Compact Hierarchical Graph Drawings via Quadratic Layer Assignment

Sven Mallach

Institut für Informatik
Universität zu Köln, 50923 Köln, Germany

August 13, 2018

Abstract

We propose a new mixed-integer programming formulation that very naturally expresses the layout restrictions of a layered (hierarchical) graph drawing and several associated objectives, such as a minimum total arc length, number of reversed arcs, and width, or the adaptation to a specific drawing area, as a special quadratic assignment problem. Our experiments show that it is competitive to another formulation that we slightly simplify as well.

1 Introduction

We consider a widely used drawing style for directed graphs where each vertex of the graph is assigned to a horizontal *layer* such that no two adjacent vertices are assigned to the same layer and all (or, at least most of the) arcs have a common direction. Frequently, this kind of layout is used to represent *hierarchical* adjacency structures such as, e.g., flow diagrams, or dependency illustrations.

Until today, the dominant hierarchical graph drawing framework considered in research and implemented in software is the one proposed by Sugiyama et al. [14]. It involves a workflow of four successive and interdependent steps for (i) cycle removal, (ii) vertex layering, (iii) crossing minimization, and (iv) horizontal coordinate assignment and arc routing. Classically, step (i) is carried out by solving the feedback arc set problem, i.e., reversing (a minimum number of) arcs such that they all have a common direction in steps (ii)–(iv), and after which the original orientations are finally restored.

In this paper, we concentrate on the first two phases and, in particular, on their integration in order to obtain a two-dimensionally compact layout. Irrespective of whether a directed graph is acyclic originally, or (temporarily) made acyclic in phase (i), the classic approach treats the respective common arc direction as inviolable within the subsequent steps. Thus, the number of vertices on any of its longest paths constitutes a lower bound on the height of the graph's layering and may impede a compact layout from the very beginning. This is true in particular if the longest path comprises a large subset of the graph's vertices and has thus a poor proportion to the width of a classical layout.

As opposed to that, an integration of phases (i) and (ii) permits to identify (a minimum number of) reasonable arcs to be drawn reverse to the intended hierarchical direction in order to support several aesthetic criteria such as a minimum total arc length and minimum width, or the adaptation to a drawing area of a certain aspect ratio. Fig. 1 illustrates the potential aesthetic effects and improved readability when using this approach. Previous experimental evaluations, e.g. in [6, 11, 12, 13]), show that significant improvements in terms of the required drawing area or aspect ratio can be achieved when optimizing the layering w.r.t. the mentioned objectives.

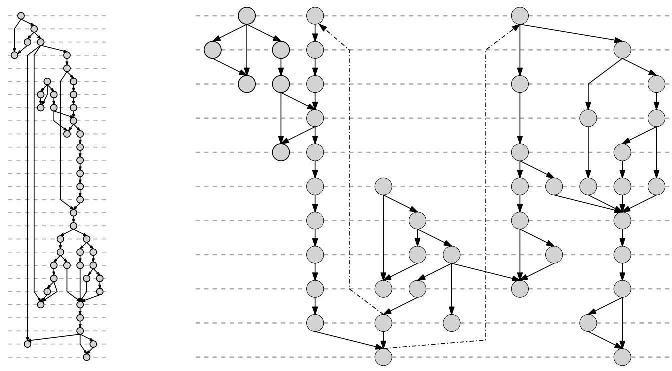


Figure 1: Two layered drawings of the same directed acyclic graph. On the left a classic one, i.e., adhering to its longest path with all arcs pointing downward, and on the right with a much better aspect ratio achieved by reversing only two arcs (marked dash-dotted).

The central contribution of this paper is a new mixed-integer programming (MIP) formulation that very naturally expresses all the mentioned objectives and the conditions of a feasible layering as a special quadratic assignment problem (QAP). Using a common benchmark set, we show that it computationally competes on equal terms with the currently best known MIP formulation. Another advantage of our formulation is that it is more compact in the number of constraints and less sensitive to a graph's density. Due to the newly drawn links to the QAP, we hope that it will as well inspire further combinatorial (heuristic) approaches to tackle larger problem instances or to support interactive user applications.

The paper is organized as follows. In Sect. 2, we restate a series of problem formulations that subsequently generalize on the classical directed graph layering problem and depict the latest state of the art. Parenthetically, we highlight the major existing and particularly exact approaches to solve these. In Sect. 3, we consider the most general and most recently discussed variant in more detail and show up some pitfalls when reformulating it as a MIP. We then present a slightly simplified version of the currently best known such model in Sect. 4 before we present our new one in Sect. 5. After discussing a few remarks in Sect. 6, we report in Sect. 7 on our computational evaluation. Finally, the the paper closes with a conclusion in Sect. 8.

