Algorithms for Graph Visualization

Introduction

Tamara Mchedlidze, Benjamin Niedermann
17.10.2016
Organizational

Lectures

- Tamara Mchedlidze
- mched@iti.uka.de
- Office 307
- Office hours: request by email

Exercises

- Benjamin Niedermann
- benjamin.niedermann@kit.edu
- Office 321
- Office hours: request by email
Organizational

Lectures
- Tamara Mchedlidze
- mched@iti.uka.de
- Office 307
- Office hours: request by email

Exercises
- Benjamin Niedermann
- benjamin.niedermann@kit.edu
- Office 321
- Office hours: request by email

Meetings
- Monday 09:45 – 11:15 Uhr, Room 301
- Wednesday 9:45 – 11:15 Uhr, Room -119
- exact plan on the web-page
Organizational Webseite

i11www.iti.kit.edu/teaching/winter2016/graphvis/

- Latest news
- Lecture slides
- Exercise sheets
- Literature & Additional material
- Lecture notes (skript)
Organizational

Webseite

i11www.iti.kit.edu/teaching/winter2016/graphvis/

- Latest news
- Lecture slides
- Exercise sheets
- Literature & Additional material
- Lecture notes (skript)

Graph Visualization in Master’s Studies

Bachelor

- Algorithms 1 & 2
- Basic Theory of Inf.
- Algorithms for Planar Graphs

Master

- Algorithms for Graph Visualization

---

Dr. Tamara Mchedlidze - Algorithms for Graph Visualization
Algorithms for Graph Visualization

**Learning goals:** At the end of the semester you:

- Know terms and problem definitions
- Know the introduced algorithms, understand how they work, can analyse them
- Can select appropriate algorithms and data structures
- Can analyse a new graph drawing problem and construct an efficient algorithm/prove hardness
Algorithms for Graph Visualization

**Learning goals:** At the end of the semester you:

- Know terms and problem definitions
- Know the introduced algorithms, understand how they work, can analyse them
- Can select appropriate algorithms and data structures
- Can analyse a new graph drawing problem and construct an efficient algorithm/prove hardness

**Prerequisites:** Algorithms 1 & 2, Theoretical Basics of inf.

**Helpful:** Algorithms for Planar Graphs
Algorithms for Graph Visualization

**Learning goals:** At the end of the semester you:

- Know terms and problem definitions
- Know the introduced algorithms, understand how they work, can analyse them
- Can select appropriate algorithms and data structures
- Can analyse a new graph drawing problem and construct an efficient algorithm/prove hardness

**Prerequisites:** Algorithms 1 & 2, Theoretical Basics of inf.

**Helpful:** Algorithms for Planar Graphs

**Suggested time requirements:**

- Attending Lecture and Exercises: ca. 35h
- Preparation/post-processing: ca. 35h
- Work on the exercises: ca. 40h
- Preparation for the exam: ca. 40h

5LP = 150h
Examination

Master Informatics

- Algorithms for Graph Visualization (IN4INALGVG)
- **new Module:** Graph visualization+ (more about this later)
Examination

Master Informatics

- Algorithms for Graph Visualization (IN4INALGVG)
- **new Module:** Graph visualization+ (more about this later)

**Examination procedure:**

- In order to take an exam you need to participate actively in the exercise sessions (e.g. present your own solutions on the board)
- oral exam(app. 20 Minutes)
Graph Visualization

**Background:** International Symposium on Graph Drawing (GD) and Graph Drawing Challenge
**Graph Visualization**

**Background:** International Symposium on Graph Drawing (GD) and Graph Drawing Challenge

**Task:** develop software for a given graph visualization problem
Graph Visualization+

**Background:** International Symposium on Graph Drawing (GD) and Graph Drawing Challenge

**Task:** develop software for a given graph visualization problem

**Winner 2014:** KIT-Team from the lecture WS 2013/14!
Graph Visualization+

**Background:** International Symposium on Graph Drawing (GD) and Graph Drawing Challenge

**Task:** develop software for a given graph visualization problem

**Winner 2014:** KIT-Team from the lecture WS 2013/14!

**Second place 2015:** KIT-Team from the lecture SS 2015!

**Best poster award 2016:** One of the students who continued working on the 2015’s topic!
Graph Visualization

**Background:** International Symposium on Graph Drawing (GD) and Graph Drawing Challenge

**Task:** develop software for a given graph visualization problem

2013/14:

- Software development was part of the lecture
- Lots of fun but time-consuming

2015:

