Coordinate Assignment for Cyclic Level Graphs

Christian Bachmaier[1], Franz J. Brandenburg[1],
Wolfgang Brunner[1], and Raymund Fülöp[2]

[1] University of Passau, Germany,
{bachmaier|brandenb|brunner}@fim.uni-passau.de
[2] Technische Universität München, Germany, fueloep@in.tum.de

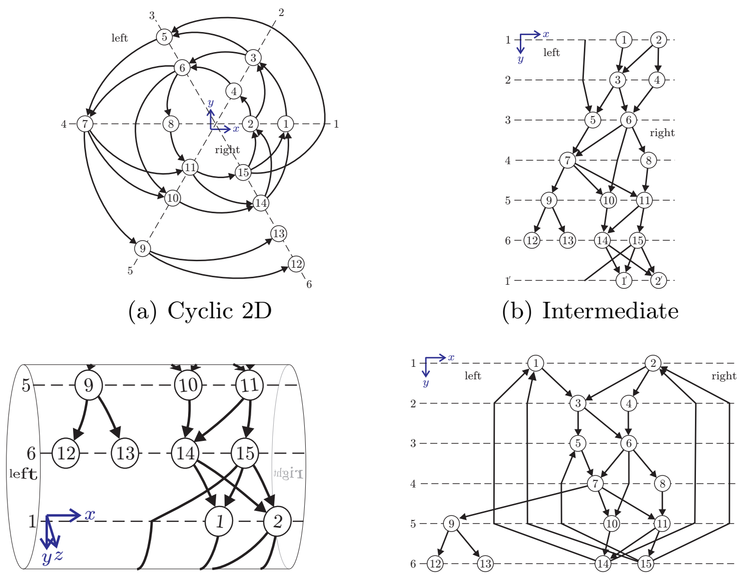
Abstract. The Sugiyama framework is the most commonly used concept for visualizing directed graphs. It draws them in a hierarchical way and operates in four phases: cycle removal, leveling, crossing reduction, and coordinate assignment. However, there are situations where cycles must be displayed as such, e. g., distinguished cycles in the biosciences and scheduling processes which repeat in a daily or weekly turn. This excludes the removal of cycles. In their seminal paper Sugiyama et al. introduced recurrent hierarchies as a concept to draw graphs with cycles. However, this concept has not received much attention in the following years. In this paper we supplement our cyclic Sugiyama framework and investigate the coordinate assignment phase. We provide an algorithm which runs in linear time and constructs drawings which have at most two bends per edge and use quadratic area.

1 Introduction

The Sugiyama framework [9] is among the most intensively studied algorithms in graph drawing. It is the standard technique to draw directed graphs, and displays them in a hierarchical manner. It consists of the four phases of cycle removal, leveling, crossing reduction, and coordinate assignment. Typical applications are schedules, UML diagrams, and flow charts.

In its first phase the Sugiyama framework destroys all cycles. However, there are many situations where this is unacceptable. There are well-known cycles in the biosciences [7], where it is a common standard to display these cycles as such. Another inevitable use are repeating processes, such as daily, weekly, or monthly schedules which define the Periodic Event Scheduling Problem [8].

In their seminal paper [9], Sugiyama et al. proposed a solution for both the hierarchic and the cyclic style. The latter is called a *recurrent hierarchy* which is a level graph with additional edges from the last to the first level. It can be drawn in 2D where the levels are rays from a common center (see Fig. 1(a)) and each edge $e = (u, v)$ is a monotone counterclockwise poly-spiral segment from u to v wrapping around the center at most once. An alternative is a 3D drawing on a cylinder (see Fig. 1(c)). A combination would be the best of both worlds: an interactive 2D view with horizontal levels. It can be scrolled upwards and downwards infinitely and always shows a different part of the cylinder, see Fig. 1(b) for a snap shot, which also represents our intermediate drawing.