2 Definitions and Related Work

A *layering* L of a directed graph $G = (V, A)$ with vertex set V and arc set A is a mapping $L : V \to \mathbb{N}^+$ that assigns each vertex $v \in V$ a unique *layer* $L(v)$. Classically, and presuming that G is acyclic, a layering L is considered *feasible* if $L(v) - L(u) \geq 1$ holds for all $uv \in A$. In 1993, the following associated problem was introduced and shown to be polynomial time solvable by Gansner et al. [4]:

Problem 1 *Directed Layering Problem (DLP). Let $G = (V, A)$ be a directed acyclic graph. Find a feasible layering L of G such that the total arc length $\sum_{uv \in A}(L(v) - L(u))$ is minimum.*

Total arc length minimization implicitly contributes to a vertically compact drawing, and has horizontal effects as well, as it minimizes the number of dummy vertices being introduced whenever an arc spans across a layer. These are necessary to perform the subsequent steps and also help to measure the width of a layering more accurately as is further discussed below. For the final drawing, they are removed again. A natural generalization of DLP to arbitrary (i.e., not necessarily acyclic) graphs is to consider a layering L feasible if $L(v) \neq L(u)$ holds for all $uv \in A$[1]. Rüegg et al. [11, 12] introduce weights ω_{len} and ω_{rev} that allow to express the respective emphasis on the minimization of the total arc length and the number of reversed arcs. As opposed to DLP, the resulting multi-objective optimization problem is $\mathcal{NP}$-hard, and it remains so even if one of ω_{len} and ω_{rev} is zero [11].

Problem 2 *Generalized Layering Problem (GLP). Let $G = (V, A)$ be a directed graph. Find a feasible layering L of G that minimizes the expression*

$$\omega_{len} \left(\textstyle\sum_{uv \in A} |L(v) - L(u)| \right) \; + \; \omega_{rev} \, |\{uv \in A \mid L(v) < L(u)\}|.$$

The next step on the path to a two-dimensionally 'compact' drawing is to explicitly minimize its width. To be precise, one needs to distinguish the *estimated width* of a layering, which is given by the maximal number (or, if these are not uniform, the maximal accumulated widths) of original and dummy vertices in any of its layers, from the *final drawing width* which is further influenced by the horizontal coordinate assignment and arc routing [12]. Here, we restrict attention to the estimated width and denote it as $\mathcal{W}$. Integrating width minimization into DLP (subsequently denoted as problem DLP-W) is, apart from the consideration of dummy vertices, equivalent to solving the precedence-constrained multiprocessor scheduling problem and thus as well $\mathcal{NP}$-hard [15]. Healy and Nikolov [5] proposed a MIP formulation for DLP-W, and Jabrayilov et al. [6] presented one that integrates width minimization into GLP, i.e., that solves the following problem:

Problem 3 *Minimum Width Generalized Layering Problem (GLP-W)*[2]*. Find a feasible layering L of a directed graph $G = (V, A)$ that minimizes the expression*

$$\omega_{len} \left(\textstyle\sum_{uv \in A} |L(v) - L(u)| \right) \; + \; \omega_{rev} \, |\{uv \in A \mid L(v) < L(u)\}| \; + \; \omega_{wid} \, \mathcal{W}.$$

[1] This naturally extends to undirected graphs considered as bidirected ones.

[2] This problem is called *Compact Generalized Layering Problem (CGLP)* in [6, 13] but renamed here to harmonize with the other variants.

Moreover, in [13], Rüegg et al. propose to optimize a layering w.r.t. a target drawing area of width r_W and height r_H. Informally, a 'best' layering is then considered one that can be scaled up the most until it exhausts one of the two dimensions while still respecting the other. Formally, define $\mathcal{H} := \max_{v \in V} L(v)$ to be the height of a layering L. This definition is suitable as we may assume w.l.o.g. that vertices are assigned to consecutive layers starting from index one. The *scaling factor* $\mathcal{S}$ to be maximized is then the minimum of the ratios between the targeted and the actually used width and height, respectively, i.e., $\mathcal{S} = \min\{\frac{r_W}{\mathcal{W}}, \frac{r_H}{\mathcal{H}}\}$, and the resulting problem is:

Problem 4 *Maximum Scale Generalized Layering Problem (GLP-MS). Given a drawing area of (normalized[3]) width r_W and (normalized) height r_H, find a feasible layering L of a directed graph $G = (V, A)$ that minimizes the expression*