- as a practical course SS 2015
- combined modul with 10 LP
- time-wise worked very well

**evaluation:** excellent practical course award
Structure of the Lecture

Media:

- **Slides** & Blackboard
- Exercise sheets are provided a week before the exercise session
- (incomplete) Lecture notes/Books
- Original literature (papers)
Structure of the Lecture

Media:
- Slides & Blackboard
- Exercise sheets are provided a week before the exercise session
- (incomplete) Lecture notes/Books
- Original literature (papers)

Content:
- We look at the **algorithmic core** of visualization problems
- **not:** graphical Rendering
- Modelling, Algorithms, formal Analysis
  - Divide & Conquer / Recursion
  - combinatorial optimization (ILP)
  - incremental algorithms
  - algorithms for special graph classes
  - drawing techniques using physical analogies
Literature (available in the library)

G. di Battista, P. Eades, R. Tamassia, I. Tollis:  
Graph Drawing  
Prentice Hall, 1998

M. Kaufmann, D. Wagner:  
Drawing Graphs: Methods and Models  
Springer, 2001

T. Nishizeki, Md. S. Rahman:  
Planar Graph Drawing  
World Scientific, 2004

R. Tamassia:  
Handbook of Graph Drawing and Visualization  
CRC Press, 2013  
http://cs.brown.edu/~rt/gdhandbook/
Usefull Knowledge

Basic knowledge of Graph Theory:

- Graph, Nodes/Vertices, Edges
- Node degree, Neighbourhood, Adjacent, Incident
- Connectivity, Tree, Cycle, Path
- ...
Usefull Knowledge

**Basic knowledge of Graph Theory:**
- Graph, Nodes/Vertices, Edges
- Node degree, Neighbourhood, Adjacent, Incident
- Connectivity, Tree, Cycle, Path
- ...

**Basic knowledge in Algorithms:**
- Asymptotic running time, $O$-notation
- Complexity, NP-Hardness
- Network flow
- Linear Programming
- Recursion
- Divide & Conquer
- Approximation
- ...

Introduction to Graph Visualization
Graph and its Representation

What is a Graph?
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n$
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$

Set of nodes $V = \{v_1, \ldots, v_n\}$

Set of edges $E = \{e_1, \ldots, e_m\}$,

$e_i = \{v_j, v_k\}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n$

Representations?
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$

Set of nodes $V = \{v_1, \ldots, v_n\}$

Set of edges $E = \{e_1, \ldots, e_m\}$,

$e_i = \{v_j, v_k\}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n$

Representations?

Set representation:

\[
V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}
\]
\[
E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\},
\{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\},
\{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\},
\{v_9, v_{10}\}\}
\]
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}$, $1 \leq i \leq m$, $1 \leq j, k \leq n$

Representations?

Set representation

Adjacency list

\begin{align*}
  v_1 & : v_2, v_8 \\
  v_2 & : v_1, v_3 \\
  v_3 & : v_2, v_5, v_9, v_{10} \\
  v_4 & : v_5, v_6, v_9 \\
  v_5 & : v_3, v_4, v_8 \\
  v_6 & : v_4, v_8, v_9 \\
  v_7 & : v_8, v_9 \\
  v_8 & : v_1, v_5, v_6, v_7, v_9, v_{10} \\
  v_9 & : v_3, v_4, v_6, v_7, v_8, v_{10} \\
  v_{10} & : v_3, v_8, v_9
\end{align*}
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}$, $1 \leq i \leq m$, $1 \leq j, k \leq n$

Representations?
Set representation
Adjacency list
Adjacency matrix

\[
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
\end{pmatrix}
\]
Graph and its Representation

What is a Graph?

TUPLE \( G = (V, E) \)
Set of nodes \( V = \{ v_1, \ldots, v_n \} \)
Set of edges \( E = \{ e_1, \ldots, e_m \} \),
\( e_i = \{ v_j, v_k \}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n \)

Representations?

Set representation
Adjacency list
Adjacency matrix
Drawing
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
\[ e_i = \{v_j, v_k\}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n \]

Representations?

