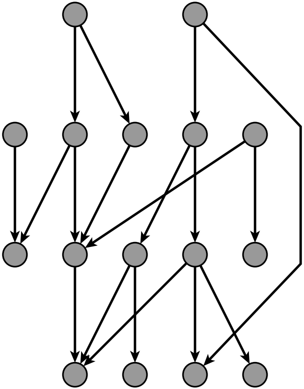
A Flow Formulation for Horizontal Coordinate Assignment with Prescribed Width

Michael Jünger[1], Petra Mutzel[2], Christiane Spisla[2]

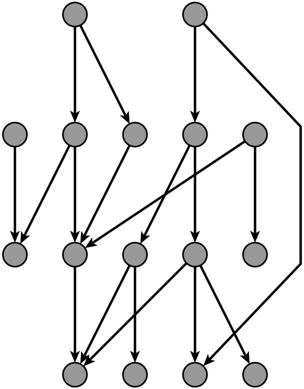
[1]University of Cologne, [2]TU Dortmund University

Symposium on Graph Drawing
and Network Visualization 2018



Sugiyama framework

- (Cycle breaking)
- Layer assignment
- Crossing minimization
- Coordinate assignment
- (Edge routing)



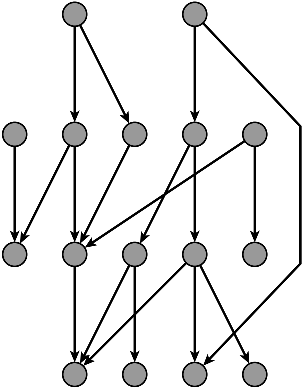
Coordinate assignment

Input:

- DAG G
- Layering $\mathcal{L}$
- Vertex ordering *ord*

Output:

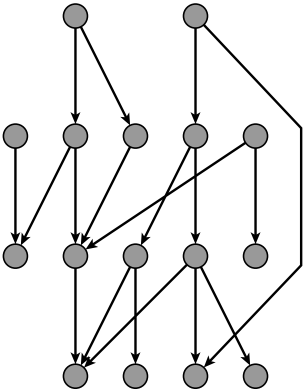
- Feasible x-coordinates, i.e. $x(u) + \delta \leq x(v)$, if $ord(u) < ord(v)$



Coordinate assignment

Goals:

- Short horizontal edge lengths
  →[Gansner et al., 1993]

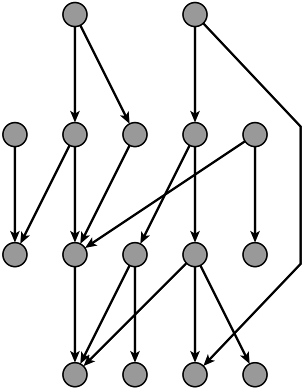


Coordinate assignment

Goals:

- Short horizontal edge lengths
  →[Gansner et al., 1993]
- Balanced node positions
  →[Buchheim et al., 2000],
  →[Brandes and Köpf, 2001]

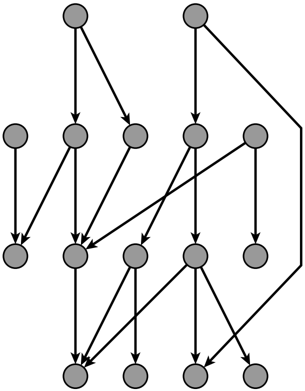
Introduction



Coordinate assignment

Goals:

- Short horizontal edge lengths
  ⟶[Gansner et al., 1993]
- Balanced node positions
  ⟶[Buchheim et al., 2000],
  ⟶[Brandes and Köpf, 2001]
- Vertical edges



Coordinate assignment

Goals:

- Short horizontal edge lengths
  →[Gansner et al., 1993]
- Balanced node positions
  →[Buchheim et al., 2000],
  →[Brandes and Köpf, 2001]
- Vertical edges

What about the width?