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

Clearly, each of the problems 2–4 generalizes over its predecessors. Hence, there is a 'backward compatibility' in the sense that MIPs for GLP can solve DLP (by setting $\omega_{rev} = \infty$), MIPs capable to minimize the width may also ignore it (by setting $\omega_{wid} = 0$), and MIPs to maximize the scaling factor can be used to minimize the width (by setting $r_W = 1$ and $r_H = \infty$). For DLP, there are two further inexact but well-known algorithms: First, the longest-path method by Eades and Sugiyama [3]. Here, a minimum total arc length is not guaranteed, but in a certain sense approximated, as the resulting height is minimal, and each vertex is placed on its lowest possible layer. Second, for a fixed width upper bound $\mathcal{W}_{max}$ (w.r.t. original but not dummy vertices), the Coffman-Graham algorithm [1], that is originally destined to the multiprocessor scheduling problem, allows for a $2 - \frac{2}{\mathcal{W}_{max}}$-approximation of the minimum height. An iterative scheme based on this algorithm, and to obtain drawings with an aspect ratio close to a desired value, was presented by Nachmanson et al. [9]. Moreover, Nikolov et al. approach DLP-W heuristically in [10], and Rüegg et al. tackle GLP-MS heuristically in [13].

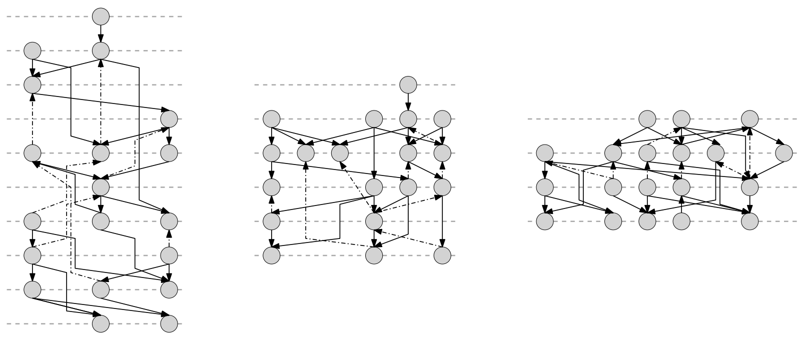


Figure 2: A directed graph drawn with a layering created by solving GLP-MS having $r_W : r_H$ set to $1:2$ (left), $1:1$ (middle), and $2:1$ (right). The obtained optimal $\mathcal{W} : \mathcal{H}$ combinations are respectively $5:10$, $6:6$, and $8:4$.

[3]Normalization is addressed in Sect. 3.

3 GLP-MS and a Slight Variation GLP-MS*

GLP-MS is 'different' than its preceding problems in Sect. 2 in several ways. First of all, the parameters r_W and r_H used to characterize the target drawing area introduce undesired economies of scale since different values representing the same aspect ratio lead to different numeric maxima of $\mathcal{S}$. At the same time, it is not necessary to specify the dimensions of the drawing area using *absolute* values as the goal is a best effort layering for the *relative* aspect ratio of the targeted area. If that exceeds, e.g., the absolute available width, this means that 'zooming out below 100%' is necessary (and inevitable) to fully display the graph at once on the target area. As a normalization, we thus propose to normalize r_W and r_H to $\frac{r_W}{\min\{r_W,r_H\}}$ and $\frac{r_H}{\min\{r_W,r_H\}}$, respectively. Fig. 2 indicates how different (normalized) choices of r_W and r_H affect optimal layerings of an example graph.

Moreover, the different role of the height $\mathcal{H}$, being now (just like and concurrently with the width $\mathcal{W}$) a variable subject to optimization, makes it particularly difficult to design a MIP model for GLP-MS. More precisely, GLP-MS asks for a maximization of $\omega_{scl}\ S$ that is expressed as a minimization of its negation in Problem 4. However, translated into a MIP, the definition of $\mathcal{S}$ to be the minimum of $\frac{r_W}{\mathcal{W}}$ and $\frac{r_H}{\mathcal{H}}$ inevitably leads to inequalities of the form $\mathcal{S} \cdot \mathcal{W} \leq r_W$ and $\mathcal{S} \cdot \mathcal{H} \leq r_H$ - which induce two products of integer variables with a potentially large value range. It is possible to linearize these, but the resulting formulation is unlikely to be well-solvable in practice. Rüegg et al. circumvent this problem elegantly in [13]. Instead of $-\omega_{scl}\ \mathcal{S}$, they minimize $\omega_{scl}\ \bar{\mathcal{S}}$ where $\bar{\mathcal{S}} := \frac{1}{\mathcal{S}}$. The above inequalities then turn into $\mathcal{W} \leq r_W \bar{\mathcal{S}}$ and $\mathcal{H} \leq r_H \bar{\mathcal{S}}$, i.e., linear ones. We call this slight variation GLP-MS*.

4 A Reference MIP for GLP-MS* and GLP-W

In [6], a MIP formulation, called CGL-W henceforth, was experimentally shown to be the superior one to solve problem GLP-W to optimality on a common benchmark set. In [13], Rüegg et al. modified it to obtain a (and the, to the best of our knowledge, yet only) MIP model for GLP-MS* and that we refer to as CGL-MS*. The following is a slightly simplified and more compact reformulation of it.