(a) Cyclic 2D (b) Intermediate

(c) Cyclic 3D (d) Hierarchic

Fig. 1. Example drawings

In cyclic drawings edges are irreversible and cycles are represented in a direct way. Thus, the cycle removal phase is not needed. This saves much effort, since the underlying feedback arc set problem is $\mathcal{NP}$-hard [4]. Further advantages over hierarchic drawings (see Fig. 1(d)) are shorter edges and fewer crossings.

A planar recurrent hierarchy is shown on the cover of the textbook by Kaufmann and Wagner [6]. There it is stated that recurrent hierarchies are "unfortunately [...] still not well studied". After investigating the leveling phase [1], we consider the coordinate assignment phase for the cyclic case. There are several algorithms for non-cyclic coordinate assignment [6]. We modify the established algorithm of Brandes and Köpf [3] for cyclic level graphs and provide a linear time algorithm using quadratic area and with at most two bends per edge.

2 Preliminaries

A *cyclic k-level graph* $G = (V, E, \phi)$ $(k \geq 2)$ is a directed graph without self-loops with a given surjective level assignment of the vertices $\phi\colon V \to \{1, 2, \ldots, k\}$. Let $V_i \subset V$ be the set of vertices v with $\phi(v) = i$. For two vertices $u, v \in V$ let $\mathrm{span}(u, v) := \phi(v) - \phi(u)$ if $\phi(u) < \phi(v)$ and $\mathrm{span}(u, v) := \phi(v) - \phi(u) + k$ otherwise. For an edge $e = (a, b) \in E$ we define $\mathrm{span}(e) := \mathrm{span}(a, b)$. An edge e with $\mathrm{span}(e) = 1$ is *short*, otherwise *long*. A graph is *proper* if all edges are short. Each cyclic level graph can be made proper by adding $\mathrm{span}(e) - 1$ dummy vertices for each edge e and thus splitting e in $\mathrm{span}(e)$ many short edges, which we call the *segments* of e. In total, this leads up to $\mathcal{O}(|E| \cdot k)$ new vertices. The first and the last segment of each edge are its *outer segments*, and all other segments between two dummy vertices are its *inner segments*. A proper cyclic k-level graph $G = (V, E, \phi, <)$ is *ordered* if $<$ is a total ordering for each V_i ($1 \leq i \leq k$). In accordance to [3] we say that in an ordered cyclic level graph there are two *conflicting* segments if they cross or share a vertex. Conflicts are of *type 0, 1* or *2*, if they are induced by 0, 1, or 2 inner segments, respectively.

We represent drawings of cyclic level graphs in an *intermediate drawing* in the remainder of the paper assigning each vertex v two coordinates $x(v) \in \mathbb{R}$ and $y(v) = \phi(v) \in \mathbb{N}$. The x-coordinate increases from *left* to *right*, the y-coordinate increases *downwards* in edge direction, see Fig. 1(b). All vertices on level 1 are duplicated on level $k+1$ using the same x-coordinates. Each segment $s = (u, v)$ is drawn straight-line from $\big(x(u), y(u)\big)$ to $\big(x(v), y(u)+1\big)$ with *slope* $\frac{1}{x(v)-x(u)}$. A *2D drawing* as in Fig. 1(a) is obtained from an intermediate drawing by transforming each point $p = (x(p), y(p))$ of the plane to $\big(x_{2\mathrm{D}}(p), y_{2\mathrm{D}}(p)\big) = \big(r(p) \cdot \cos(\alpha(p)), r(p) \cdot \sin(\alpha(p))\big)$, with the radius $r(p) = (\mathrm{offset}_x + \max_{v \in V}(x(v))) - x(p) \cdot \delta_x$ and the angle $\alpha(p) = (y(p) - 1) \cdot \frac{2\pi}{k}$. The constant offset_x defines the minimum distance of a vertex to the center and δ_x the minimum distance of vertices on the same level. A *3D drawing* as in Fig. 1(c) uses the coordinates $\big(x_{3\mathrm{D}}(p), y_{3\mathrm{D}}(p), z_{3\mathrm{D}}(p)\big) = \big(x(p) \cdot \delta_x, -r_k \cdot \sin(\alpha(p)), r_k \cdot \cos(\alpha(p))\big)$ where r_k is the radius of the cylinder. These equations transform straight lines of the intermediate drawing to spiral segments in the 2D or 3D drawings.

