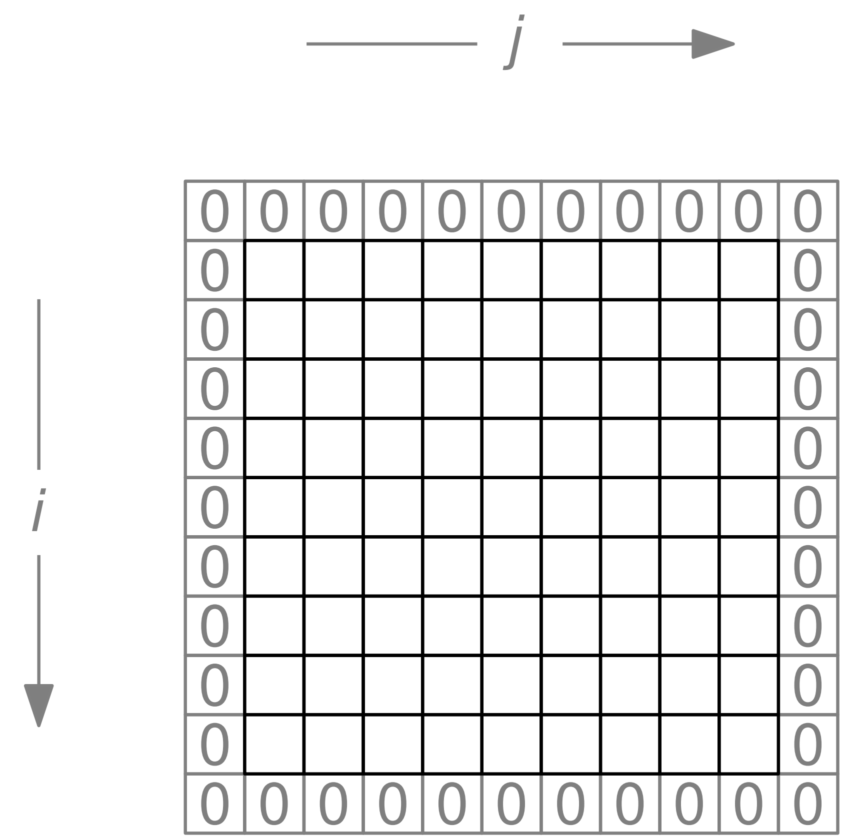
Grid of binary variables $x_{i,j}$



Representing Boxes

Represent box B by ILP constraints (here in 2D).

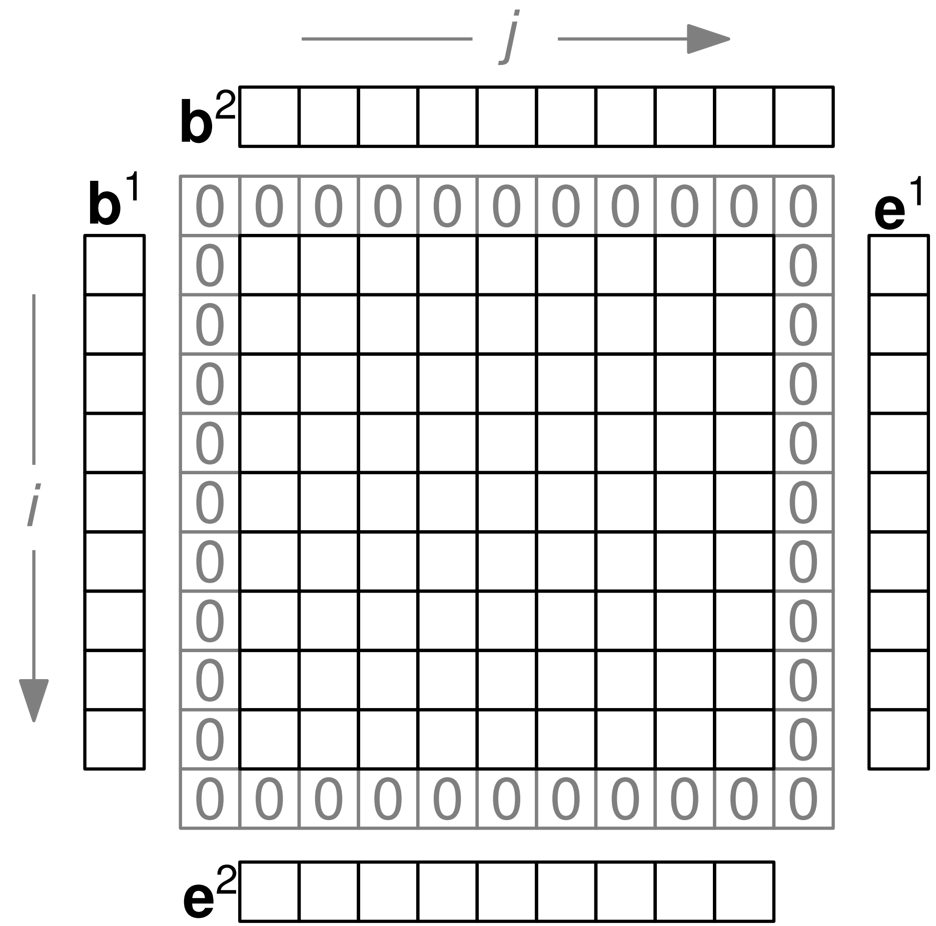
Grid of binary variables $x_{i,j}$



Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$



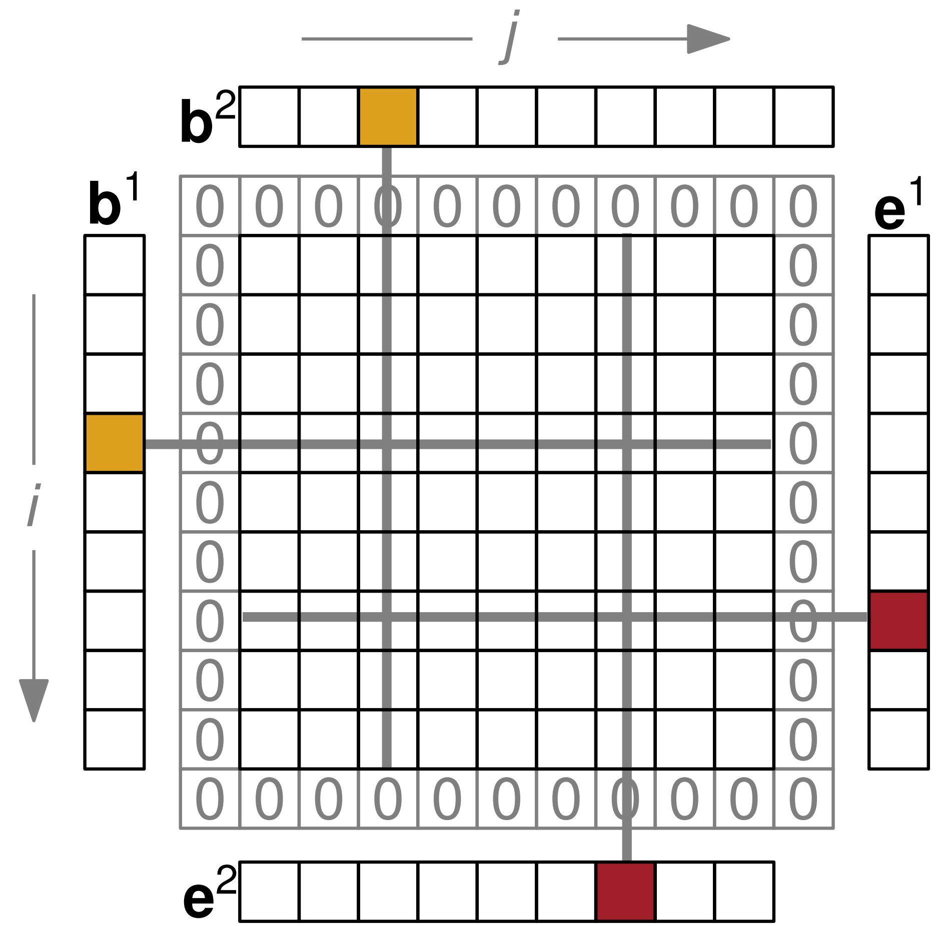
Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$

1.) There is a first/last row/column.

$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1, 2\}$



Representing Boxes

Represent box B by ILP constraints (here in 2D).

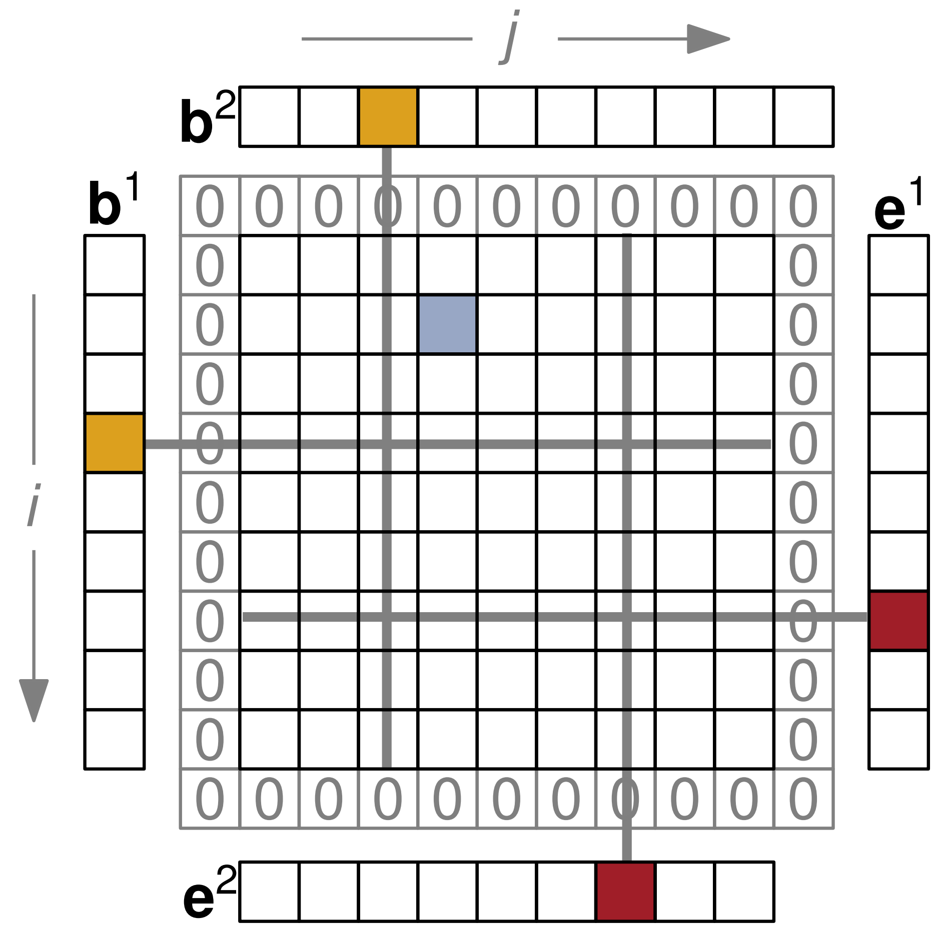
Grid of binary variables $x_{i,j}$

1.) There is a first/last row/column.

$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1, 2\}$

2.) Boxes are not empty.