Let Y be a height upper bound to be chosen a priori. Then, for all $v \in V$ and all layers $k \in \{1, \ldots, Y-1\}$, CGL-MS* involves binary variables $y_{k,v}$ being equal to 1 if and only if $k < L(v)$. In particular $L(v) = 1$ if and only if $y_{k,v} = 0$, $L(v) = Y$ if and only if $y_{k,v} = 1$ for all $k \in \{1, \ldots, Y-1\}$, and $L(v) = k$ if and only if $y_{k-1,v} - y_{k,v} = 1$. There are further auxiliary variables r_{uv} (equal to 1 if and only if $uv \in A$ is reversed) and $d_{uv,k}$ (equal to 1 if $uv \in A$ causes a dummy vertex on layer $k \in \{2, \ldots, Y-1\}$), and the variable $\bar{\mathcal{S}}$ to represent the inverse scaling factor. The total number of variables amounts to $(|V|+|A|) \cdot (Y-1)+1$.

$$
\begin{array}{llll}
\text{minimize} \sum\limits_{uv \in A} \big(\omega_{rev}\, r_{uv} + \omega_{len}(1 + \sum\limits_{k=2}^{Y-1} d_{uv,k})\big) + \omega_{scl}\, \bar{\mathcal{S}} & & & \\
\text{subject to:} & & & \\
y_{k,v} - y_{k-1,v} & \leq 0 & \forall\, v \in V; k \in \{2,\ldots,Y-1\} & (1) \\
y_{1,u} - r_{uv} & \geq 0 & \forall\, uv \in A & (2) \\
y_{1,v} + r_{uv} & \geq 1 & \forall\, uv \in A & (3) \\
y_{k-1,u} - y_{k,v} - r_{uv} & \leq 0 & \forall\, uv \in A; k \in \{2,\ldots,Y-1\} & (4) \\
y_{k-1,v} - y_{k,u} + r_{uv} & \leq 1 & \forall\, uv \in A; k \in \{2,\ldots,Y-1\} & (5) \\
y_{Y-1,u} - r_{uv} & \leq 0 & \forall\, uv \in A & (6) \\
y_{Y-1,v} + r_{uv} & \leq 1 & \forall\, uv \in A & (7) \\
y_{k,u} - y_{k-1,v} - d_{uv,k} & \leq 0 & \forall\, uv \in A; k \in \{2,\ldots,Y-1\} & (8) \\
y_{k,v} - y_{k-1,u} - d_{uv,k} & \leq 0 & \forall\, uv \in A; k \in \{2,\ldots,Y-1\} & (9) \\
\sum\limits_{v \in V}(1 - y_{1,v}) & \leq r_W\, \bar{\mathcal{S}} & & (10) \\
\sum\limits_{v \in V} y_{Y-1,v} & \leq r_W\, \bar{\mathcal{S}} & & (11) \\
\sum\limits_{v \in V}(y_{k-1,v} - y_{k,v}) + \sum\limits_{uv \in A} d_{uv,k} & \leq r_W\, \bar{\mathcal{S}} & \forall\, k \in \{2,\ldots,Y-1\} & (12) \\
1 + \sum\limits_{k \in \{1,\ldots,Y-1\}} y_{k,v} & \leq r_H\, \bar{\mathcal{S}} & \forall\, v \in V & (13) \\
y_{k,v} & \in \{0,1\} & \forall\, v \in V; k \in \{1,\ldots,Y-1\} & \\
r_{uv} & \in [0,1] & \forall\, uv \in A & \\
d_{uv,k} & \in [0,1] & \forall\, uv \in A; k \in \{2,\ldots,Y-1\} & \\
\bar{\mathcal{S}} & \in \mathbb{R}_{\geq 0} & &
\end{array}
$$