A drawing is *(cyclic level) plane* if the edges do not cross except on common endpoints. A cyclic k-level graph is *(cyclic level) planar* if such a drawing exists.

3 Layout Algorithm

In this section we describe our coordinate assignment phase for cyclic level graphs. We adapt the algorithm of Brandes and Köpf [3] and use their notation. Like them, we also assume that the crossing reduction has avoided type 2 conflicts. These can be avoided even for cyclic level graphs as shown recently [5].

The input to our algorithm is the output of the third phase and thus a proper ordered cyclic level graph. Note that dummy vertices were introduced after the leveling. Algorithm 1 consists of three basic steps: block building (lines 4–5), horizontal compaction (lines 6–12), and balancing (line 14) which reflect the steps in [3]. The first two steps are carried out four times (*runs*) for each combination of left/right with up/down alignment (line 2). The four results are merged by the balancing step. We describe the left top run only. The other three runs are realized by flipping the graph horizontally and/or vertically before and after (lines 3, 13) each run. The computed intermediate drawing can be transformed into the 2D or 3D drawing, where dummy vertices are replaced by edge bends.

In the cyclic case there may be unavoidable cyclic dependencies in the left-to-right ordering among vertically aligned paths. Thus, it is impossible to draw inner segments vertically, in general. We solve this problem by shearing the drawing of such a cycle s. t. all inner segments have the same slope.

Algorithm 1: cyclicCoordinateAssignment

Input: $G = (V, E, \phi, <)$: An ordered and proper cyclic k-level graph
Output: Coordinates $(x(v), y(v))$ for each $v \in V$ in the intermediate drawing $\mathcal{I}$

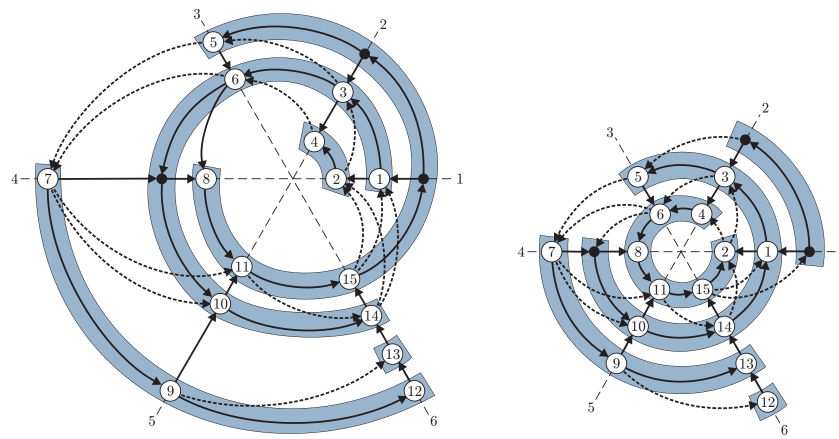
$\mathcal{P} \leftarrow \emptyset$
foreach $(h, v) \in \{\mathit{left}, \mathit{right}\} \times \{\mathit{up}, \mathit{down}\}$ **do**
  $G' \leftarrow \text{flip}(G, h, v)$ *// according to current run*
  $H \leftarrow \text{buildCyclicBlockGraph}(G')$
  splitLongBlocks(H) *// split long and closed blocks*
  $\mathcal{S} \leftarrow \text{computeSCCs}(H)$
  foreach *complex SCC* $S \in \mathcal{S}$ **do**
    $S' \leftarrow \text{cutSCC}(S)$ *// returns non-cyclic block graph*
    $\text{width}(S') \leftarrow \text{compact}(S')$ *// using two topsorts*
    $\text{shear}(S', -(\text{wind}(S') \cdot k) / \text{width}(S'))$ *// shear S' with given slope*
    $\mathcal{S} \leftarrow \mathcal{S} \setminus S \cup S'$
  compact($\mathcal{S}$) *// globally all SCCs*
  $\mathcal{P} \leftarrow \mathcal{P} \cup \text{flip}(\mathcal{S}, h, v)$
$\mathcal{I} \leftarrow \text{balance}(\mathcal{P})$ *// balance four runs*
return $\mathcal{I}$