Introduction

- Width of the layering:
  maximum number of nodes in a layer →[Coffman and Graham, 1972]

Introduction

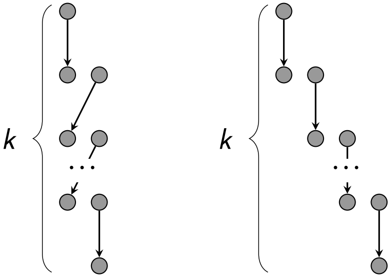
- Width of the layering:
  maximum number of nodes in a layer $\longrightarrow$[Coffman and Graham, 1972]

$\neq$ Width of the drawing:
  maximum horizontal distance between any two nodes

- Width of the layering:
  maximum number of nodes in a layer ⟶ [Coffman and Graham, 1972]

≠ Width of the drawing:
  maximum horizontal distance between any two nodes

width $= 1$
edge length $= k-3$



width $= k-2$
edge length $= 0$

Our Contribution

Efficient algorithm for drawings with minimized total edge length under the restriction of bounded drawing width.

Our Contribution

Efficient algorithm for drawings with minimized total edge length under the restriction of bounded drawing width.

In addition it realizes

- vertical inner segments
- minimum/maximum/exact distances between nodes

Our Contribution

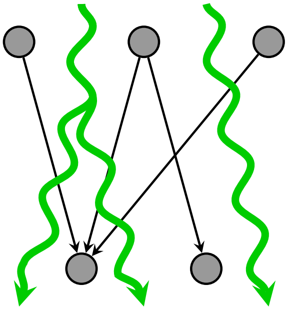
Efficient algorithm for drawings with minimized total edge length under the restriction of bounded drawing width.

In addition it realizes

- vertical inner segments
- minimum/maximum/exact distances between nodes

Method: Minimum cost flow formulation
for the coordinate assignment problem

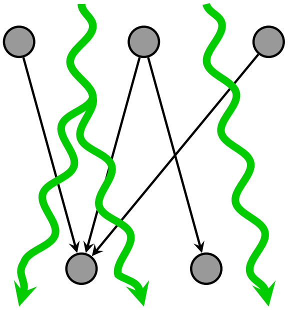
Network Construction



Idea:

- Network with super source and super sink
- Send flow from top to bottom

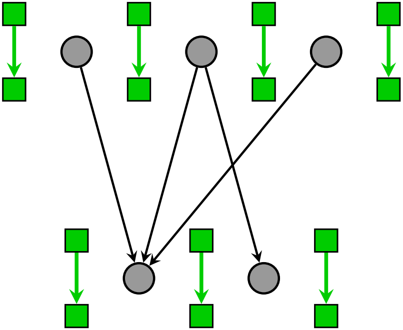
Network Construction



Idea:

- Network with super source and super sink
- Send flow from top to bottom
- Flow = horizontal distance
- Cost = edge length

Network Construction



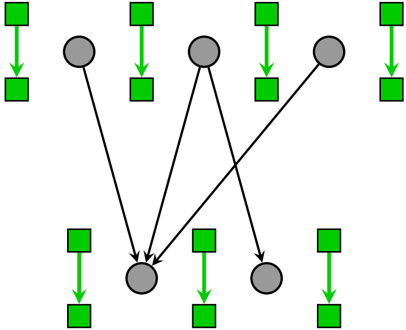
Minimum distance δ between nodes

$+$

Flow means distance

$\Rightarrow$ Network arcs between nodes with lower bound δ

Network Construction

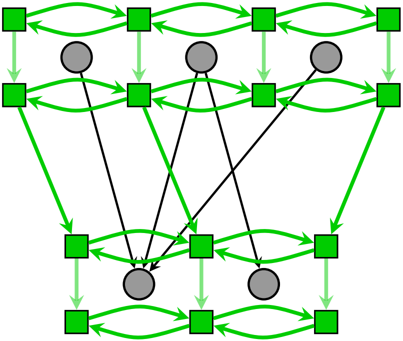


⇒ x-coordinate for node v:

$$x(v) = \sum_{a \text{ left of } v} f(a),$$

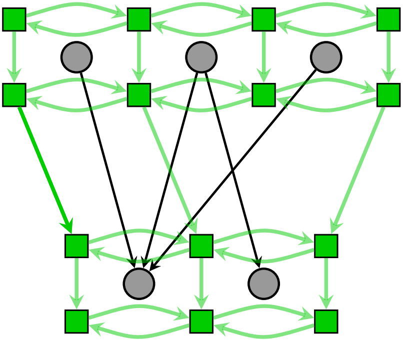
where $f(a)$ is the flow over network arc a

Network Construction



Send flow along layers and
over some inter-layer connections
without lower or upper bounds

