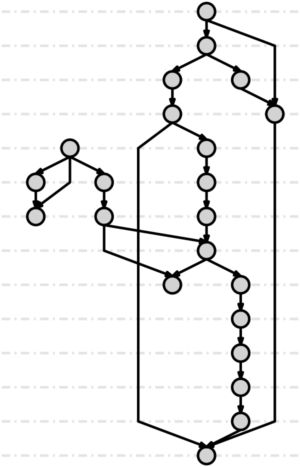
A Natural Quadratic Approach to the Generalized Graph Layering Problem

Sven Mallach

Department of Mathematics & Computer Science
University of Cologne, Germany

Graph Drawing & Network Visualization
Pruhonice, 20th September 2019

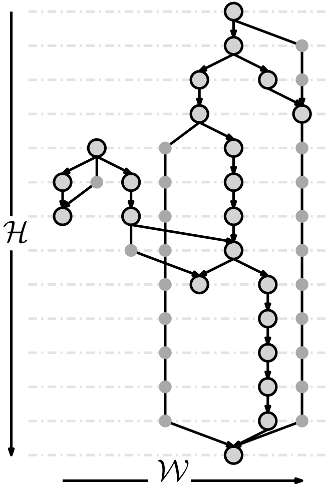
Layered Drawings of Directed Graphs



Drawing Restictions:

- Vertices on consecutive layers
- No two adjacent vertices on the same layer
- (Major) Common arc direction

Layered Drawings of Directed Graphs

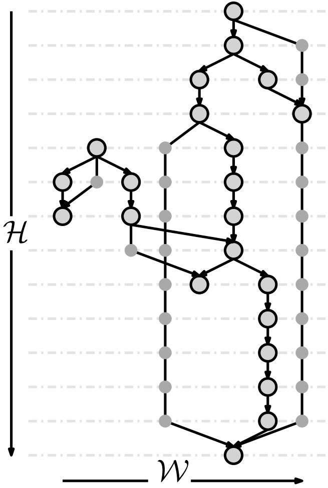


Drawing Restictions:

- Vertices on consecutive layers
- No two adjacent vertices on the same layer
- (Major) Common arc direction

- Aesthetic *layering* objectives:
  - 'Compactness' (Width $\mathcal{W}$, Height $\mathcal{H}$, Total Arc Length),
  - Few Arc Reversals

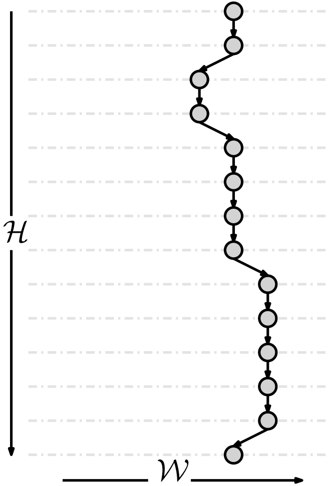
Sugiyama-Style Drawings of Directed Graphs



Classic Approach
(Sugiyama et al. [1981]):

1. Cycle Removal
2. **Vertex Layering**
3. Crossing Minimization
4. Horizontal Coordinates & Arc Routing

Sugiyama-Style Drawings of Directed Graphs



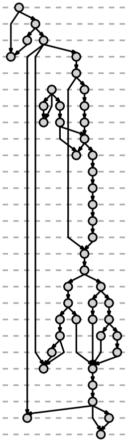
Classic Approach
(Sugiyama et al. [1981]):

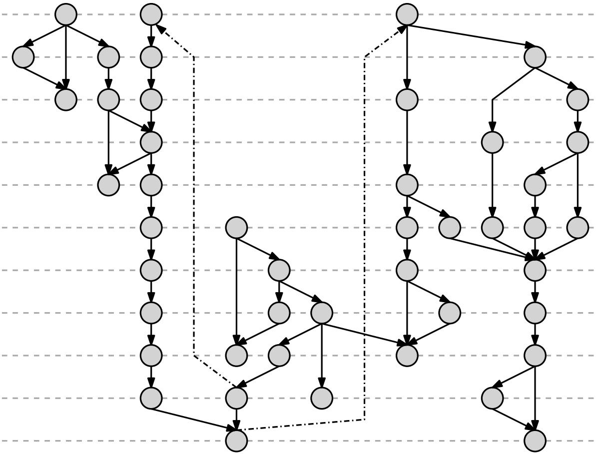
1. Cycle Removal
2. **Vertex Layering**
3. Crossing Minimization
4. Horizontal Coordinates & Arc Routing

Limitations w.r.t. steps 1 & 2:
Longest path may impede 'compactness' / good aspect ratio from the very beginning.