3.1 Block Building

The block building phase is done in the same way as in [3]. We try to align vertices with its median adjacent vertices to blocks and remove all other segments level by level until we obtain a cyclic path graph and thus a cyclic block graph.

Definition 1. *A* cyclic path graph $H' = (V, E_{intra}, \phi, <)$ *is an ordered and proper cyclic level graph with a plane embedding respecting the ordering* $<$*. Each vertex of* H' *has indegree and outdegree at most one. We call each connected component of* H' *a* block *and all edges* $e \in E_{intra}$ intra block edges*. A block* B *is* closed *if each vertex of* B *has indegree and outdegree one or* open*, otherwise. The* height *of* B *is defined as the number of intra block edges in* B*. The* cyclic block graph $H = (V, E_{intra} \dot\cup E_{inter}, \phi)$ *of* H' *is obtained by adding an edge* $e \in E_{inter}$ *from each vertex in* H' *to its consecutive right vertex on the same level (if there is one), which we call* inter block edges*.*

To create such a graph we first mark outer segments involved in type 1 conflicts between two levels. Then, we traverse the lower level from left to right and try to align each vertex with one of its median predecessor vertices. First we try its upper left median, then its upper right median. An alignment is impossible if the segment is marked or if it would cross a segment already used for aligning. The current vertex becomes the top vertex of a new block if both alignments fail. All inner segments of an edge are aligned and thus lie in the same block. As each block is drawn with constant slope, this ensures at most two bends per edge. E_{intra} is the set of all remaining edges. See Fig. 2 for an example. Vertices and intra block edges of the same block are framed. The inter block edges lie on the level lines. The dotted segments were removed in the block building phase.



(a) Left upper run (b) Right upper run

Fig. 2. Block graphs of Fig. 1(a) to (c) as 2D drawings

The cyclic block graph can have closed blocks (with height k) and open blocks with height $\geq k$ (spirals) which shall be avoided to simplify the algorithm (line 5). In both cases we split such a block by removing outer segments until each resulting block has at most height $k-1$. Such outer segments always exist, as no edge can span more than k levels. Therefore, the invariant of at most two bends per edge still holds. Note that an originally closed block will not be sheared like other blocks in Sect. 3.2 as it cannot be part of a cyclic dependency. See Fig. 2(b) for an example: The segment $(2, 4)$ was removed to open a closed block and the segment $(5, 7)$ was used to split a long block in two shorter ones. The result is a cyclic block graph with open blocks of height at most $k - 1$.

3.2 Horizontal Compaction

In this section we compact the cyclic block graph by arranging all blocks as close to each other as possible minimizing the width of the drawing. Not all blocks can be drawn vertically as there can be cyclic dependencies in the left-to-right ordering among blocks, which we call rings.

Definition 2. *A* block path P *in a cyclic block graph* $H = (V, E_{intra} \dot\cup E_{inter}, \phi)$ *is a sequence of vertices* $v_1, \ldots, v_s \in V$ *s. t. for each pair of consecutive vertices* v_i *and* $v_{i+1}, 1 \leq i < s$, $(v_i, v_{i+1}) \in E_{intra}$ *or* $(v_{i+1}, v_i) \in E_{intra}$ *or* $(v_i, v_{i+1}) \in E_{inter}$. *It is* simple *if all vertices are mutually distinct. A block path is a* ring R *if* $v_1 = v_s$. *In a* simple ring *the vertices* $v_1, \ldots, v_{s-1}$ *are mutually distinct. The* width *of* R *is the number of inter block edges in* R. *Let* c_{down} *and* c_{up} *be the number of intra block edges traversed in* R *in and against their direction, respectively. The* number of windings *of* R *is then defined as* $\mathrm{wind}(R) = (c_{down} - c_{up})/k$.