$\sum_{i,j} x_{i,j} \geq 1$



Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$

1.) There is a first/last row/column.

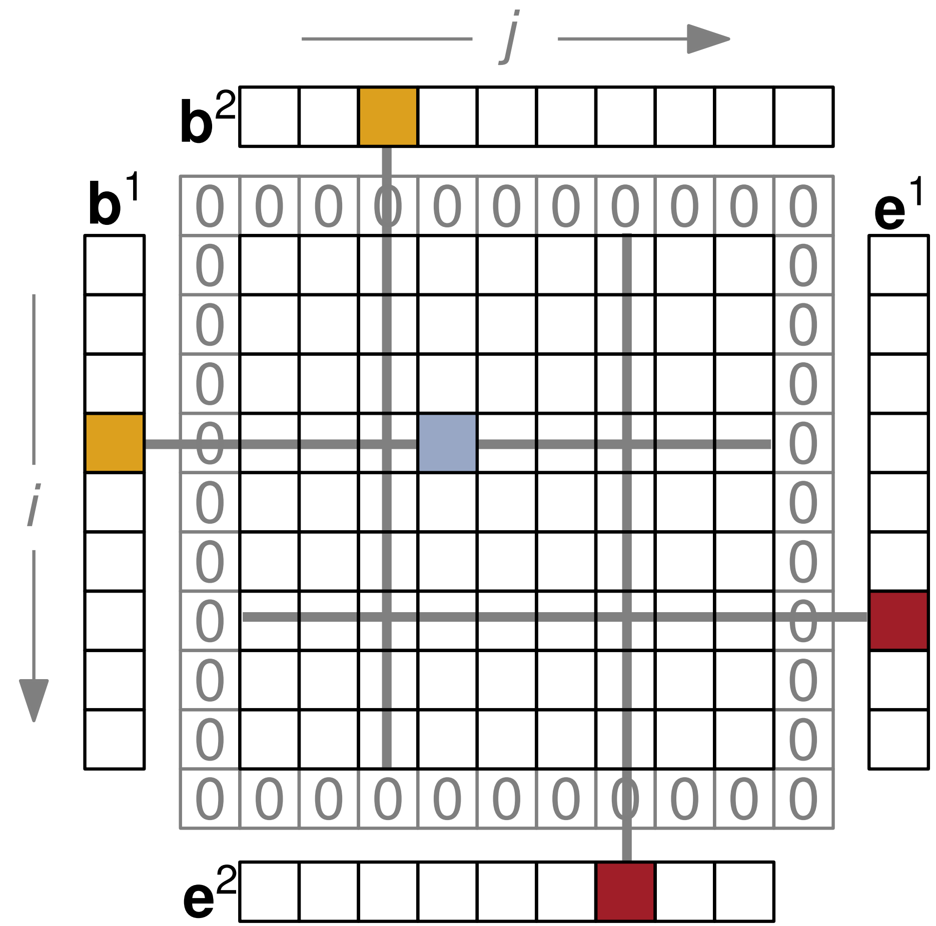
$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1, 2\}$

2.) Boxes are not empty.

$\sum_{i,j} x_{i,j} \geq 1$

3.) The first row of B is indicated by b_i^1.

$x_{i,j} \leq x_{i-1,j} + b_i^1$



Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$

1.) There is a first/last row/column.

$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1,2\}$

2.) Boxes are not empty.

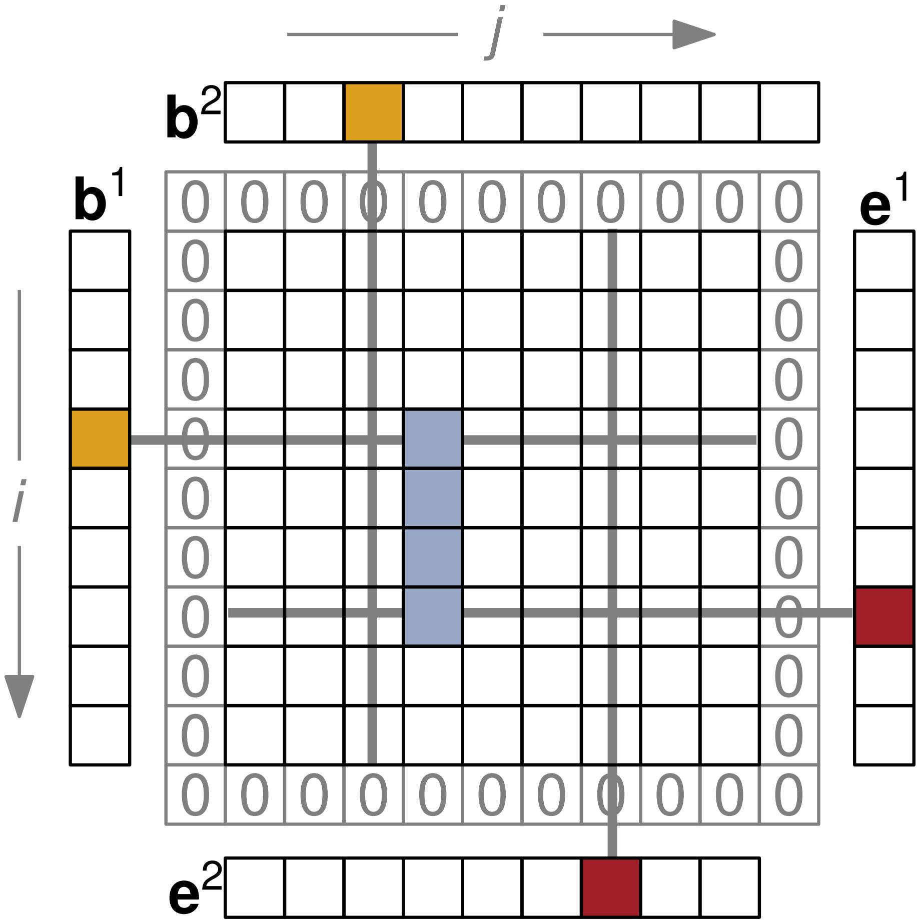
$\sum_{i,j} x_{i,j} \geq 1$

3.) The first row of B is indicated by b_i^1.

$x_{i,j} \leq x_{i-1,j} + b_i^1$

4.) The last row of B is indicated by e_i^1.

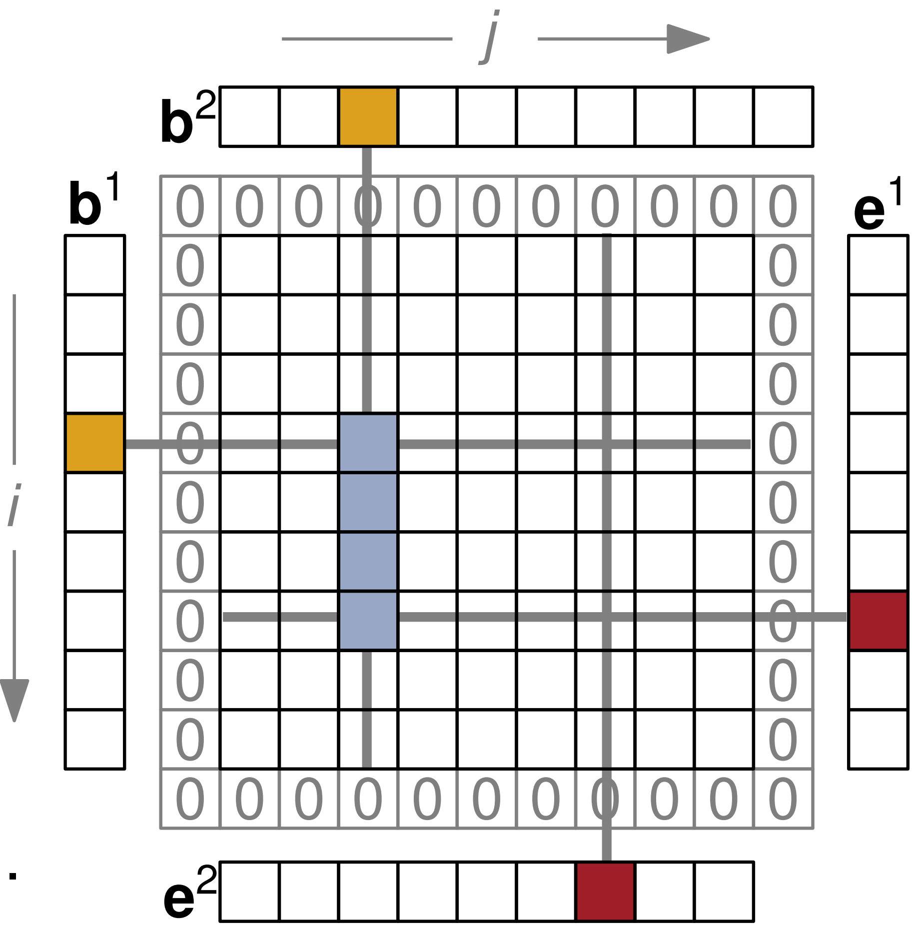
$x_{i,j} \leq x_{i+1,j} + e_i^1$



Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$



1.) There is a first/last row/column.

$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1, 2\}$

2.) Boxes are not empty.

$\sum_{i,j} x_{i,j} \geq 1$

3.) The first row of B is indicated by b_i^1.

$x_{i,j} \leq x_{i-1,j} + b_i^1$

4.) The last row of B is indicated by e_i^1.

$x_{i,j} \leq x_{i+1,j} + e_i^1$

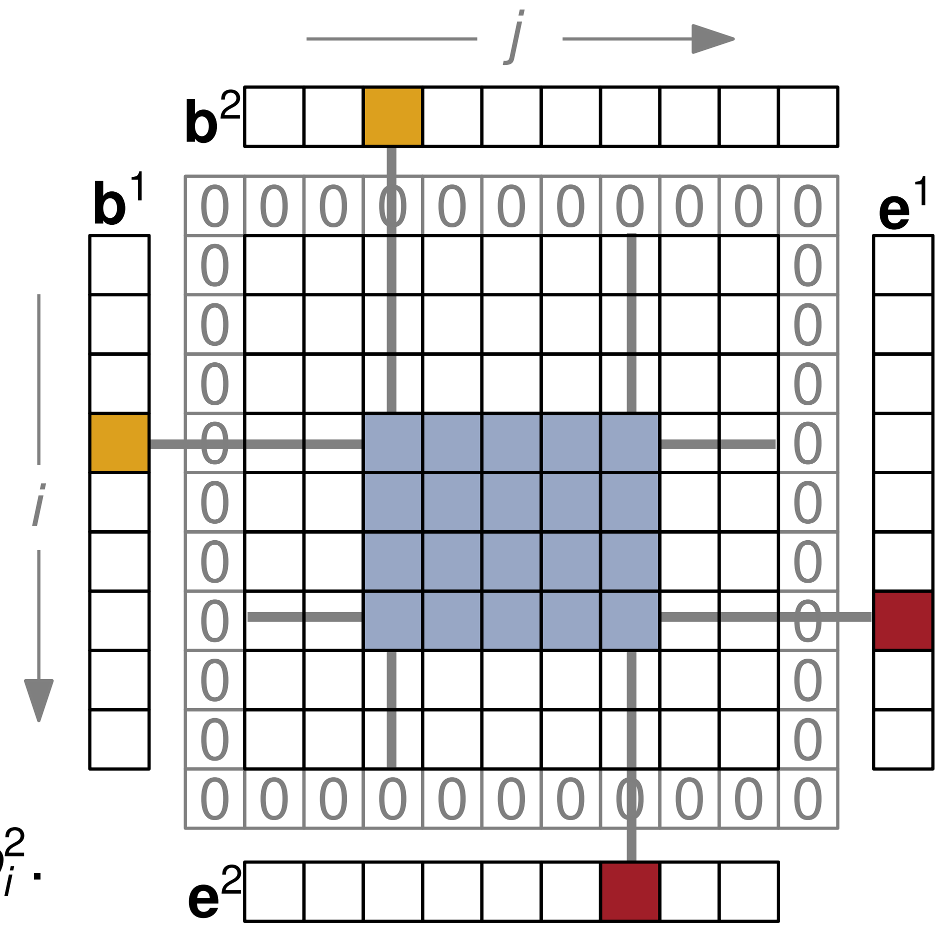
5.) The first column of B is indicated by b_i^2.