Set representation
Adjacency list
Adjacency matrix
Drawing
Graph and its Representation

\[ V = \{ v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10} \} \]

\[ E = \{ \{ v_1, v_2 \}, \{ v_1, v_8 \}, \{ v_2, v_3 \}, \{ v_3, v_5 \}, \{ v_3, v_9 \}, \{ v_3, v_{10} \}, \{ v_4, v_5 \}, \{ v_4, v_6 \}, \{ v_4, v_9 \}, \{ v_5, v_8 \}, \{ v_6, v_8 \}, \{ v_6, v_9 \}, \{ v_7, v_8 \}, \{ v_7, v_9 \}, \{ v_8, v_{10} \}, \{ v_9, v_{10} \} \} \]

\[
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
\end{pmatrix}
\]

\[ v_1 : v_2, v_8 \]
\[ v_2 : v_1, v_3 \]
\[ v_3 : v_2, v_5, v_9, v_{10} \]
\[ v_4 : v_5, v_6, v_9 \]
\[ v_5 : v_3, v_4, v_9 \]
\[ v_6 : v_4, v_8, v_9 \]
\[ v_7 : v_8, v_9 \]
\[ v_8 : v_1, v_5, v_6, v_7, v_9, v_{10} \]
\[ v_9 : v_3, v_4, v_6, v_7, v_8, v_{10} \]
\[ v_{10} : v_3, v_8, v_9 \]
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social netowrks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

![Graph Visualization Image]
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...)

![Graph Visualization Diagram](image-url)
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

![Graph visualization example](image.png)
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

---

Dr. Tamara Mchedlidze - Algorithms for Graph Visualization
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...)
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...).
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...).
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...).
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...).
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...)
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...)

Roles in Models:
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolic networks, VLSI-network, UML-diagrams, citation networks, ...).
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

[Images of various graphs and diagrams]
Why to draw graphs?

Graphs are mathematical models of real physical and abstract networks (social networks, metabolical networks, VLSI-network, UML-diagrams, citation networks, ...)

- **People think visually** – without a good visualization, complex graphs are not understandable to us.
- A visualization helps to **communicate** and **explore** the graphs/networks.
- We need **algorithms** to draw graphs, and make graphs and networks accessible to people.
How to draw graphs?

Guiding questions

- what makes a drawing good?
- what is a bad drawing?
- are there measures to quantify that?
How to draw graphs?

**Guiding questions**

- what makes a drawing good?
- what is a bad drawing?
- are there measures to quantify that?

**Warm-up** We will draw couple of graphs

- Teams of two
- ”tasks“ in form of adjacency matrix/list
- Given time 15 minutes
- We will shortly discuss the results afterwards
- https://www.yworks.com/downloads/#yEd
Network Visualizations

a small Film
Biblical characters and events (1202)

Source: Joachim de Fiore
"Tree of Life" (1516)

Source: Paul Riccius, Portae Lucis
Geometrical Concepts (1587)

Source: Christophe de Savigny
Genealogical Tree (1879)

Source: Ernst Haeckel
Sociogramm (1933)

Source: Moreno, 1933
Social Network – Organization within UBS

*Member of the Group Executive Board
CPAN Developer-Graph

Source: cpan-explorer.org
last.fm Graph of musics as political map

(Gansner, Hu, Kobourov: GMap, 2009)
last.fm Graph of musics as political map

(Gansner, Hu, Kobourov: GMap, 2009)
Blogosphere 2004 Elections USA

Source: Adamic, Glance, 2005
Social Network – World Finance System

World Finance Corporation
© Mark Lombardi
Temporal Graph Layout: Storylines

Source: ABC news, Australia
Traffic network – Highways USA
Traffic network – Highways USA

THE EISENHOWER INTERSTATE SYSTEM
(SIMPLIFIED)
London Tube Map (1933)

Source: Henry Beck
Co-centric Tube Map

Source: Maxwell Roberts
Curvilinear S/U-bahn map

Source: KVV and Maxwell Roberts
Flight Connections
Flow-Map: Whiskey Export

Source: Verbeek, Buchin, Speckmann., 2011
Monitoring of Energy Network

Source: Eir Grid, Ireland
Wiring plan/ Cable plan

Please note that the wiring plan is not labeled for the purpose of demonstration. The components listed are for reference:

* **Terminal Connections:**
  - For T2, T3, T4, T5

* **Color Codes:**
  - Red: +, Black: -

* **Battery Current:**
  - No Signal: 50 mA
  - Output: 14 mA

* **Voltage Measurements:**
  - All voltage values are given in volts.
  - All resistance values are in ohms.

* **Antenna Assembly:**
  - Frequency Loop Antenna

* **Transformer Connections:**
  - Types 234 & 235

* **Miscellaneous:**
  - Remote Loop Antenna
  - Color Code: Red (+), Black (-)

* **Additional Notes:**
  - All components are standard and are not custom made.
  - All measurements are approximate and may vary.