Informally, a ring is a cycle in the block graph where the direction of the inter block edges is preserved and the intra block edges are used in any direction.

wind(R) counts how often R wraps around the center. As each ring is an ordered sequence, we count windings along increasing and decreasing levels positive and negative, respectively. We consider the strongly connected components (SCCs) connected by block paths of the block graph separately. The *simple SCCs* consist of one block. All other *complex SCCs* contain rings. Figure 2(a) consists of three simple SCCs ($(2, 4)$, $(7, 9, 12)$ and (13)) and one complex SCC (the remaining two blocks) in which all simple rings R have $\text{width}(R) = 2$ and $\text{wind}(R) = 1$.

Lemma 1. *For each ring R of a cyclic block graph G $\text{wind}(R) \neq 0$.*

Proof. Assume for contradiction that there exists a ring R with $\text{wind}(R) = 0$ in the block graph of G. Unwrapping G several times, i. e., placing multiple copies of the intermediate drawing one below the other and merging first and last levels, leads to a block graph of a (non-cyclic) level graph H which contains R completely. Then, H has a cyclic dependency, which is a contradiction. □

Lemma 2. *For each simple ring R of a cyclic block graph $|\text{wind}(R)| \leq 1$.*

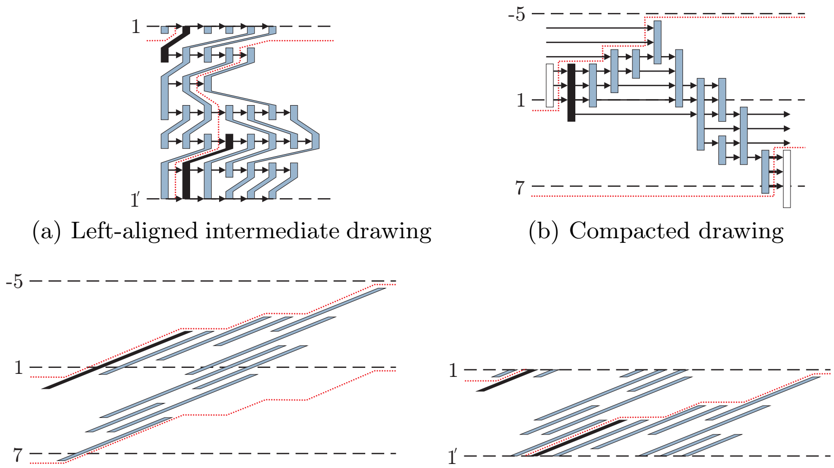
Proof. Assume for contradiction that there is a simple ring R with $|\text{wind}(R)| > 1$. As R wraps around the center in the 2D drawing more than once, each drawing of R crosses itself. Each cyclic path graph is planar. Further, each drawing of it respecting its ordering can be extended to a planar drawing of its cyclic block graph by adding the inter block edges along the level lines. Since R is a subgraph of the cyclic block graph, this is a contradiction. □

Theorem 1. *Let $\mathcal{R}$ be the set of all simple rings of an SCC in a cyclic block graph. For each $R \in \mathcal{R}$ $\text{wind}(R) = 1$ or for each $R \in \mathcal{R}$ $\text{wind}(R) = -1$.*