Visual effects of poor and good aspect ratios

Two drawings of a graph, the right of which has *two* arcs reversed.



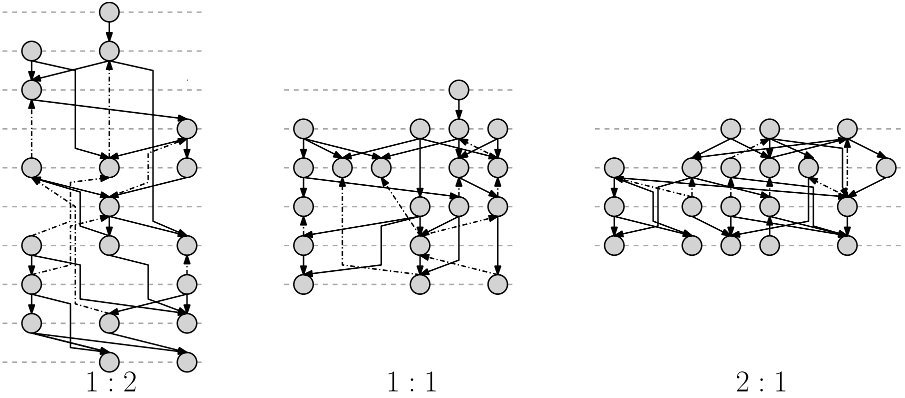


Area-Adaptive Graph Layering

Rüegg et al. [2017]: Adapt Layering w.r.t. *target drawing area*.

Input: (Relative) Area width r_W and height r_H, denoted $r_W : r_H$.

Goal: Maximum Resolution or *Scaling Factor* $\mathcal{S} := \min\{\frac{r_W}{\mathcal{W}}, \frac{r_H}{\mathcal{H}}\}$
(plus possibly minimum edge length / number of reversed arcs).



Area-Adaptive Graph Layering

Rüegg et al. [2017]: Adapt Layering w.r.t. *target drawing area*.

Input: (Relative) Area width r_W and height r_H, denoted $r_W : r_H$.

Goal: Maximum Resolution or *Scaling Factor* $\mathcal{S} := \min\{\frac{r_W}{\mathcal{W}}, \frac{r_H}{\mathcal{H}}\}$
(plus possibly minimum edge length / number of reversed arcs).

Maximum-Scale Generalized Layering Problem (GLP-MS)
Given $G = (V, A)$, r_W, and r_H, find a feasible layering $L : V \mapsto \mathbb{N}_+$ minimizing

$$\omega_{len} \Big(\sum_{uv \in A} |L(v) - L(u)|\Big) + \omega_{rev} \, |\{uv \in A \mid L(v) < L(u)\}| - \omega_{scl} \, \mathcal{S}$$

Graph Layering - Evolution of Optimization Problems

Name	Objective	Exact Approach
DLP	$\sum_{uv \in A} \big(L(v) - L(u)\big)$	Gansner et al. [1993]
DLP-W	$\sum_{uv \in A} \omega_{len} \big(L(v) - L(u)\big) + \omega_{wid}\, \mathcal{W}$	Healy, Nikolov [GD 2002]
GLP	$\sum_{uv \in A} \omega_{len}\, \lvert L(v) - L(u) \rvert +$	
	$\omega_{rev}\, \lvert \{uv \in A \mid L(v) < L(u)\} \rvert$	Rüegg et al. [GD 2016]
GLP-W	$\sum_{uv \in A} \omega_{len}\, \lvert L(v) - L(u) \rvert +$	
	$\omega_{rev}\, \lvert \{uv \in A \mid L(v) < L(u)\} \rvert + \omega_{wid}\, \mathcal{W}$	Jabrayilov et al. [GD 2016]
GLP-MS*	$\sum_{uv \in A} \omega_{len}\, \lvert L(v) - L(u) \rvert +$	
	$\omega_{rev}\, \lvert \{uv \in A \mid L(v) < L(u)\} \rvert + \omega_{scl}\, \bar{\mathcal{S}}$	Rüegg et al. [JGAA 2017]