To model the objective function, it is exploited that the length of an arc (i.e., the difference of the layer indices its endpoints are assigned to) is equivalent to the number of dummy vertices it causes plus one. Inequalities (1) establish transitivity in the sense that $L(v) > k$ implies $L(v) > k-1$ for available layers $k > 1$. Constraints (2) and (3) make sure that it is infeasible to assign both vertices of any arc to layer 1. More precisely, they imply $L(v) > 1$ if the arc $uv \in A$ is decided to have its original direction ($r_{uv} = 0$) or that $L(u) > 1$ if it is reversed, respectively. A consistent identification of reversed arcs based on the assignment of vertices to the other layers is then established by constraints (4)–(7). More precisely, inequalities (4) state that if $L(u) \geq k$, then $L(v) > k$ or the arc $uv \in A$ is reversed (or both, which is then prohibited by inequalities (5)). Rather unintuitively, inequalities (4) as well imply the condition $L(u) \neq L(v)$ if $r_{uv} = 0$, since there must be some layer k such that $L(u) \geq k$, and the constraint then enforces $L(v) > k$. Similarly, inequalities (5) state on the one hand that if $L(v) \geq k$ and the arc $uv \in A$ is reversed, then $L(u) > k$. On the other, since there must be some layer k such that $L(v) \geq k$, they implement the condition $L(u) \neq L(v)$ if $r_{uv} = 1$, as then $L(u) > k$ is enforced. Constraints (6) and (7) simply represent (4) and (5) for the case $k = Y$ (as the variables $y_{Y,v}$ do not exist). Each arc $uv \in A$ causes a dummy vertex on layer k if either $L(u) > k$ and $L(v) < k$ or vice versa. In the first case, inequalities (8) will force $d_{uv,k}$ to be 1, in the second case, inequalities (9) will do so. In any other case, an optimum solution has $d_{uv,k} = 0$ whenever $\omega_{len} > 0$. The maximal layer width and the height are related to the inverse maximum scaling factor $\bar{\mathcal{S}}$, in the way explained in Sect. 3, by inequalities (10)–(12) and (13), respectively. The total number of constraints is exactly $(4|A|+|V|+1)\cdot(Y-1)+1$. Finally, to reconvert the presented formulation to CGL-W, it is sufficient to replace the variable $\bar{\mathcal{S}}$ (and its weight ω_{scl}) by $\mathcal{W}$ (and ω_{wid}), the occurrences of $r_W\bar{\mathcal{S}}$ in inequalities (10)–(12) by $\mathcal{W}$, and to remove inequalities (13).

5 Neq Quadratic Assignment Formulations QLA-MS* and QLA-W

All the layering problems defined in Sect. 2 may be seen as specialized (quadratic) assignment problems. In this sense, the seminal integer program for DLP-W by Healy and Nikolov [5] is natural in that it uses binary variables $x_{v,k}$ taking on value 1 if and only if vertex $v \in V$ is assigned to layer $k \in \{1, \ldots, Y\}$. However, like CGL-W and CGL-MS*, its extension to solve GLP-W in [6] requires additional variables in order to model dummy vertices and arc reversals. In contrast to that, the following new model QLA-MS* involves exceptionally assignment variables and products of these. More precisely, for all $uv \in A$ and $k, l \in \{1 \ldots, Y\}$, the variable $p_{u,k,v,l}$ models the product $x_{u,k}\ x_{v,l}$. The total number of variables is thus $|V| \cdot Y + |A| \cdot Y^2 + 1$, i.e., larger than in case of CGL-MS*. However, we will require less constraints and are able to express these as well as the objective function very intuitively.

For instance, an arc $uv \in A$ is reversed if and only if $L(u) > L(v)$, i.e, if

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

Its length (again, the difference of the layer indices u and v are assigned to) is given by the expression

$$\sum_{k=2}^{Y} \sum_{l=1}^{k-1} (k-l) \cdot (p_{u,l,v,k} + p_{u,k,v,l})$$

since we know that exactly one of the products summarized over takes on value one. Moreover, the arc causes a dummy vertex on a layer $k \in \{2, \ldots, Y-1\}$ if and only if k is between the layers u and v are assigned to, i.e., if

$$\sum_{l=1}^{k-1} \sum_{m=k+1}^{Y} (p_{u,l,v,m} + p_{u,m,v,l}) = 1.$$

Finally, the condition that u and v must be assigned to different layers is expressed by simply fixing $p_{u,k,v,k}$ to zero for all $k \in \{1, \ldots, Y\}$. In practice, they can of course be just omitted instead. The full model QLA-MS* is the following:

$$
\begin{aligned}
&\text{minimize} \sum_{uv \in A} \Big(\sum_{k=2}^{Y} \sum_{l=1}^{k-1} \omega_{len}(k-l)(p_{u,l,v,k} + p_{u,k,v,l}) + \omega_{rev}\, p_{u,k,v,l} \Big) + \omega_{scl}\, \bar{\mathcal{S}} \\
&\text{subject to:} \\
&\sum_{k=1}^{Y} x_{v,k} && = 1 && \forall\, v \in V && (14) \\
&\sum_{l=1}^{Y} p_{u,k,v,l} && = x_{u,k} && \forall\, uv \in A; k \in \{1, \dots, Y\} && (15) \\
&\sum_{k=1}^{Y} p_{u,k,v,l} && = x_{v,l} && \forall\, uv \in A; l \in \{1, \dots, Y\} && (16) \\
&p_{u,k,v,k} && = 0 && \forall\, uv \in A; k \in \{1, \dots, Y\} && (17) \\
&\sum_{v \in V} x_{v,k} && \leq r_W\, \bar{\mathcal{S}} && \forall\, k \in \{1, Y\} && (18) \\
&\sum_{uv \in A} \sum_{l=1}^{k-1} \sum_{m=k+1}^{Y} (p_{u,l,v,m} + p_{u,m,v,l}) + \sum_{v \in V} x_{v,k} && \leq r_W\, \bar{\mathcal{S}} && \forall\, k \in \{2, \dots, Y-1\} && (19) \\
&\sum_{k=1}^{Y} k\, x_{v,k} && \leq r_H\, \bar{\mathcal{S}} && \forall\, v \in V && (20) \\
&x_{v,k} && \in \{0, 1\} && \forall\, v \in V; k \in \{1, \dots, Y\} \\
&p_{u,k,v,l} && \in [0, 1] && \forall\, uv \in A; k, l \in \{1, \dots, Y\} \\
&\bar{\mathcal{S}} && \in \mathbb{R}_{\geq 0}
\end{aligned}
$$