Proof. According to Lemmas 1 and 2 $|\text{wind}(R)| = 1$ holds for each ring $R \in \mathcal{R}$. Assume for contradiction that there exist two rings $R_1, R_2 \in \mathcal{R}$ with $\text{wind}(R_1) = 1$ and $\text{wind}(R_2) = -1$. Let v_1 and v_2 be vertices in R_1 and R_2, respectively. Let S be a (not necessarily simple) ring through v_1 and v_2, which always exists as v_1 and v_2 lie in the same SCC. Due to Lemma 1 $\text{wind}(S) \neq 0$. If $\text{wind}(S) > 0$, let T be a non-simple ring consisting of S and $\text{wind}(S)$ many copies of R_2 joined via v_2. Otherwise, let T be a ring consisting of S and $-\text{wind}(S)$ many copies of R_1 joined via v_1. In both cases $\text{wind}(T) = 0$, which contradicts Lemma 1. □

Definition 3. *Let S be a complex SCC of a cyclic block graph containing a simple ring R. We define* $\text{wind}(S) = \text{wind}(R)$ *and* $\text{width}(S)$ *as the maximum width of all simple rings in S.*

Horizontal Compaction of a complex Strongly Connected Component
It is not possible to draw all blocks of a complex SCC S straight-line and vertically. Therefore, we will draw all blocks of S with the same slope. The slope has to be chosen s. t. each ring, and thus the resulting curve, in S starts and ends at the same coordinates. All rings in S have the same number of windings wind(S), which is either 1 or -1. Thus, each simple ring spans $\text{wind}(S) \cdot k$ levels. To draw



(a) Left-aligned intermediate drawing (b) Compacted drawing

(c) Sheared drawing (d) Final intermediate drawing

Fig. 3. Drawing of a complex SCC

one simple ring R we could use the slope $-(\mathrm{wind}(R) \cdot k)/\,\mathrm{width}(R)$, which would result in inter block edges of unit length. In order to draw all blocks in S with the same slope (line 10 in Algorithm 1) we must use the width of the widest simple ring of S and the slope $-(\mathrm{wind}(S) \cdot k)/\,\mathrm{width}(S)$. With this slope the widest ring will fit exactly and use unit length inter block edges. All narrower rings will have some unused horizontal space in the drawing and thus have inter block edges which are longer than one unit.

To use this slope we have to determine the width of the widest simple ring in S. The general problem of finding a longest cycle is $\mathcal{NP}$-hard [4]. However, here it can be solved in linear time by cutting the SCC. Finding the length and compacting the layout is done simultaneously. To cut an SCC (line 8 in Algorithm 1), we start at an arbitrary block B and temporarily remove all incoming inter block edges of B. We then follow the outgoing inter block edge of the topmost vertex of B to the next block B'. We temporarily remove all incoming inter block edges of B' which are above the traversed incoming edge. We repeat this procedure until the topmost vertex of the current block has no outgoing inter block edge. Note that this happens before B is visited a second time, as otherwise the SCC would contain unreachable blocks. Thus, a block path P_r from B to a rightmost vertex in S is found. The same procedure is repeated from starting block B using the outgoing inter block edge of the lowest vertex to reach B' and deleting all incoming inter block edges below the traversed edge until a lowest vertex with no incoming inter block edge is found, which gives the block path P_l.

Combining P_r and P_l results in a y-monotone path P from a rightmost vertex through B to a leftmost vertex, using inter block edges in both directions and preserving intra block edge directions. Due to the removed incoming inter block edges left of P all rings in S are cut exactly once. We then assign an arbitrary vertex $v \in V$ of B the new coordinate $y'(v) = \phi(v)$. In a traversal of the block graph we assign each vertex a y'-coordinate: Using an inter block edge we assign both end vertices the same y'-coordinate. Using an intra block edge in or against its direction we increase or decrease the y'-coordinate by 1, respectively, without using a modulo operation. The result is an acyclic block graph, which we compact (in contrast to [3]) in the following way (line 9 in Algorithm 1): We place each block which is a source in the acyclic block graph on an imaginary zero line, treat all other blocks in the topological order and move them as much to the left as possible preserving unit distance. Afterwards, we fix all sinks on their positions, treat all other blocks against the topological order and move them as much to the right as possible. For placing a block as close as possible to the already placed ones, we traverse its levels. After the compaction each block (and therefore each vertex v in S) has an assigned x'-coordinate. Let $e = (u, v)$ be a removed inter block edge. The width of the widest simple ring of S through e is then $x'(u) - x'(v) + 1$. Considering all removed inter block edges and computing the maximum value gives the width of the widest simple ring $\mathrm{width}(S)$ in S.