$x_{i,j} \leq x_{i,j-1} + b_j^2$

Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$



1.) There is a first/last row/column.

$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1, 2\}$

2.) Boxes are not empty.

$\sum_{i,j} x_{i,j} \geq 1$

3.) The first row of B is indicated by b_i^1.

$x_{i,j} \leq x_{i-1,j} + b_i^1$

4.) The last row of B is indicated by e_i^1.

$x_{i,j} \leq x_{i+1,j} + e_i^1$

5.) The first column of B is indicated by b_i^2.

$x_{i,j} \leq x_{i,j-1} + b_j^2$

6.) The last column of B is indicated by e_i^2.

$x_{i,j} \leq x_{i,j+1} + e_j^2$

Representing Boxes

Represent box B by ILP constraints (here in 2D).

Grid of binary variables $x_{i,j}$

1.) There is a first/last row/column.

$\sum_i b_i^d = 1 \quad \sum_i e_i^d = 1$ for $d \in \{1, 2\}$

2.) Boxes are not empty.

$\sum_{i,j} x_{i,j} \geq 1$

3.) The first row of B is indicated by b_i^1.

$x_{i,j} \leq x_{i-1,j} + b_i^1$

4.) The last row of B is indicated by e_i^1.

$x_{i,j} \leq x_{i+1,j} + e_i^1$

5.) The first column of B is indicated by b_i^2.

$x_{i,j} \leq x_{i,j-1} + b_j^2$

6.) The last column of B is indicated by e_i^2.

$x_{i,j} \leq x_{i,j+1} + e_j^2$

Aufgaben:

- modelliere Höhe(B) = 1
- modelliere Höhe(B) = k
- modelliere $B_1 \cap B_2 = \emptyset$
- modelliere $B_1 \subset B_2$

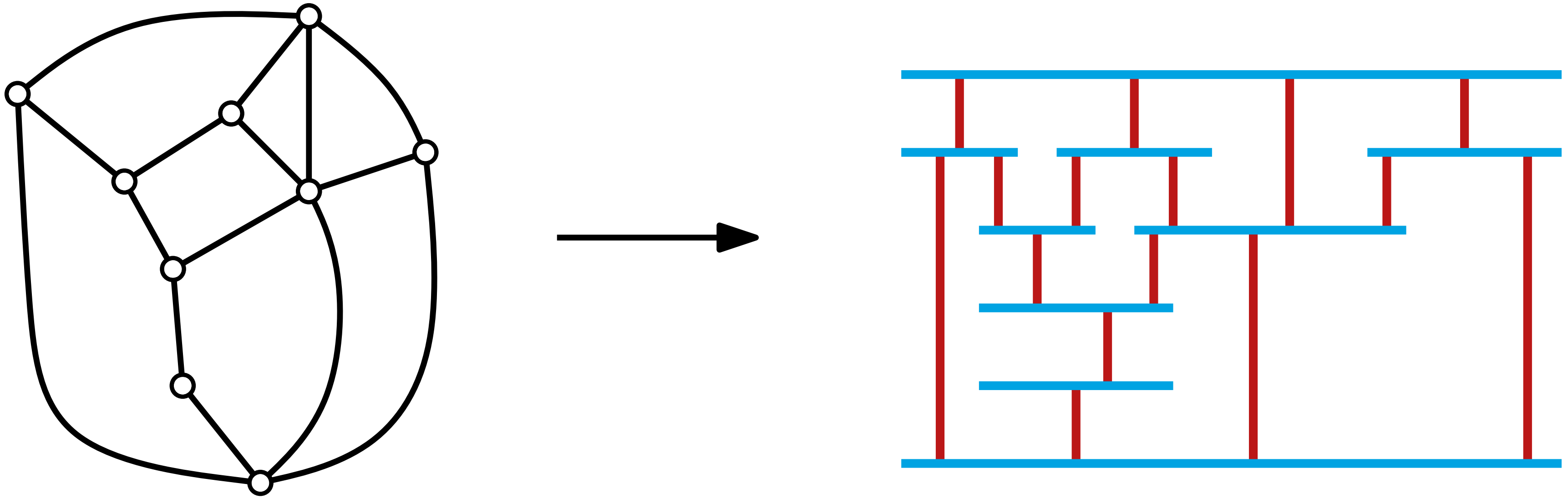
Visibility Representations

Def: A **visibility representation** of a graph $G = (V, E)$ draws

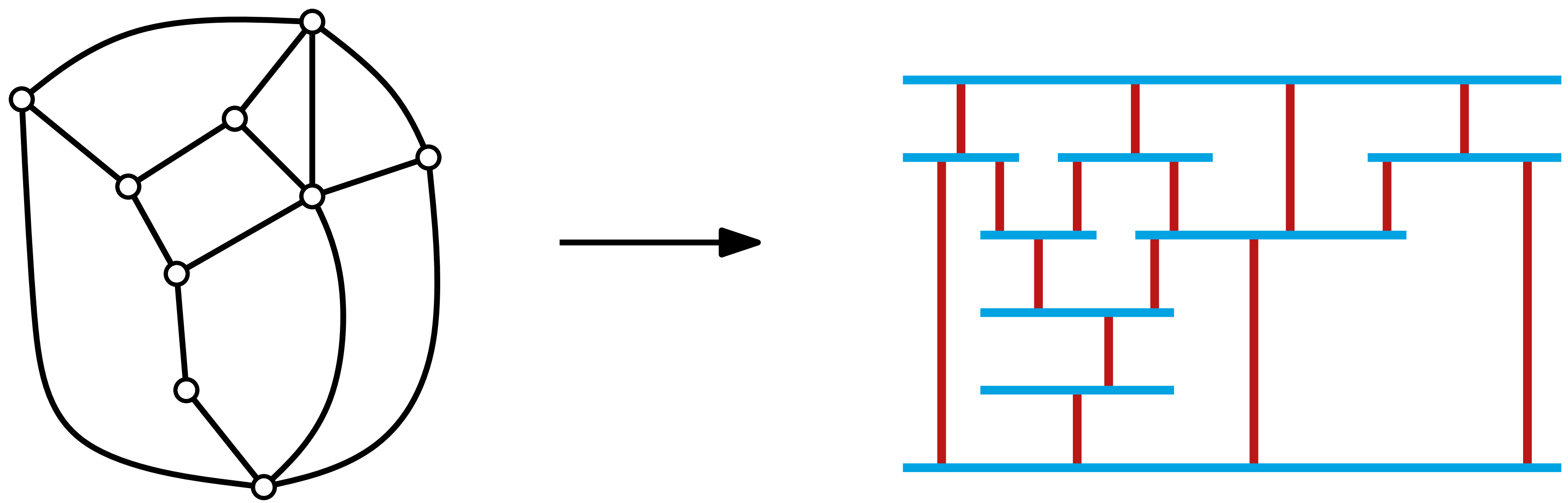
- each vertex v as a horizontal line segment $\Gamma(v)$
- each edge $e = (u, v)$ as a vertical line segment $\Gamma(e)$

such that

- no two vertex segments intersect
- no two edge segments intersect
- each edge segment $\Gamma(u, v)$ has its endpoints on $\Gamma(u)$ and $\Gamma(v)$ and intersects no other vertex segment



Properties of Visibility Representations



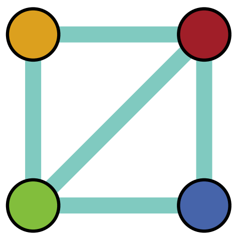
- graph G must be planar (obviously)
- every planar graph has a visibility representation

  [Wismath '85], [Tamassia, Tollis '86]

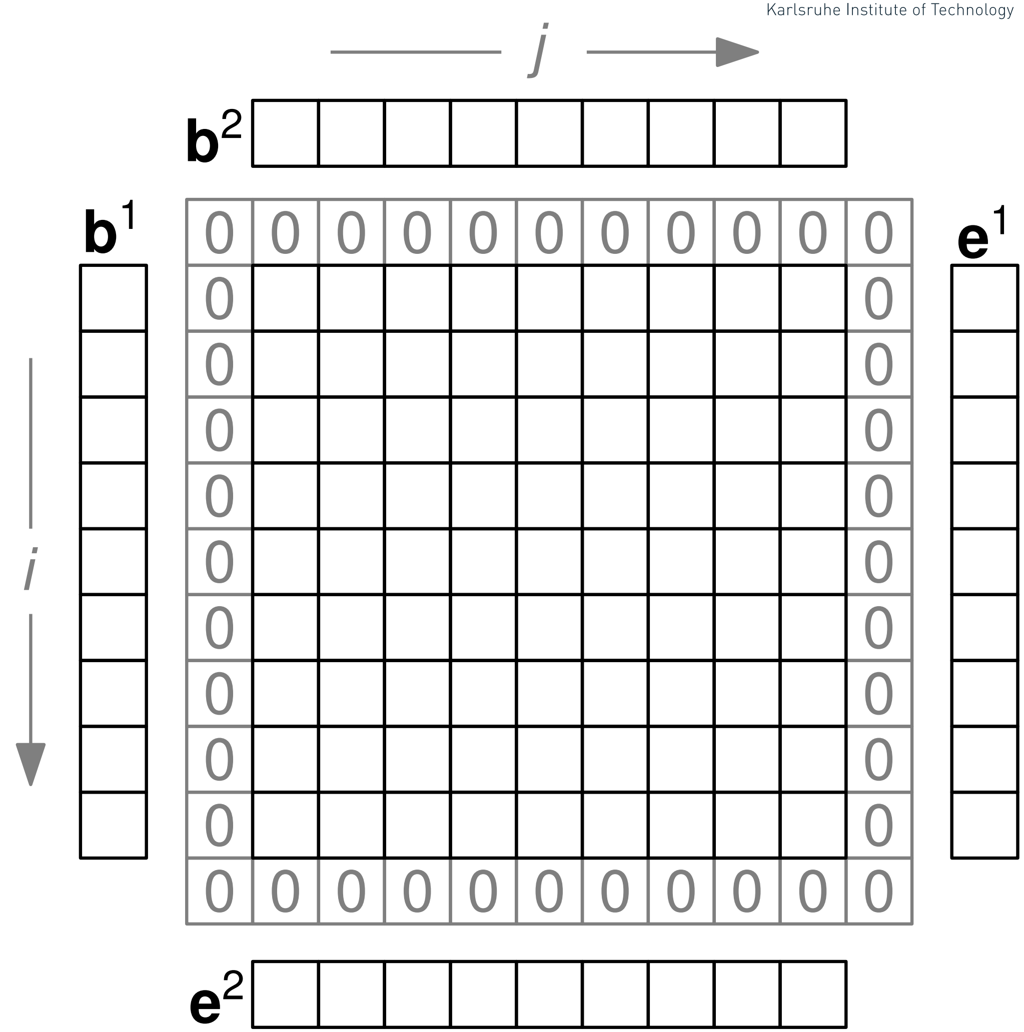
- minimizing the area of a visibility representation is NP-hard