* **Custom Instructions:**
  - All components are ready for installation.
  - All connections are tight and secure.

* **Special Instructions:**
  - All components are grounded.
  - All labels are clearly marked.

* **Manufacturer:**
  - Custom Electronics

* **Model:**
  - Wiring Plan/ Cable Plan

* **Version:**
  - Latest Edition

* **Disclaimer:**
  - All information provided is based on the manufacturer's specifications and is subject to change without notice. Use of this wiring plan is at your own risk. For professional use, please consult a licensed electrician.
Medicine – Diseases
Medicine – phylogenetic Tree
Software-Network – UML Diagram
Clustered Software-Graph in 3D

Source: Balzer, Deussen, 2007
Software Call-Graph with edge-bundling

Source: Danny Holten, 2011
Web Trend Map

Source: information Architects, 2009
Large Graphs – Object Mesh

Source: Yifan Hu
Alternative Visualizations: Explorer vs Treemap
Alternative Visualizations: Contact map
# Tools

## Libraries for graph visualization

- JUNG jung.sourceforge.net (Java)
- OGDF www.ogdf.net (C++)

## Visualization tools

- visone visone.info
- graphviz www.graphviz.org
- yEd www.yworks.com
- Gephi www.gephi.org
Basic Definitions
Visual Variables according to Bertin (1967)

- **position**
- **size**
- **value**
- **shape**
- **orientation**
- **color**
- **texture**
Visual Variables according to Bertin (1967)

- position
- size
- value
- shape
- orientation
- color
- texture

→ Layout problem
Layout Problem

Restriction in the following: **drawing** is always meant to be in form of a node-link diagram

---

**Graph visualization problem**

**given**: Graph $G = (V, E)$

**find**: good drawing $\Gamma$ of $G$

- $\Gamma : V \rightarrow \mathbb{R}^2$, nodes $v \mapsto$ point $\Gamma(v)$
- $\Gamma : E \rightarrow$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple open curve $c_{uv} : [0, 1] \rightarrow \mathbb{R}^2$ where $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$
Layout Problem

Restriction in the following: **drawing** is always meant to be in form of a node-link diagram

Graph visualization problem

given: Graph $G = (V, E)$
find: **good** drawing $\Gamma$ of $G$

- $\Gamma : V \rightarrow \mathbb{R}^2$, nodes $v \mapsto$ point $\Gamma(v)$
- $\Gamma : E \rightarrow$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple open curve $c_{uv} : [0, 1] \rightarrow \mathbb{R}^2$ where $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$

But what is a **good** Drawing?
Requirement to the graph layout

1) **Drawing conventions**, required properties, for example
   - straight-line edges
   - orthogonal edges (with bends 90 degrees)
   - Drawing on a grid
   - crossing-free
   - ...
Requirement to the graph layout

1) **Drawing conventions**, required properties

2) **Aesthetics** (to be optimized), for example:

   - Number of crossing
   - Number of bends
   - Uniform edge length
   - Area/length
   - Angular resolution
   - Symmetry
   - ...

→ often lead to NP-hard optimization problems!
→ often several competing criteria
Requirement to the graph layout

1) **Drawing conventions**, required properties

2) **Aesthetics** (to be optimized)

3) **Partial/local constraints**, for example:
   - Positions of several vertices
   - Relative positions of vertices
   - Group of nodes drawn close to each other
## Layout Problem – Second Attempt

**Graph visualization problem**

**given:** Graph $G = (V, E)$

**find:** good drawing $\Gamma$ of $G$, that
- complies with drawing conventions
- optimizes aesthetics
- satisfies local/partial constraints
Graph visualization problem

given: Graph $G = (V, E)$
find: good drawing $\Gamma$ of $G$, that
- complies with drawing conventions
- optimizes aesthetics
- satisfies local/partial constraints

- this definition drives to interesting algorithmic problems
Lecture topics

- Drawings of Trees and other recursively defined graph classes
- straight-line drawings of planar graphs
- incremental layouts
- orthogonal drawings
- contact representation of graphs
- hierarchical drawings of layered graphs
- force-based drawing algorithm
- ...
Lecture topics

- Drawings of Trees and other recursively defined graph classes
- straight-line drawings of planar graphs
- incremental layouts
- orthogonal drawings
- contact representation of graphs
- hierarchical drawings of layered graphs
- force-based drawing algorithm
- ...

Next Meetings
Lecture on 24.10 9:45
Exercise on 26.10 9:45 → please bring your Laptops
→ install the Software yEd (www.yworks.com)