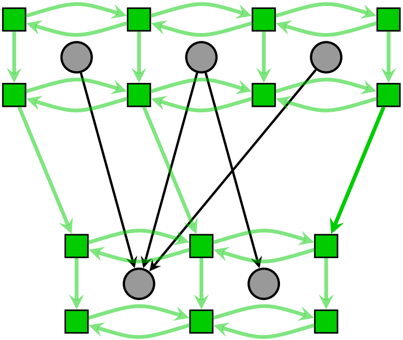
Network Construction



Connections between layers:

- arc from leftmost upper node to leftmost lower node

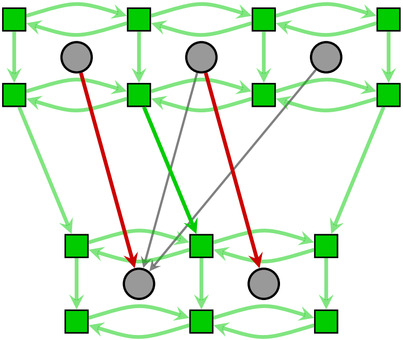
Network Construction



Connections between layers:

- arc from leftmost upper node to leftmost lower node
- arc from rightmost upper node to rightmost lower node

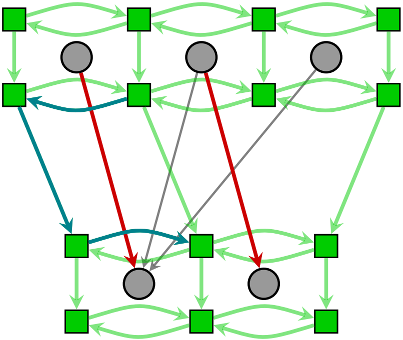
Network Construction



Connections between layers:

- arc from leftmost upper node to leftmost lower node
- arc from rightmost upper node to rightmost lower node
- in *hug* situations: two network nodes enclosed by graph edges

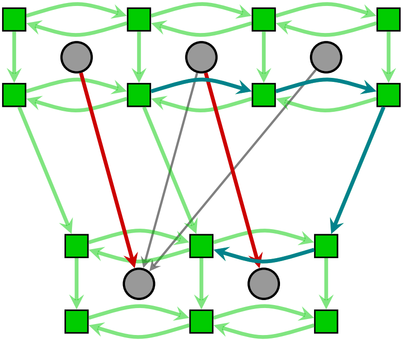
Network Construction



Connections between layers:

- arc from leftmost upper node to leftmost lower node
- arc from rightmost upper node to rightmost lower node
- in *hug* situations: two network nodes enclosed by graph edges

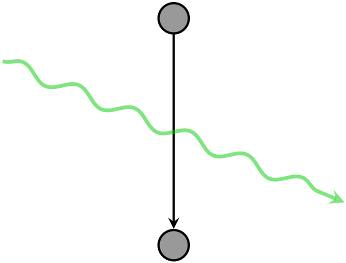
Network Construction



Connections between layers:

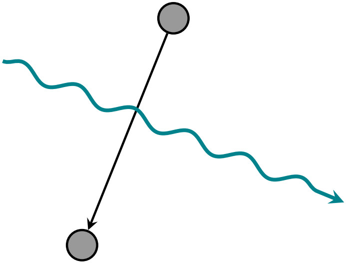
- arc from leftmost upper node to leftmost lower node
- arc from rightmost upper node to rightmost lower node
- in *hug* situations: two network nodes enclosed by graph edges

Network Construction

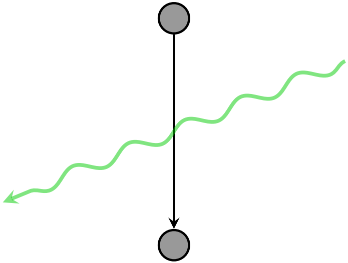


Cost corresponds to edge length
and flow corresponds to distance

Network Construction

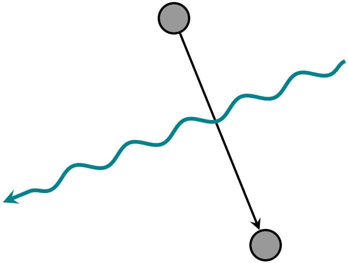


Cost corresponds to edge length
and flow corresponds to distance



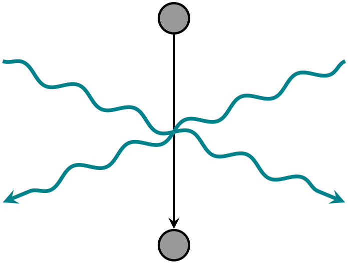
Cost corresponds to edge length
and flow corresponds to distance