Equations (14) let each vertex be assigned a unique layer, and equations (15) and (16) establish that variable $p_{u,k,v,l}$ correctly represents the value of $x_{u,k} \cdot x_{v,l}$ for $uv \in A$, and $k, l \in \{1, \dots, Y\}$ by following the compact linearization approach [7, 8]. An intuitive interpretation for (15) is, that whenever $x_{u,k}$ is zero, all products involving it need to be zero as well. Conversely, if $x_{u,k} = 1$, then exactly one of the products on the left hand side (which are *all* the products of $x_{u,k}$ with different $x_{v,l}$ for some fixed vertex $v \in V$) must be equal to one as well due to (14). Equations (15) establish the same relationship for the second factor of any product variable. The width and height are related to $\bar{\mathcal{S}}$ by inequalities (18)–(20) in a similar fashion as in model CGL-MS*. Without accounting for the variable fixings (17), the total number of constraints is $(2|A| + 1) \cdot Y + 2|V|$ and thus about halved compared to model CGL-MS* as well as less sensitive to a higher graph density. Finally, QLA-W is obtained from the displayed model by omitting (20) and r_W, and by replacing $\bar{\mathcal{S}}$ by $\mathcal{W}$ and ω_{scl} by ω_{wid}.

6 Remarks on the Height and Alternative Expressions to Measure it

In both presented MIP models, the height is measured as the maximum layer index used plus one without loss of generality as discussed in Sect. 2. In practice, there may however exist optimal solutions to these MIPs that do not assign any vertex to layer one or that do not place vertices on consecutive layers only. But, this may happen only if w.r.t. the drawing area, the height (respectively the inequalities (13) or (20)) is (are) dominated by the width (respectively constraints (10)–(12) or (18)–(19)) due to the choice of r_W, r_H, and Y for the particular graph instance under consideration. Thus, in this case, the computed optimal layering and its height measure can be normalized a posteriori by simply ignoring empty layers.

Moreover, both inequalities (13) in CGL-MS* and (20) in QLA-MS* express the height of the layering in a way that emphasizes compactness and supports bounding $\bar{\mathcal{S}}$ from below. There are however alternative expressions and the respective choice may have an impact on the sustained performance when solving the respective MIP in practice. First of all, the height can as well be computed by taking the maximum over single (instead of summarized) layering variables. In CGL-MS*, this amounts to the following $|V|(Y-1)$ constraints as a replacement for (13):

$$(k+1)\, y_{k,v} \quad \leq r_H\, \bar{\mathcal{S}} \qquad \forall\, v \in V; k \in \{1, \ldots, Y-1\} \tag{21}$$

Similarly, in QLA-MS*, one might use the following $|V|Y$ constraints as a substitute for (20):

$$k\, x_{v,k} \quad \leq r_H\, \bar{\mathcal{S}} \qquad \forall\, v \in V; k \in \{1, \ldots, Y\} \tag{22}$$

Another alternative is proposed as part of the initial version of CGL-MS* by Rüegg et al. in [13]. It involves the addition of two auxiliary vertices s and t connected by an auxiliary arc (s,t) and the replacement of (13) by the following value assignments and constraints:

$$y_{k,s} = 0 \qquad \forall\, k \in \{1, \ldots, Y-1\} \tag{23}$$

$$y_{k,v} - y_{k,t} \leq 0 \qquad \forall\, v \in V; k \in \{1, \ldots, Y-1\} \tag{24}$$

$$1 + \sum_{k \in \{1,\ldots,Y-1\}} y_{k,t} \leq r_H\, \bar{\mathcal{S}} \tag{25}$$

Since all variables related to s are fixed and r_{st} can be fixed to zero as well, the extension causes only $3(Y-2)$ additional other constraints (1)–(13).