  [Lin, Eades '03]

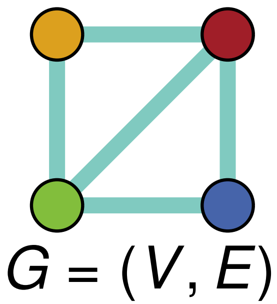
Modeling Visibility Representation



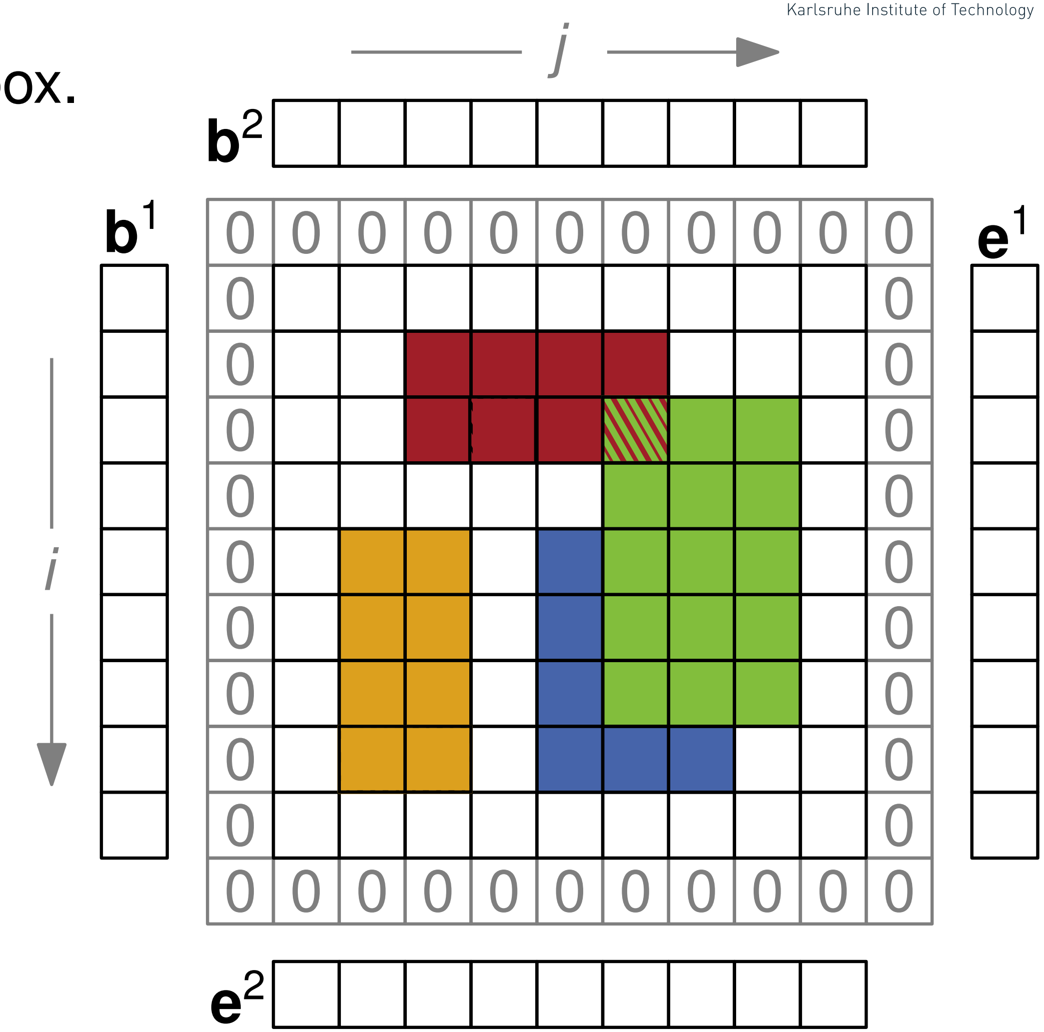
$G = (V, E)$



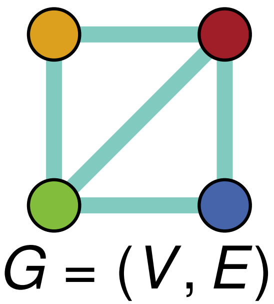
Modeling Visibility Representation



1.) $\forall v \in V$ introduce box.

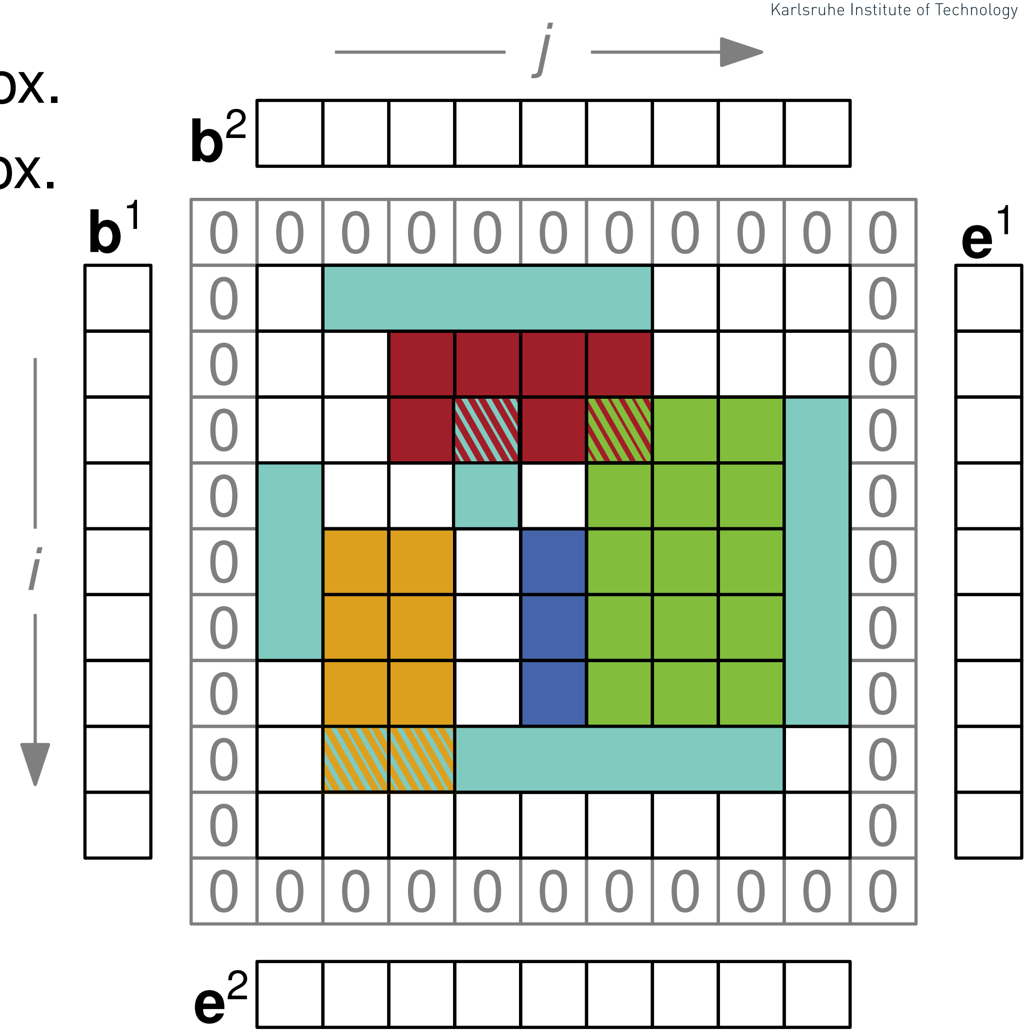


Modeling Visibility Representation

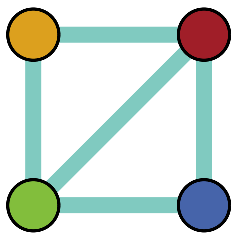


1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.



Modeling Visibility Representation

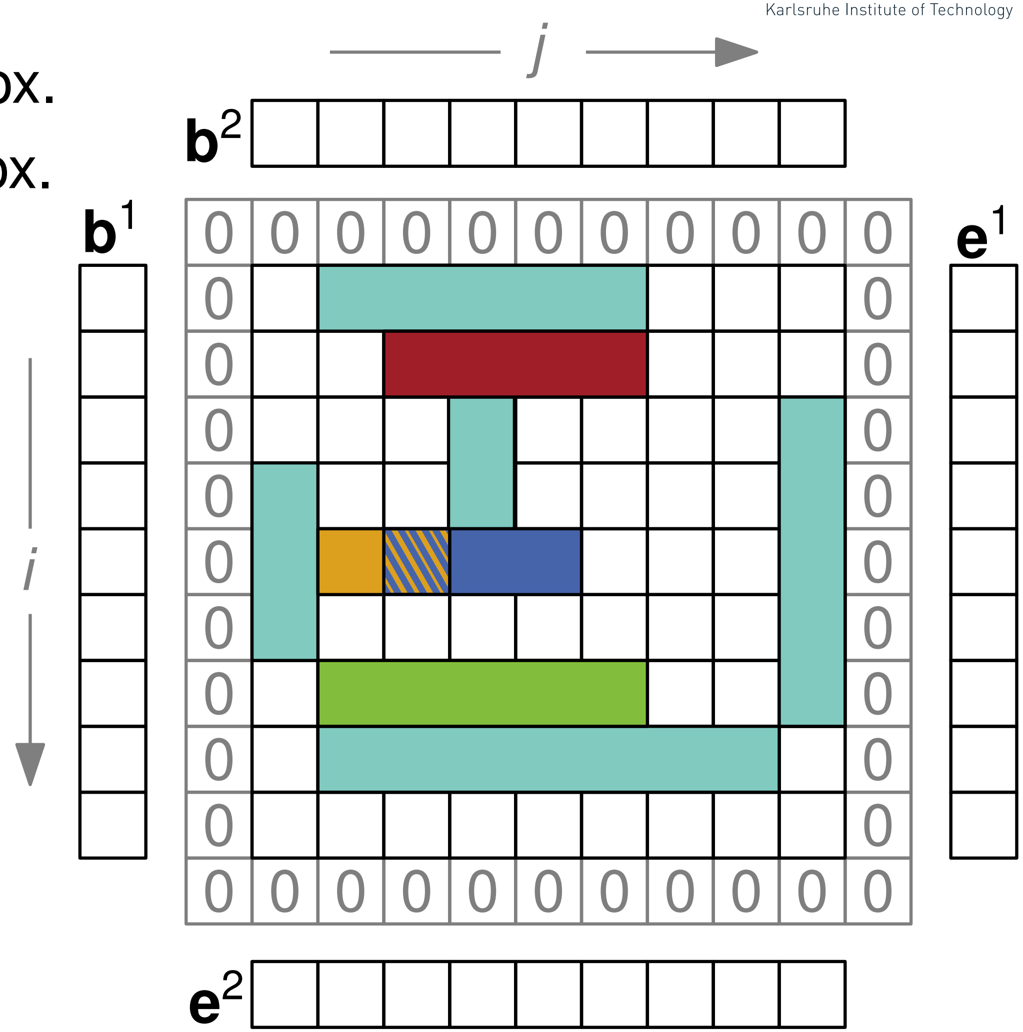


$G = (V, E)$

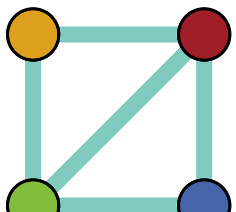
1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.