Network Construction



Cost corresponds to edge length
and flow corresponds to distance

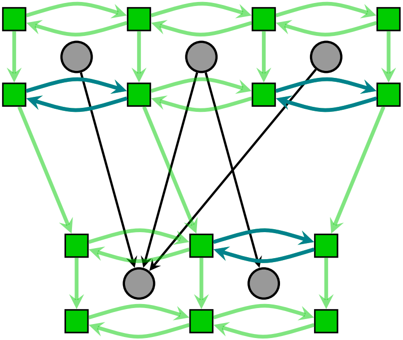
Network Construction



Cost corresponds to edge length
and flow corresponds to distance

$\Rightarrow$ cost = number of crossed edges

Network Construction

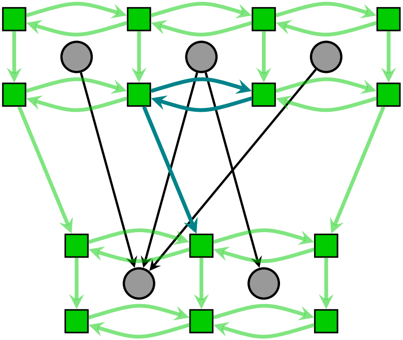


$\text{cost} = 1$

$\text{lower bound} = 0$

$\text{upper bound} = \infty$

Network Construction

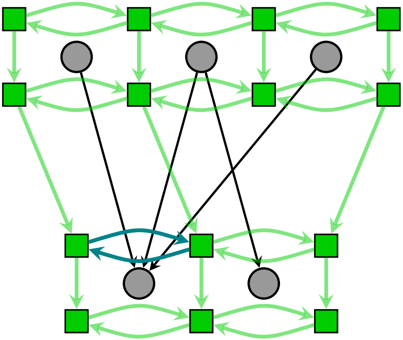


$\text{cost} = 2$

$\text{lower bound} = 0$

$\text{upper bound} = \infty$

Network Construction

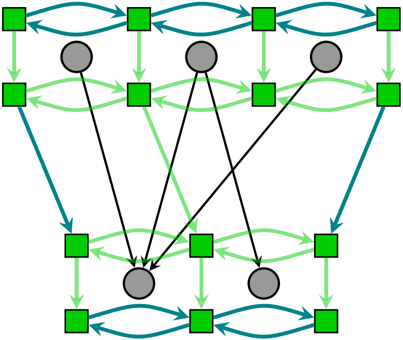


cost = 3

lower bound = 0

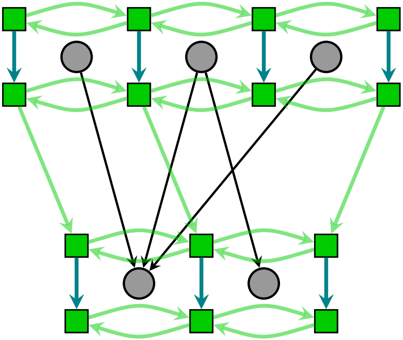
upper bound = ∞

Network Construction



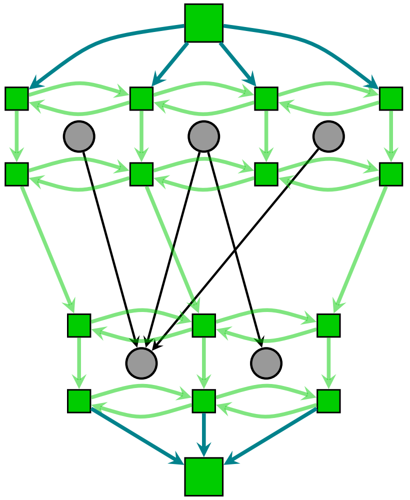
$$\text{cost} = 0$$
$$\text{lower bound} = 0$$
$$\text{upper bound} = \infty$$

Network Construction



$$\text{cost} = 0$$
$$\text{lower bound} = \delta \text{ (or 0)}$$
$$\text{upper bound} = \infty$$