$(\bar{\mathcal{S}} := \frac{1}{\mathcal{S}})$

Graph Layering and Mixed-Integer *Linear* Programming

Prior models are based on either *assignment* or *ordering* variables.

Graph Layering and Mixed-Integer *Linear* Programming

Prior models are based on either *assignment* or *ordering* variables.

Assignment variables: $x_{v,k} := \begin{cases} 1, & \text{if } L(v) = k \\ 0, & \text{otherwise} \end{cases}$

Ordering variables: $y_{k,v} := \begin{cases} 1, & \text{if } L(v) > k \\ 0, & \text{otherwise} \end{cases}$

Graph Layering and Mixed-Integer *Linear* Programming

Prior models are based on either *assignment* or *ordering* variables.

Assignment variables: $x_{v,k} := \begin{cases} 1, & \text{if } L(v) = k \\ 0, & \text{otherwise} \end{cases}$

Ordering variables: $y_{k,v} := \begin{cases} 1, & \text{if } L(v) > k \\ 0, & \text{otherwise} \end{cases}$

Linear expression of restrictions and objectives?

Graph Layering and Mixed-Integer *Linear* Programming

Prior models are based on either *assignment* or *ordering* variables.

Assignment variables: $x_{v,k} := \begin{cases} 1, & \text{if } L(v) = k \\ 0, & \text{otherwise} \end{cases}$

Ordering variables: $y_{k,v} := \begin{cases} 1, & \text{if } L(v) > k \\ 0, & \text{otherwise} \end{cases}$

Linear expression of restrictions and objectives?

- Easy if arc directions are fixed (DLP cases).

Graph Layering and Mixed-Integer *Linear* Programming

Prior models are based on either *assignment* or *ordering* variables.

Assignment variables: $x_{v,k} := \begin{cases} 1, & \text{if } L(v) = k \\ 0, & \text{otherwise} \end{cases}$

Ordering variables: $y_{k,v} := \begin{cases} 1, & \text{if } L(v) > k \\ 0, & \text{otherwise} \end{cases}$

Linear expression of restrictions and objectives?

- Easy if arc directions are fixed (DLP cases).
- DLP-W: Dummy vertex variables:
  $d_{uv,k} := \begin{cases} 1, & \text{if } uv \in A \text{ spans layer } k \\ 0, & \text{otherwise} \end{cases}$
- Additional option to “count” edge lengths.

But: ***Variable Arc Directions*** change the scene:

But: ***Variable Arc Directions*** change the scene:

- Need to count arc reversals in addition

  $\Rightarrow$ Need arc reversal variables: $r_{uv} := \begin{cases} 1, & \text{if } L(v) < L(u) \\ 0, & \text{otherwise} \end{cases}$

Graph Layering and Mixed-Integer *Linear* Programming

But: ***Variable Arc Directions*** change the scene:

- Need to count arc reversals in addition
  $\Rightarrow$ Need arc reversal variables: $r_{uv} := \begin{cases} 1, & \text{if } L(v) < L(u) \\ 0, & \text{otherwise} \end{cases}$
- Need to model $|L(v) - L(u)|$ (instead of $L(v) - L(u)$).
- Need to model dummy vertices based on two possible arc directions.