3.) Vertices are horizontal segments:



Modeling Visibility Representation



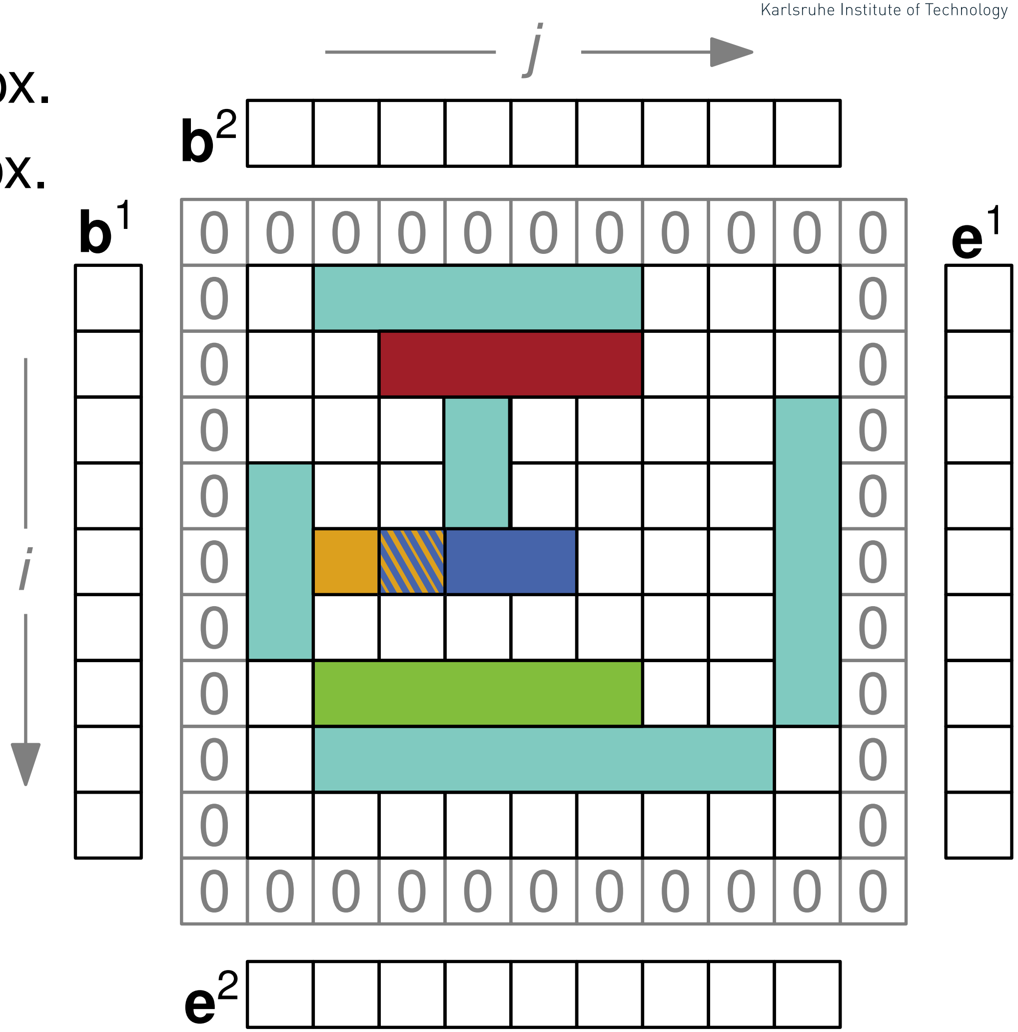
$G = (V, E)$

1.) $\forall v \in V$ introduce box.

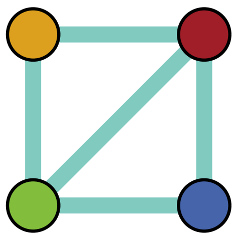
2.) $\forall e \in E$ introduce box.

3.) Vertices are horizontal segments:

$$b_i^1(v) = e_i^1(v)$$



Modeling Visibility Representation



$G = (V, E)$

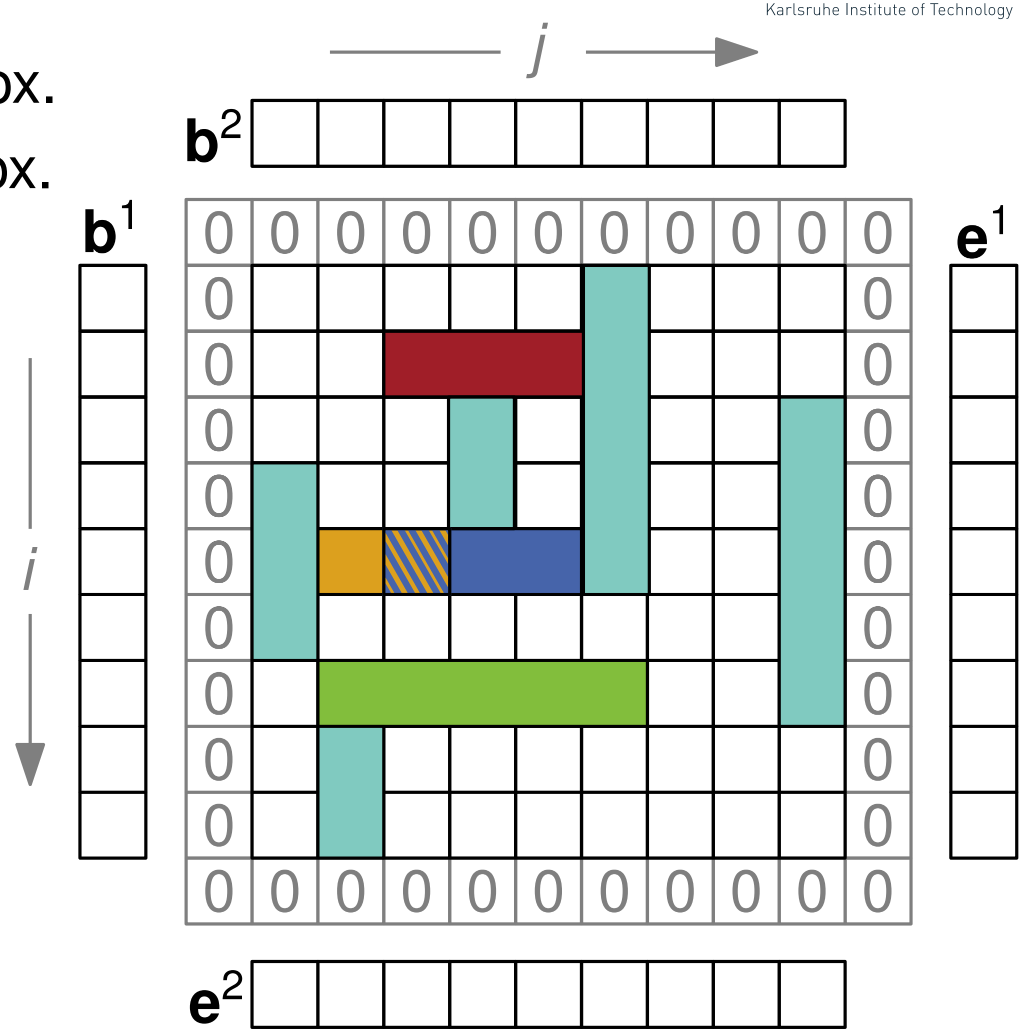
1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.

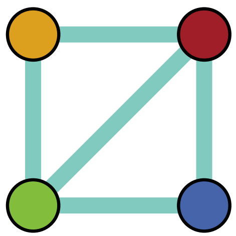
3.) Vertices are horizontal segments:

$$b_i^1(v) = e_i^1(v)$$

4.) Edges are vertical segments:



Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

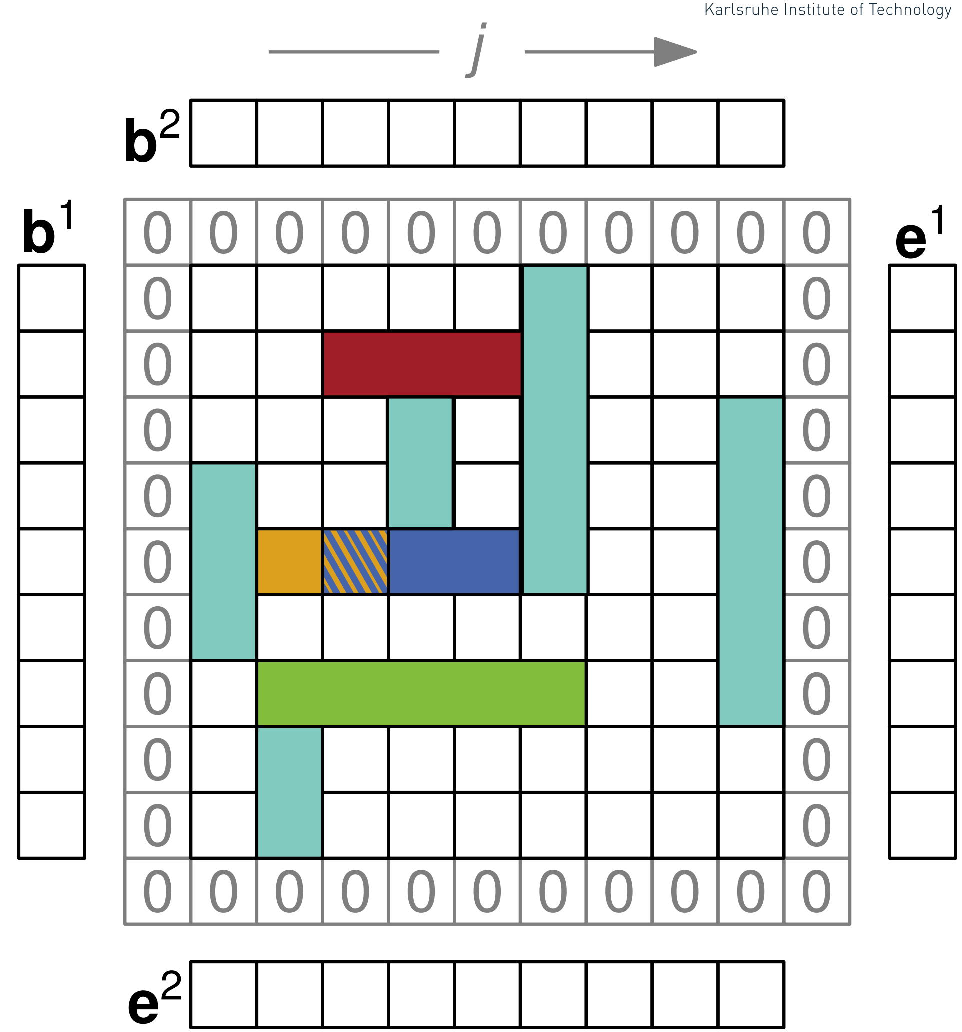
2.) $\forall e \in E$ introduce box.