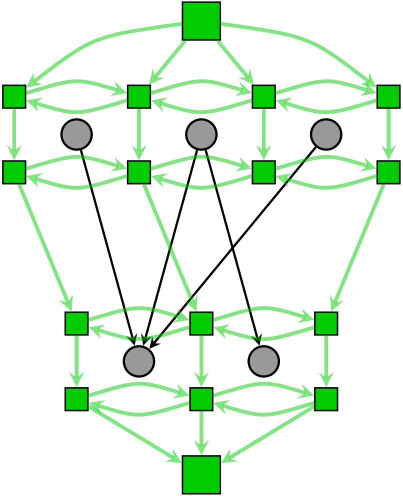
Network Construction



super source and super sink

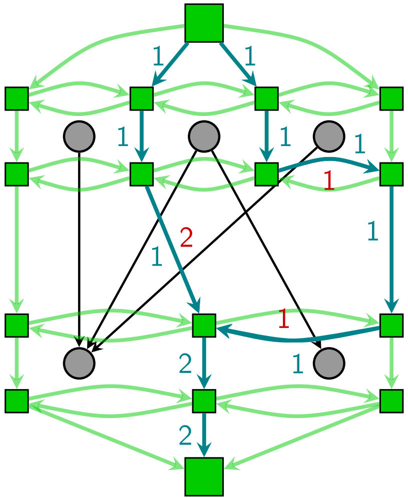
$$\text{cost} = 0$$
$$\text{lower bound} = 0$$
$$\text{upper bound} = \infty$$

Example Flows and Corresponding Drawings



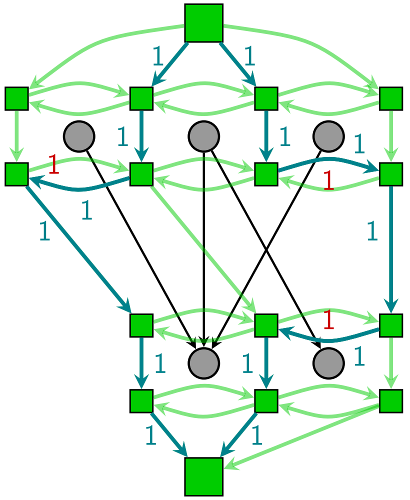
$\delta = 1$
flow in blue
arc cost in red

Example Flows and Corresponding Drawings



$$\sum \text{cost} \cdot \text{flow} = 2 \cdot 1 + 1 \cdot 1 + 1 \cdot 1$$
$$= 4$$

Example Flows and Corresponding Drawings



$$\sum \text{cost} \cdot \text{flow} = 2 \cdot 1 + 1 \cdot 1$$
$$= 3$$

Results

Main Theorem

Every minimum cost flow in network described above defines feasible x-coordinates and minimum total horizontal edge length.

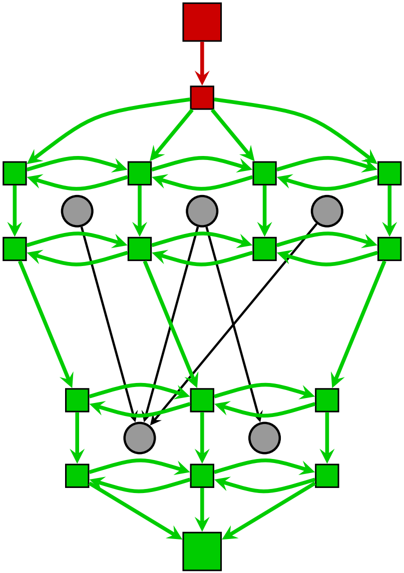
Results

Main Theorem

Every minimum cost flow in network described above defines feasible x-coordinates and minimum total horizontal edge length.

What about other contraints?

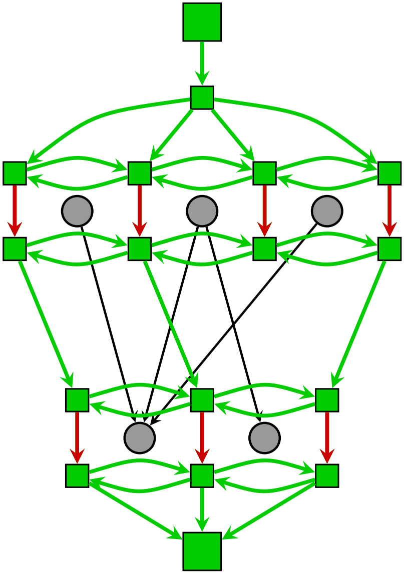
Further constraints



Width:

split super source and
set upper bound of new arc
to maximum width

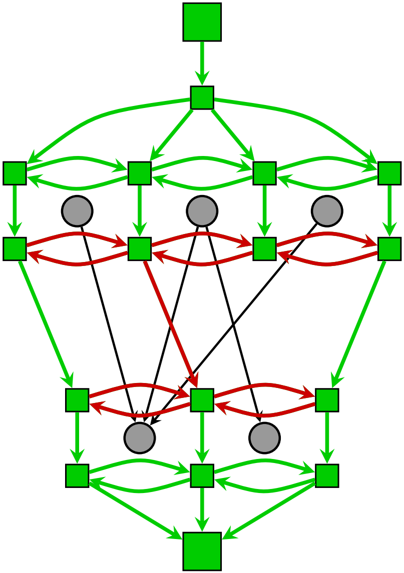
Further constraints



Maximum/exact distances:

set bounds to
appropriate values

Further constraints

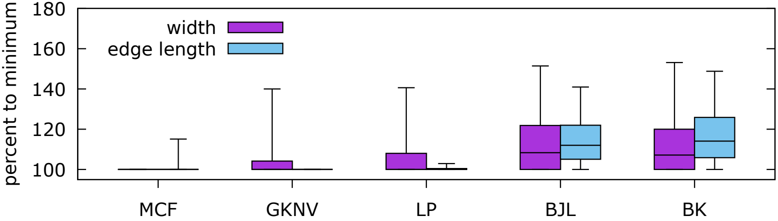


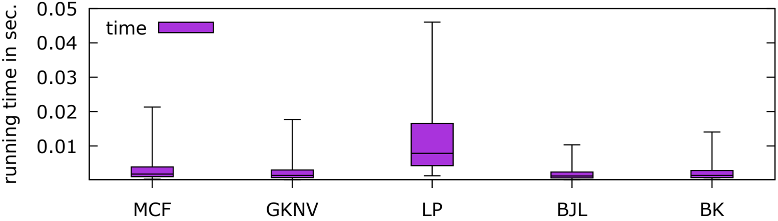
Vertical edges:

delete appropriate arcs

Computational results

Test set: 1277 graphs, |V| = 10-100 (subset of AT&T)
Implementation: OGDF, network simplex algorithm



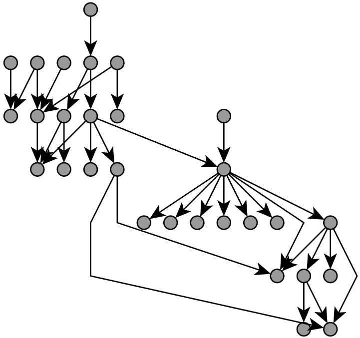


Conclusion

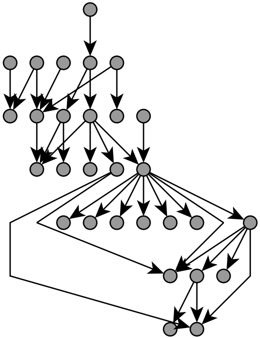
- Minimum cost flow formulation for the coordinate assignment problem
- Minimize edge length while respecting further constraints
- Comparable running time to state-of-the-art algorithms

Conclusion

- Minimum cost flow formulation for the coordinate assignment problem
- Minimize edge length while respecting further constraints
- Comparable running time to state-of-the-art algorithms



GKNV
width: 13, edge length: 54



MCF
width: 9, edge length: 58

References I

Brandes, U. and Köpf, B. (2001).
Fast and simple horizontal coordinate assignment.
In Mutzel, P., Jünger, M., and Leipert, S., editors, *Graph Drawing, 9th International Symposium, GD 2001 Vienna, Austria, September 23-26, 2001, Revised Papers*, volume 2265 of *Lecture Notes in Computer Science*, pages 31–44. Springer.

Buchheim, C., Jünger, M., and Leipert, S. (2000).
A fast layout algorithm for k-level graphs.
In Marks, J., editor, *Graph Drawing, 8th International Symposium, GD 2000, Colonial Williamsburg, VA, USA, September 20-23, 2000, Proceedings*, volume 1984 of *Lecture Notes in Computer Science*, pages 229–240. Springer.

References II

Coffman, E. G. and Graham, R. L. (1972).
Optimal scheduling for two-processor systems.
Acta Informatica, 1(3):200–213.

Gansner, E. R., Koutsofios, E., North, S. C., and Vo, K.-P. (1993).
A technique for drawing directed graphs.
Software Engineering, 19(3):214–230.