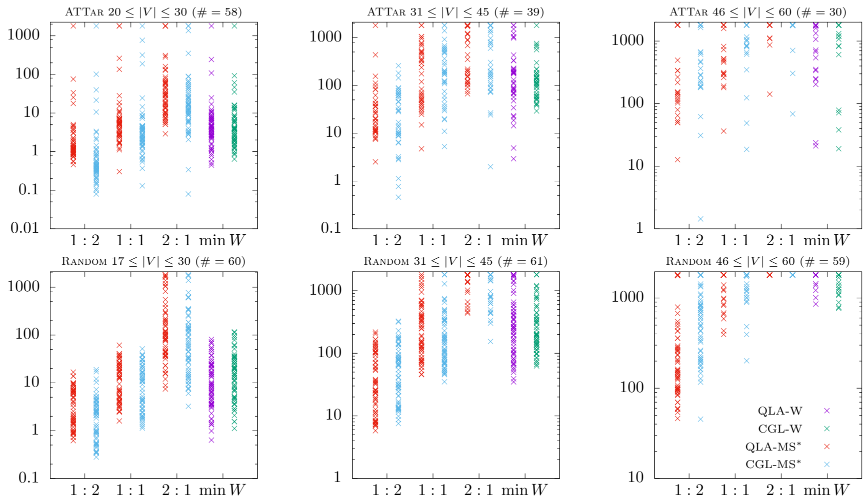
For the experimental evaluation that follows next, we implemented CGL-MS* using the approach described last, and QLA-MS* using (22), as these variants performed best on average.

7 Experimental Evaluation

Concerning aesthetic drawing criteria, the effects of integrating GLP-MS and GLP-W into the hierarchical framework by Sugiyama et al. have been evaluated already in [6, 12, 13]. We thus confine ourselves to show that QLA-MS* and QLA-W compare well to CGL-MS* and CGL-W in terms of solution times based on the same two instance sets as in these references. The first set ATTAR are the AT&T graphs from [2] whereof we extracted all non-tree instances with 20 to 60 vertices. These have 20 to 168 arcs, and density $\frac{|A|}{|V|}$ within $[1, 4.72]$ (on average 1.47). The second set RANDOM consists of 180 randomly generated and also acyclic and non-tree graphs with 17 to 60 vertices, 30 to 91 arcs, and about 1.5 arcs per vertex. The experiments cover the $r_H : r_W$ combinations $1:2$, $1:1$, and $2:1$ for QLA-MS* and CGL-MS*, and an additional run for QLA-W and CGL-W. The objective weights[4] were set to $\omega_{len} = 1$, $\omega_{rev} = Y\omega_{len}$,

[4] These are different than in [6, 13] and explain the degraded performance of CGL-W compared to [6] where $\omega_{wid} = 1$. QLA-W performs on equal terms also in this case.

and $\omega_{wid} = \omega_{scl} = |A|\omega_{rev} + |A| \cdot Y + 1$. Priority is thus laid on a maximum scaling factor or minimum width, and an arc is reversed only if this supports the respective main goal. Moreover, we set $Y = |V|$ in case of GLP-MS* like in [13] and $Y = \lceil 1.6 \cdot \sqrt{|V|} \rceil$ for GLP-W like in [6]. All MIPs were solved using Gurobi 8[5] single-threadedly on a Debian Linux system with an Intel Core i7-3770T CPU (2.5 GHz) and 8 GB RAM, and with a time limit set to half an hour.



ATTAR $\vert V\vert$	#	QLA-MS* 1:2	1:1	2:1	CGL-MS* 1:2	1:1	2:1	QLA-W min W	CGL-W min W
20–30	58	1	2	5	1	1	3	1	1
31–45	39	1	2	14	-	-	10	2	1
46–60	30	11	14	26	11	16	25	18	19

RANDOM $\vert V\vert$	#	QLA-MS* 1:2	1:1	2:1	CGL-MS* 1:2	1:1	2:1	QLA-W min W	CGL-W min W
17–30	60	-	-	4	-	-	2	-	-
31–45	61	-	1	50	-	4	41	9	4
46–60	59	6	47	59	12	45	59	54	48

Figure 3: The plots depict the solution times in seconds (a cross per graph) for the different $r_H : r_W$ combinations as well as pure width minimization (min W). Crosses at the top (1800s) correspond to timeouts, their quantities are listed in the tables below.

Averaged over the whole experiment, QLA-MS* and CGL-MS* perform almost on equal terms. As is visualized in Fig. 3, the latter is faster and causes less timeouts in the 2 : 1-case, but in the 1 : 2-experiments often the opposite is true, especially for the RANDOM and larger ATTAR instances. In the 1 : 1-experiments, their relative performance is very similar with CGL-MS* being a bit faster on those instances solved but observing slightly less timeouts with QLA-MS*. The 2 : 1-case, where the height is constrained to be *at most* twice the width, turns out to be the hardest - which is not surprising as e.g. the ATTAR graphs are *at least* twice as high as wide when drawn with the classic framework [6]. Minimizing the width w.r.t. a fixed (but larger) height limit appears to be considerably easier compared to the implicit 'width emphasis' in the 2 : 1-case that additionally involves finding the best $\mathcal{H}$:$\mathcal{W}$-pair under the

[5] A proprietary MIP solver, see www.gurobi.com

given aspect ratio constraint. In this discipline, QLA-W often performs slightly faster if an instance can be solved, but causes more timeouts than CGL-W on the RANDOM ones.