3.) Vertices are horizontal segments:

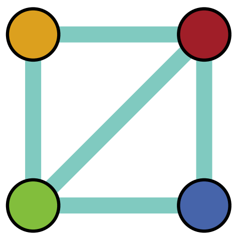
$$b_i^1(v) = e_i^1(v)$$

4.) Edges are vertical segments:

$$b_i^2(e) = e_i^2(e)$$



Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.

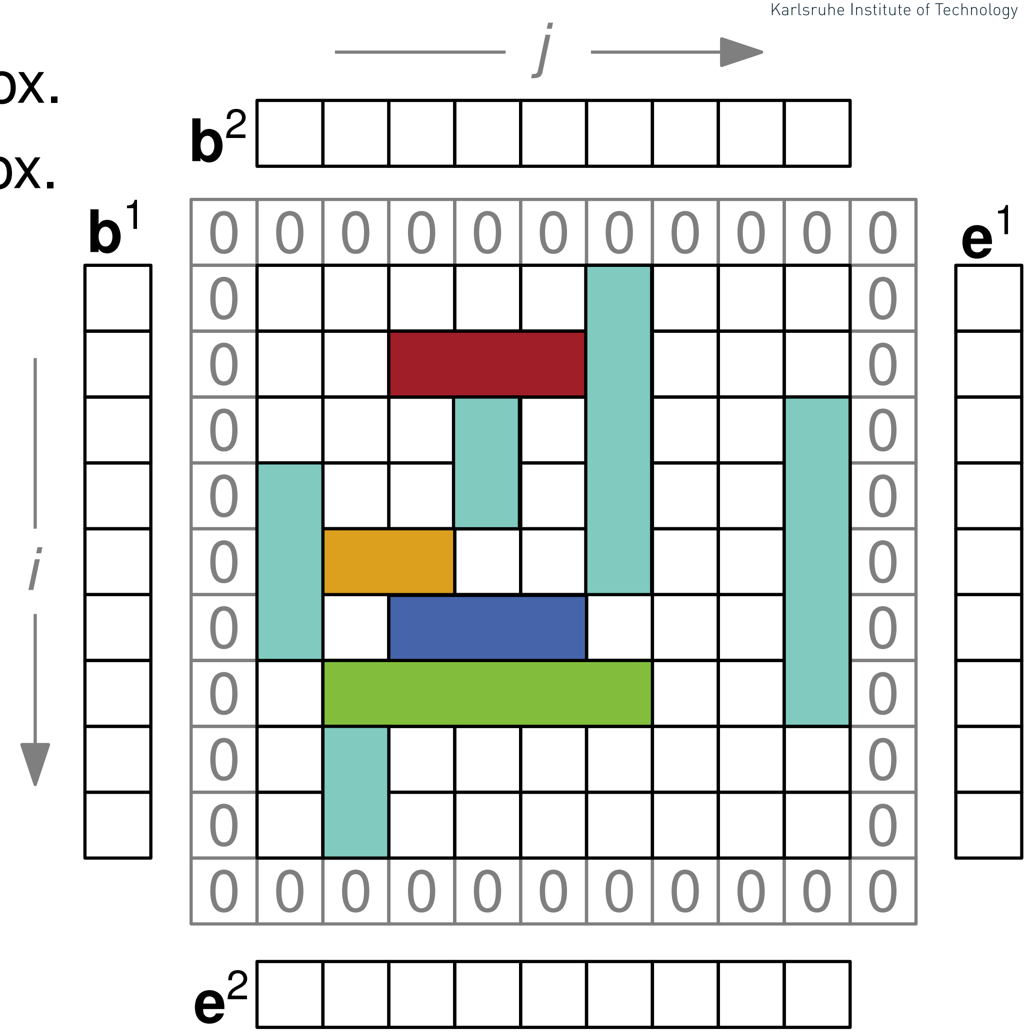
3.) Vertices are horizontal segments:

$$b^1_i(v) = e^1_i(v)$$

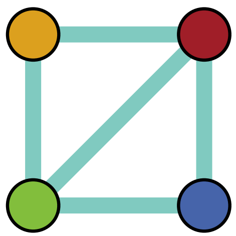
4.) Edges are vertical segments:

$$b^2_i(e) = e^2_i(e)$$

5.) Vertices do not overlap:



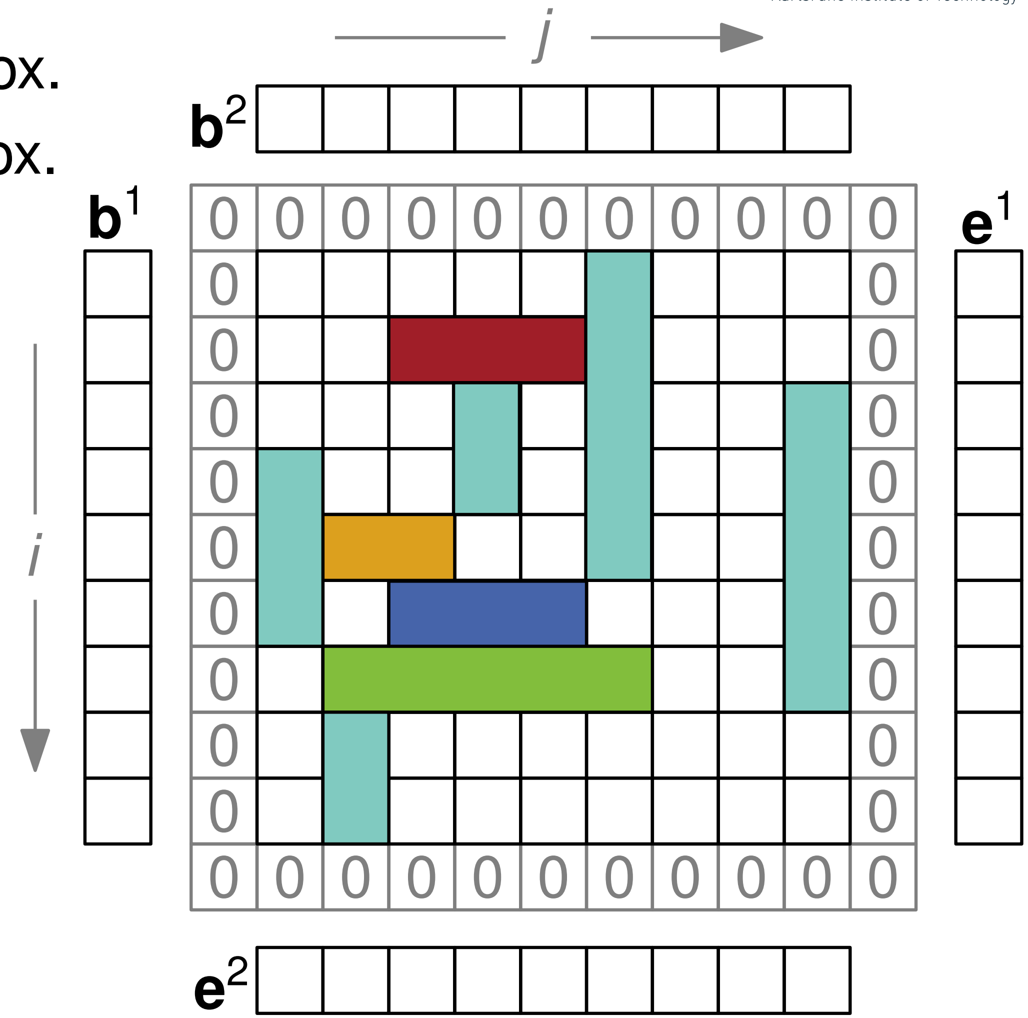
Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.



3.) Vertices are horizontal segments:

$$b_i^1(v) = e_i^1(v)$$

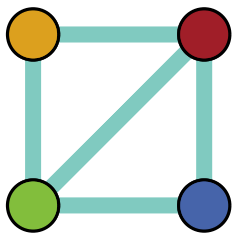
4.) Edges are vertical segments:

$$b_i^2(e) = e_i^2(e)$$

5.) Vertices do not overlap:

$$\sum_{v \in V} x_{i,j}(v) \leq 1$$

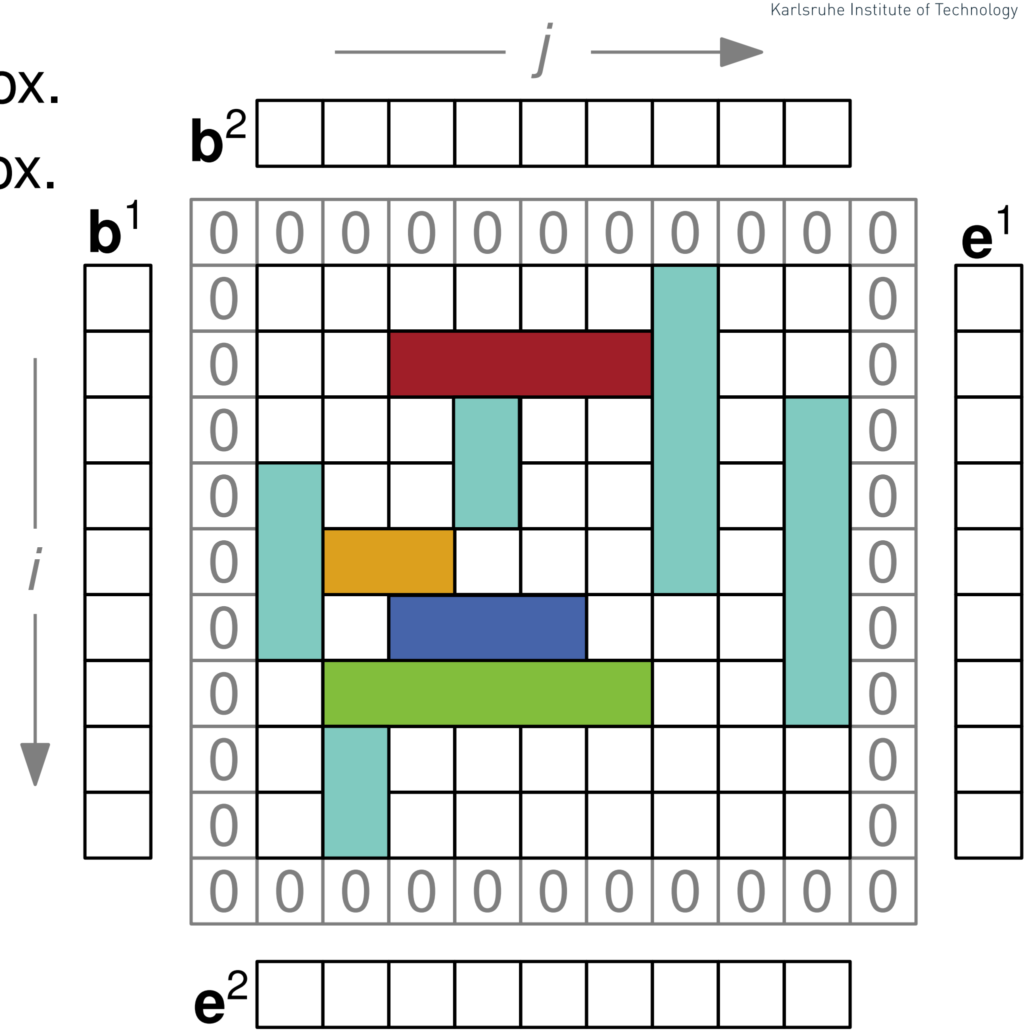
Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.