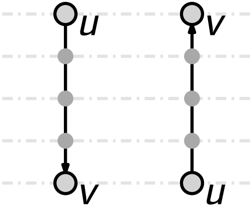
Graph Layering and Mixed-Integer *Linear* Programming

But: ***Variable Arc Directions*** change the scene:

- Need to count arc reversals in addition
  $\Rightarrow$ Need arc reversal variables: $r_{uv} := \begin{cases} 1, & \text{if } L(v) < L(u) \\ 0, & \text{otherwise} \end{cases}$
- Need to model $|L(v) - L(u)|$ (instead of $L(v) - L(u)$).
- Need to model dummy vertices based on two possible arc directions.
- Case Distinctions: **More** and **weaker** *linear* constraints to enforce correct values on r_{uv} and $d_{uv,k}$.

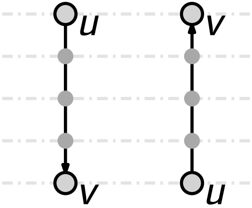
A *Quadratic* Perspective on Graph Layering

Graph Layering is of *quadratic* nature - not only geometrically.



A *Quadratic* Perspective on Graph Layering

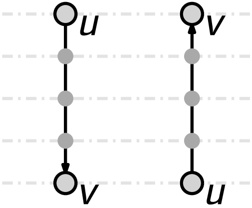
Graph Layering is of *quadratic* nature - not only geometrically.



- Arc directions, (absolute) edge lengths, and dummy vertices are all based on ***conjunctive*** vertex placement decisions.

A *Quadratic* Perspective on Graph Layering

Graph Layering is of *quadratic* nature - not only geometrically.

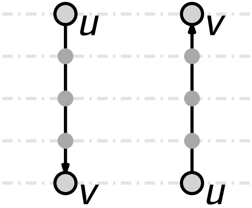


- Arc directions, (absolute) edge lengths, and dummy vertices are all based on ***conjunctive*** vertex placement decisions.

Idea: Model restrictions and objective from a *quadratic* assignment perspective (and linearize afterwards).

A *Quadratic* Perspective on Graph Layering

Graph Layering is of *quadratic* nature - not only geometrically.



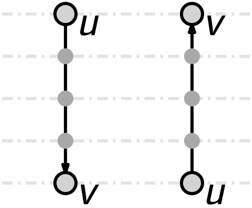
- Arc directions, (absolute) edge lengths, and dummy vertices are all based on ***conjunctive*** vertex placement decisions.

Idea: Model restrictions and objective from a *quadratic* assignment perspective (and linearize afterwards).

- There is a **stronger and compact** linearization technique.

A *Quadratic* Perspective on Graph Layering

Graph Layering is of *quadratic* nature - not only geometrically.



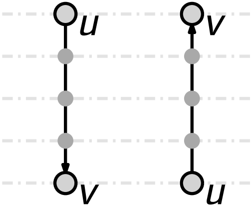
- Arc directions, (absolute) edge lengths, and dummy vertices are all based on ***conjunctive*** vertex placement decisions.

Idea: Model restrictions and objective from a *quadratic* assignment perspective (and linearize afterwards).

- There is a **stronger and compact** linearization technique.
- For any arc $uv \in A$, there is exactly one pair of layers k and ℓ, $k \neq \ell$, such that $x_{u,k} \cdot x_{v,\ell} = 1$. All other products are zero.

A *Quadratic* Perspective on Graph Layering

Graph Layering is of *quadratic* nature - not only geometrically.



- Arc directions, (absolute) edge lengths, and dummy vertices are all based on ***conjunctive*** vertex placement decisions.

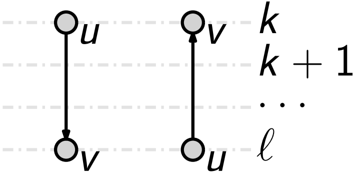
Idea: Model restrictions and objective from a *quadratic* assignment perspective (and linearize afterwards).

- There is a **stronger and compact** linearization technique.
- For any arc $uv \in A$, there is exactly one pair of layers k and ℓ, $k \neq \ell$, such that $x_{u,k} \cdot x_{v,\ell} = 1$. All other products are zero.
- Assignment variables more intuitive than ordering variables.