8 Conclusion

In this paper, we have drawn a series of subsequent generalizations of the classical directed graph layering problem. We studied in more detail the most general one among them, where the aim is to best possibly fit a layered drawing to a given area under further minimization of arc lengths and the number of reversed arcs. Besides discussing practical issues when modeling this problem as a mixed-integer program, we presented a slightly simplified version of the (to the best of our knowledge) so far only existing such formulation, and then proposed a new and much more intuitive one that expresses the problem in terms of a quadratic assignment problem. Further, we showed that using it, a performance comparable to the previous one can be sustained in practice. Both modeling concepts can as well be used to solve the slightly more special problem variant where the drawing area is not fully specified, but the width shall be minimized w.r.t. a given height restriction. We hope that the drawn links to the quadratic assignment problem inspire further and also heuristic solution approaches.

References

[1] Edward G. Coffman and Ronald L. Graham. Optimal scheduling for two-processor systems. *Acta Informatica*, 1(3):200–213, 1972.

[2] Giuseppe Di Battista, Ashim Garg, Giuseppe Liotta, Armando Parise, Roberto Tamassia, Emanuele Tassinari, Francesco Vargiu, and Luca Vismara. Drawing directed acyclic graphs: An experimental study. In *Proceedings of the Symposium on Graph Drawing (GD'96)*, volume 1190 of *LNCS*, pages 76–91. Springer, 1997.

[3] Peter Eades and Kozo Sugiyama. How to draw a directed graph. *Journal of Information Processing*, 13(4):424–437, 1990.

[4] Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Kiem-Phong Vo. A technique for drawing directed graphs. *Software Engineering*, 19(3):214–230, 1993.

[5] Patrick Healy and Nikola S. Nikolov. A branch-and-cut approach to the directed acyclic graph layering problem. In Michael T. Goodrich and Stephen G. Kobourov, editors, *Proceedings of the 10th International Symposium on Graph Drawing (GD'02)*, volume 2528 of *LNCS*, pages 98–109. Springer, 2002.

[6] Adalat Jabrayilov, Sven Mallach, Petra Mutzel, Ulf Rüegg, and Reinhard von Hanxleden. Compact layered drawings of general directed graphs. In Yifan Hu and Martin Nöllenburg, editors, *Proceedings of the 24th International Symposium on Graph Drawing and Network Visualization (GD'16)*, pages 209–221, Cham, 2016. Springer.

[7] Sven Mallach. Compact linearization for binary quadratic problems subject to assignment constraints. *4OR*, pages 1–15, Dec 2017.

[8] Sven Mallach. Compact linearization for binary quadratic problems subject to linear equations. *CoRR*, abs/1712.05872, 2017.

[9] Lev Nachmanson, George Robertson, and Bongshin Lee. Drawing graphs with GLEE. In Seok-Hee Hong, Takao Nishizeki, and Wu Quan, editors, *Proceedings of the 15th International Symposium on Graph Drawing (GD'07)*, volume 4875 of *LNCS*, pages 389–394. Springer, 2008.

[10] Nikola S. Nikolov, Alexandre Tarassov, and Jürgen Branke. In search for efficient heuristics for minimum-width graph layering with consideration of dummy nodes. *Journal of Experimental Algorithmics*, 10, 2005.

[11] Ulf Rüegg, Thorsten Ehlers, Miro Spönemann, and Reinhard von Hanxleden. A generalization of the directed graph layering problem. Technical Report 1501, Kiel University, Department of Computer Science, February 2015. ISSN 2192-6247.

[12] Ulf Rüegg, Thorsten Ehlers, Miro Spönemann, and Reinhard von Hanxleden. A generalization of the directed graph layering problem. In Yifan Hu and Martin Nöllenburg, editors, *Proceedings of the 24th International Symposium on Graph Drawing and Network Visualization (GD'16)*, pages 196–208, Cham, 2016. Springer.

[13] Ulf Rüegg, Thorsten Ehlers, Miro Spönemann, and Reinhard von Hanxleden. Generalized layerings for arbitrary and fixed drawing areas. *Journal of Graph Algorithms and Applications*, 21(5):823–856, 2017.

[14] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, 11(2):109–125, February 1981.

[15] Jeffrey D. Ullman. NP-complete scheduling problems. *Journal of Computer and System Sciences*, 10(3):384–393, 1975.