3.) Vertices are horizontal segments:

$$b_i^1(v) = e_i^1(v)$$

4.) Edges are vertical segments:

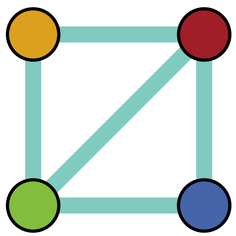
$$b_i^2(e) = e_i^2(e)$$

5.) Vertices do not overlap:

$$\sum_{v \in V} x_{i,j}(v) \leq 1$$

6.) Edges do not overlap non-incident vertices:

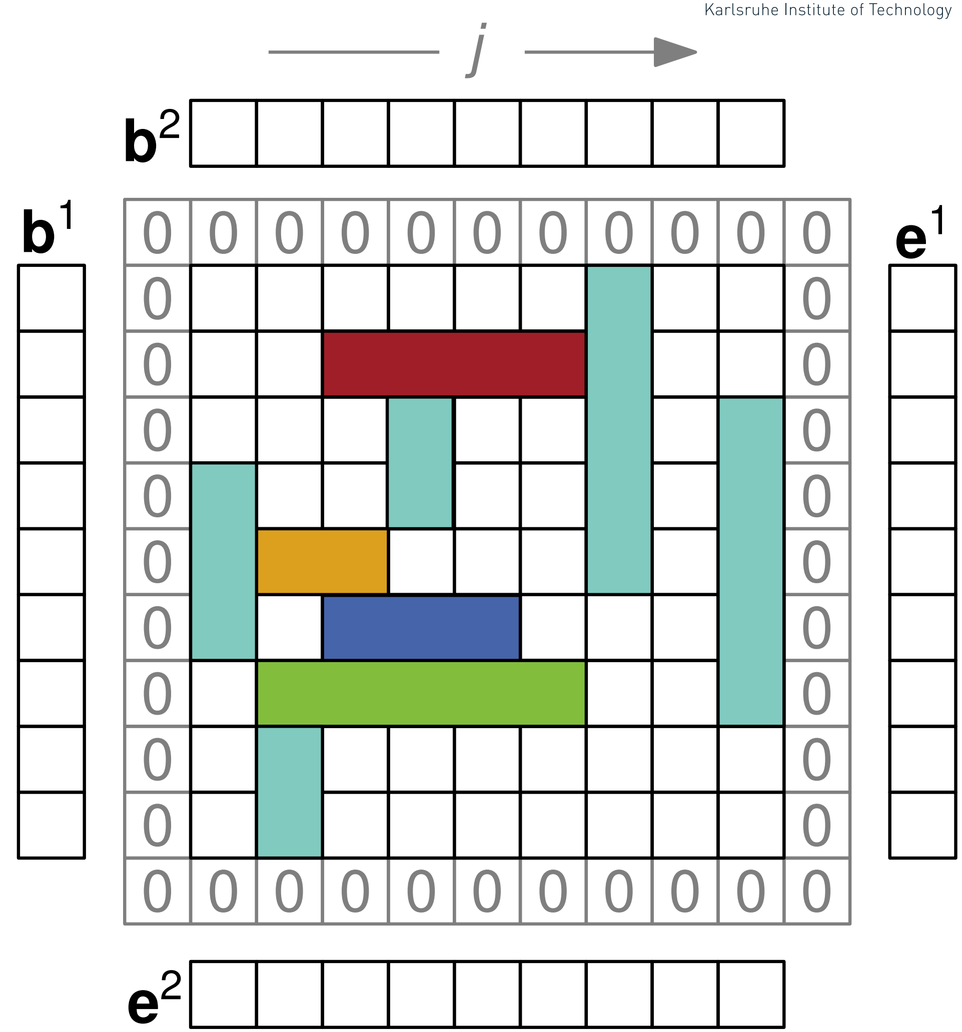
Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.



3.) Vertices are horizontal segments:

$$b_i^1(v) = e_i^1(v)$$

4.) Edges are vertical segments:

$$b_i^2(e) = e_i^2(e)$$

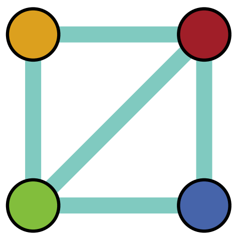
5.) Vertices do not overlap:

$$\sum\nolimits_{v \in V} x_{i,j}(v) \leq 1$$

6.) Edges do not overlap non-incident vertices:

$$\sum\nolimits_{v \in V \setminus e} x_{i,j}(v) \leq (1 - x_{i,j}(e)) \text{ for all } e \in E$$

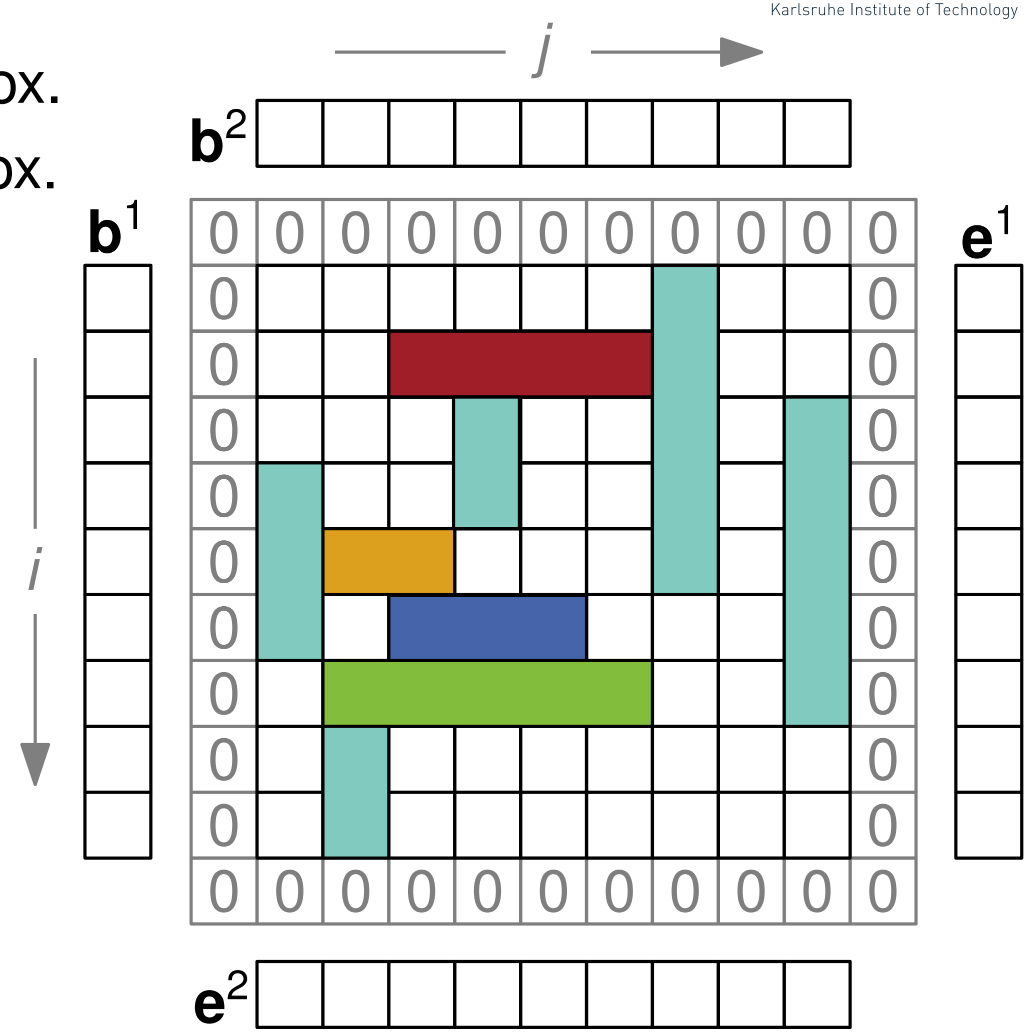
Modeling Visibility Representation



$G = (V, E)$

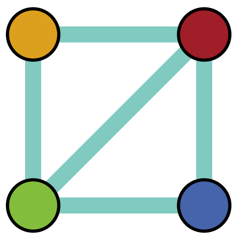
1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.



7.) Edges intersect incident vertices:

Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

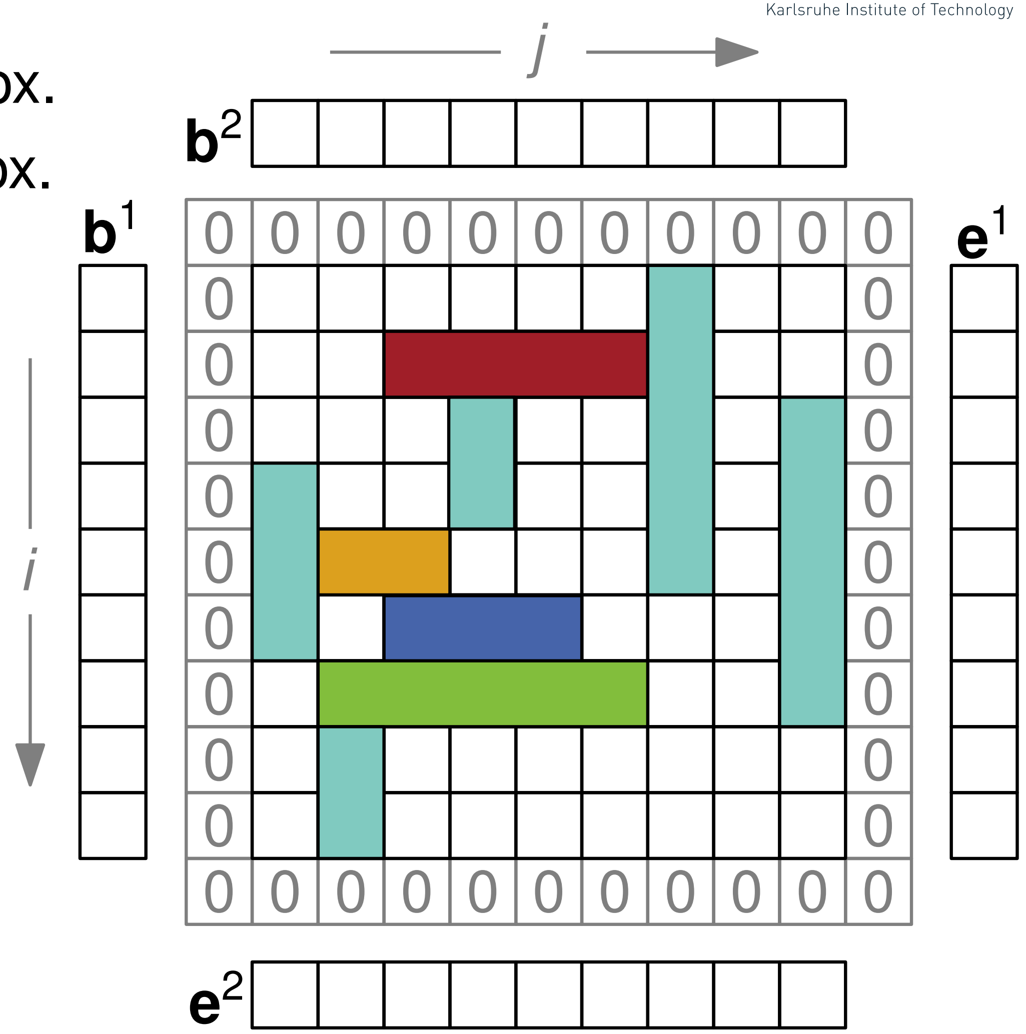
2.) $\forall e \in E$ introduce box.