A *Quadratic Assignment* Perspective on Graph Layering

If there are Y layers, the length of $uv \in A$ thus equals

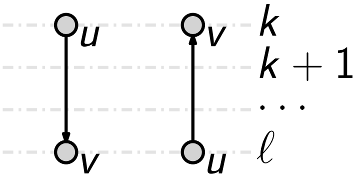
$$\sum_{\ell=2}^{Y} \sum_{k=1}^{\ell-1} \left((\ell - k) \cdot (x_{u,\ell} \cdot x_{v,k} + x_{u,k} \cdot x_{v,\ell}) \right)$$



A *Quadratic Assignment* Perspective on Graph Layering

If there are Y layers, the length of $uv \in A$ thus equals

$$\sum_{\ell=2}^{Y}\sum_{k=1}^{\ell-1}\left((\ell-k)\cdot(x_{u,\ell}\cdot x_{v,k}+x_{u,k}\cdot x_{v,\ell})\right)$$



An arc $uv \in A$ is reversed if and only if the expression

$$\sum_{\ell=2}^{Y}\left(x_{u,\ell}\cdot\sum_{k=1}^{\ell-1}x_{v,k}\right)$$

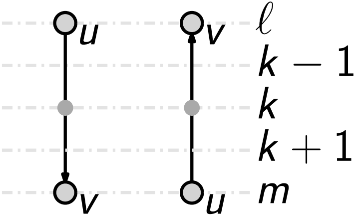
evaluates to one. Otherwise, the expression is zero.

A *Quadratic Assignment* Perspective on Graph Layering

An arc $uv \in A$ causes a dummy vertex on layer $k \in \{2, \ldots, Y-1\}$ if and only if k is between the layers of u and v, i.e., if

$$\sum_{\ell=1}^{k-1} \sum_{m=k+1}^{Y} (x_{u,\ell} \cdot x_{v,m} + x_{u,m} \cdot x_{v,\ell})$$

evaluates to one. Again, the term will be zero otherwise.



A Basic Quadratic Layer Assignment Model (QLA)

Replace the product $x_{u,k} \cdot x_{v,\ell}$ by variables $p_{u,k,v,\ell}$ for all $uv \in A$ and all $k, \ell \in \{1, \dots, Y\}$.

A Basic Quadratic Layer Assignment Model (QLA)

Replace the product $x_{u,k} \cdot x_{v,\ell}$ by variables $p_{u,k,v,\ell}$ for all $uv \in A$ and all $k, \ell \in \{1, \ldots, Y\}$.

Then a feasible layering is characterized by the restrictions:

$$
\begin{aligned}
\sum_{k=1}^{Y} x_{v,k} &= 1 && \text{for all } v \in V \\
\sum_{\ell=1}^{Y} p_{u,k,v,\ell} &= x_{u,k} && \text{for all } uv \in A,\ k \in \{1, \ldots, Y\} \\
\sum_{k=1}^{Y} p_{u,k,v,\ell} &= x_{v,\ell} && \text{for all } uv \in A,\ \ell \in \{1, \ldots, Y\} \\
p_{u,k,v,k} &= 0 && \text{for all } uv \in A,\ k \in \{1, \ldots, Y\} \\
x_{v,k} &\in \{0, 1\} && \text{for all } v \in V,\ k \in \{1, \ldots, Y\} \\
p_{u,k,v,\ell} &\in [0, 1] && \text{for all } uv \in A,\ k, \ell \in \{1, \ldots, Y\}
\end{aligned}
$$

Computational Study

Two runtime competitions:

QLA-W vs. CGL-W (Jabrayilov et al. [2016])

QLA-MS* vs. CGL-MS* (Rüegg et al. [2017])

Computational Study

Two runtime competitions:

QLA-W vs. CGL-W (Jabrayilov et al. [2016])

QLA-MS* vs. CGL-MS* (Rüegg et al. [2017])

Model Sizes:

Computational Study

Two runtime competitions:

QLA-W vs. CGL-W (Jabrayilov et al. [2016])

QLA-MS* vs. CGL-MS* (Rüegg et al. [2017])

Model Sizes:

CGL-W/MS* $\quad \approx |V| \cdot Y + |A| \cdot Y$ variables

QLA-W/MS* $\quad \approx |V| \cdot Y + |A| \cdot Y^2$ variables

Computational Study

Two runtime competitions:

QLA-W vs. CGL-W (Jabrayilov et al. [2016])

QLA-MS* vs. CGL-MS* (Rüegg et al. [2017])

Model Sizes:

CGL-W/MS* $\approx |V| \cdot Y + |A| \cdot Y$ variables

QLA-W/MS* $\approx |V| \cdot Y + |A| \cdot Y^2$ variables

CGL-W $\approx (4|A| + |V|) \cdot Y + 4|A|$ constraints

QLA-W $\approx (2|A|) \cdot Y + |V|$ constraints

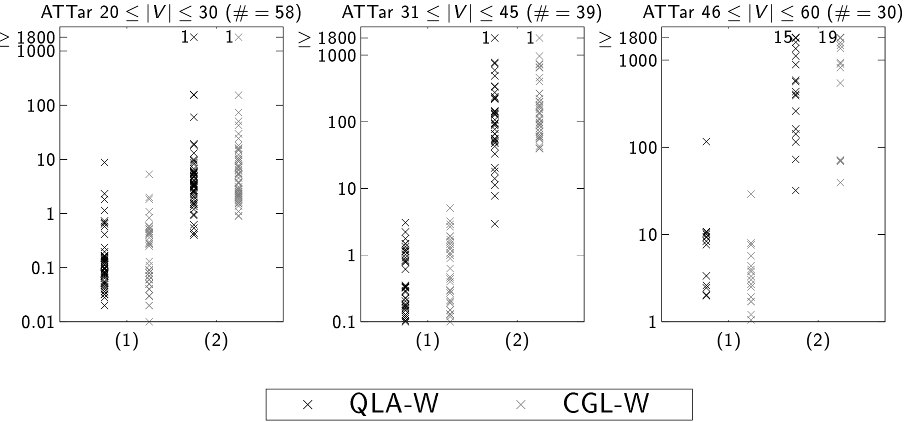
QLA-/CGL-MS* versions: $|V|$ more constraints each.

GLP and GLP-W - Results ATTar (Di Battista et al. [1997])

Two experiments (Gurobi 8, timeout at 1800s (30 min.)):

(1) Almost no width emphasis (GLP setting)

(2) Major emphasis on width minimization

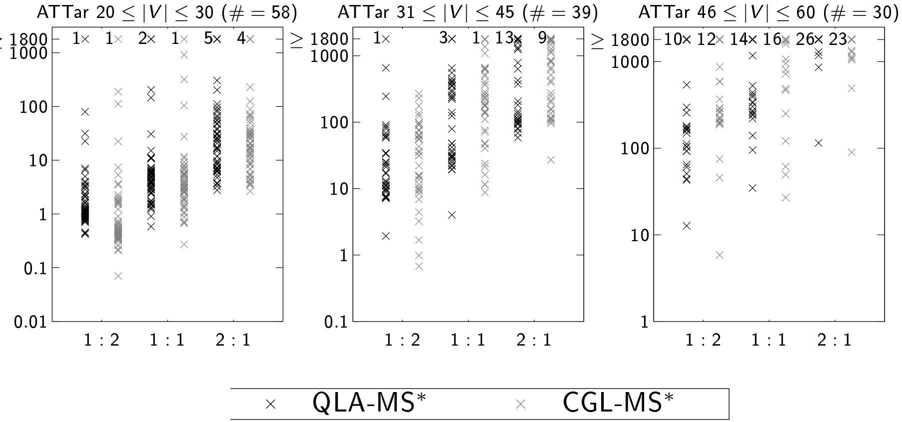


Intel Core i7-3770T (2.5 GHz), 1 Thread, 8 GB RAM, Linux

GLP-MS* - Results ATTar (Di Battista et al. [1997])

Three experiments (Gurobi 8, timeout at 1800s (30 min.)):

- $r_W : r_H$ ratios 1 : 2, 1 : 1, and 2 : 1.
- Major emphasis on maximum scaling factor.



Intel Core i7-3770T (2.5 GHz), 1 Thread, 8 GB RAM, Linux