7.) Edges intersect incident vertices:

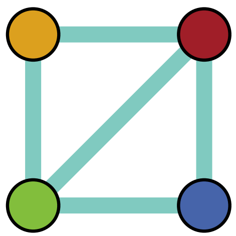
For all $e = \{u, v\} \in E$:

a.) Grid of binary variables

$x_{i,j}(e, u)$ $x_{i,j}(e, v)$



Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.

7.) Edges intersect incident vertices:

For all $e = \{u, v\} \in E$:

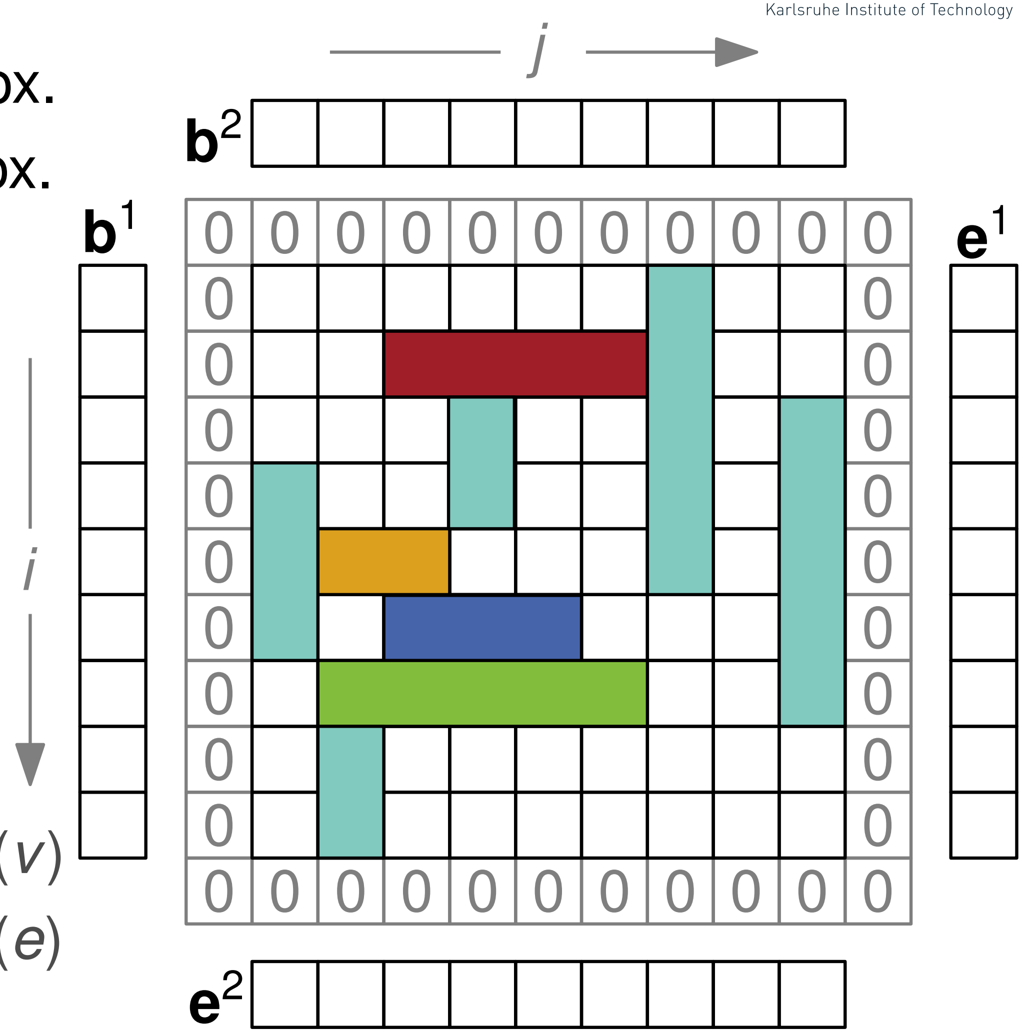
a.) Grid of binary variables

$x_{i,j}(e, u) \quad x_{i,j}(e, v)$

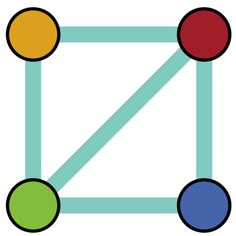
b.) Common point

$x_{i,j}(e, u) \leq x_{i,j}(u) \quad x_{i,j}(e, v) \leq x_{i,j}(v)$

$x_{i,j}(e, u) \leq x_{i,j}(e) \quad x_{i,j}(e, v) \leq x_{i,j}(e)$



Modeling Visibility Representation



$G = (V, E)$

1.) $\forall v \in V$ introduce box.

2.) $\forall e \in E$ introduce box.

7.) Edges intersect incident vertices:

For all $e = \{u, v\} \in E$:

a.) Grid of binary variables

$x_{i,j}(e, u)$ $\quad x_{i,j}(e, v)$

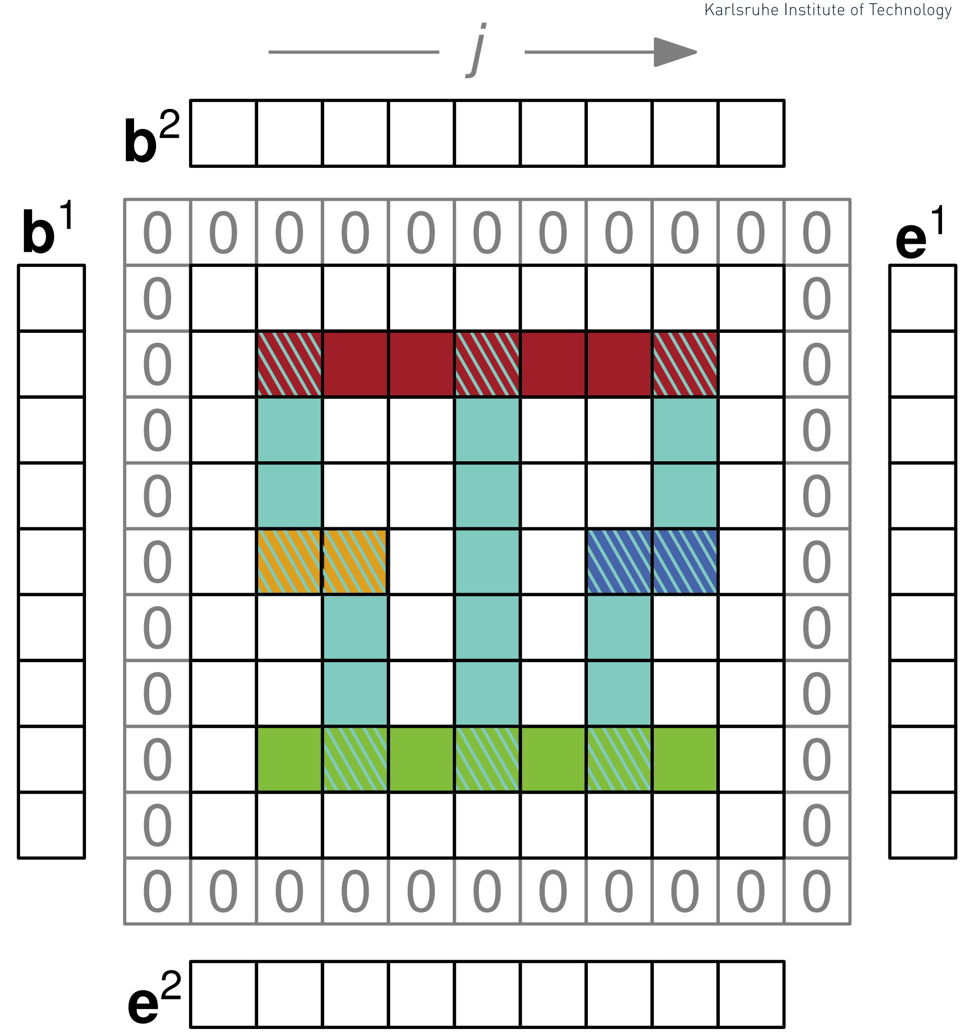
b.) Common point

$x_{i,j}(e, u) \leq x_{i,j}(u)$ $\quad x_{i,j}(e, v) \leq x_{i,j}(v)$

$x_{i,j}(e, u) \leq x_{i,j}(e)$ $\quad x_{i,j}(e, v) \leq x_{i,j}(e)$

c.) Existence of common point.

$\sum_{i,j} x_{i,j}(e, u) \geq 1$ $\quad \sum_{i,j} x_{i,j}(e, v) \geq 1$



Optimum *st*-orientation

Recall: An *st***-orientation** of an undirected graph $G = (V, E)$ and two vertices $s, t \in V$ with $(s, t) \in E$ is an orientation of the edges in E s.t.

- s is the unique source
- t is the unique sink.

Optimum st-orientation

Recall: An st**-orientation** of an undirected graph $G = (V, E)$ and two vertices $s, t \in V$ with $(s, t) \in E$ is an orientation of the edges in E s.t.

- s is the unique source
- t is the unique sink.

Properties:

- works only for biconnected graphs
- can be computed in linear time
- ingredient for orthogonal layout algorithm (see lecture 14)
- NP-complete to find st-orientation that minimizes the length of the longest st-path, even if G is planar [Sadasivam, Zhang '10]

Optimum *st*-orientation

Recall: An *st*-**orientation** of an undirected graph $G = (V, E)$ and two vertices $s, t \in V$ with $(s, t) \in E$ is an orientation of the edges in E s.t.

- s is the unique source
- t is the unique sink.

Properties:

- works only for biconnected graphs
- can be computed in linear time
- ingredient for orthogonal layout algorithm (see lecture 14)
- NP-complete to find *st*-orientation that minimizes the length of the longest *st*-path, even if G is planar [Sadasivam, Zhang '10]

Find minimum-length *st*-orientation using 1D grid-based model.

Summary

- flexible framework for modeling many grid-based graph layout problems
- suitable for NP-hard problems
- less efficient than tailored algorithms
- gets faster if ILP constraints are transformed into SAT

Summary

- flexible framework for modeling many grid-based graph layout problems
- suitable for NP-hard problems
- less efficient than tailored algorithms
- gets faster if ILP constraints are transformed into SAT

But is it easy to use in practice?

PIGRA Demo (Fabian Klute)