See Fig. 3 for an example of an SCC S with $\mathrm{wind}(S) = 1$ and $k = 6$ levels. Figure 3(a) shows a leftmost intermediate drawing with the black start block B. Its lowest vertex is already the leftmost vertex on its level. To reach a rightmost vertex four other blocks have to be visited. The dashed line cuts six inter block edges. Figure 3(b) shows the resulting compacted horizontal block graph using y'-coordinates. The widths of the widest rings through each of the six cutted edges are (from top to bottom): 5, 5, 7, 10, 10, 10. Thus, $\mathrm{width}(S) = 10$. Shearing the drawing with slope $\frac{-1 \cdot 6}{10}$ results in Fig. 3(c). Using the modulo operation for the y-coordinates gives the final intermediate drawing in Fig. 3(d).

Theorem 2. *The intermediate drawing of S uses the coordinates $x(v) = x'(v) - (\mathrm{width}(S)/(\mathrm{wind}(S) \cdot k)) \cdot y'(v)$ and $y(v) = ((y'(v) - 1) \bmod k) + 1 = \phi(v)$ for each vertex v in S. In the drawing all intra block edges have $\mathrm{slope}(S) = -(\mathrm{wind}(S) \cdot k)/\,\mathrm{width}(S)$. The ordering of vertices on the same level is the one given by the crossing reduction phase and these vertices have at least unit distance.*

Proof. In the compacted drawing of the open block graph all blocks are drawn vertically. Shearing the drawing results in unchanged y'-coordinates and new x-coordinates $x(v) = x'(v) + y'(v)/\,\mathrm{slope}(S)$. Now all edges have slope $\mathrm{slope}(S)$. Using the y-coordinates $y(v) = ((y'(v) - 1) \mod k) + 1 = \phi(v)$ does not affect the slope of the edges. But now all vertices on the same level have the same y-coordinate again. Let u and v be two consecutive vertices on the same level. Let u be left of v according to the crossing reduction. If the inter block edge (u, v) was not cut before, then u and v have the same y'-coordinates in the compacted drawing and u is still the left neighbor of v with at least unit distance between them. This does not change in the sheared or final intermediate drawing. If (u, v) was cut, then $y'(v) = y'(u) - k \cdot \mathrm{wind}(S)$. There exists a simple block path P from v to u as we are compacting an SCC. P cannot have been cut as otherwise P and (u, v) form a simple ring that would have been cut twice. The ring formed by P and (u, v) is at most $\mathrm{width}(S)$ wide and thus $x'(v) \geq x'(u) - (\mathrm{width}(S) - 1)$.

After the drawing is sheared, $x(v) \geq x(u) + 1$ holds. Therefore, u is still left of v and the two vertices are at least unit distance apart. As a result, all consecutive vertices and thus all vertices on the same level are separated by unit distance and are in the ordering given by the crossing reduction phase. □

Horizontal Compaction Our next step is to globally compact the set of compacted complex SCCs and simple SCCs (line 12 in Algorithm 1).

Lemma 3. *In a drawing that respects the order of the crossing reduction phase all vertices of an SCC on the same level are consecutive.*

Proof. Let u, v be two vertices of an SCC on the same level s. t. u is left of v. Note that there is a block path from v to u. Also, there is a horizontal path from u to v using inter block edges only. Therefore, all vertices between u and v lie on a ring containing u and v and thus belong to the same SCC. □

This means that no SCCs can interleave. We interpret the SCCs as super vertices and perform a topological sorting on the resulting DAG. We then compact the SCCs as we compacted the blocks of a (non-cyclic) block graph before.

3.3 Balancing

In this phase (line 14 in Algorithm 1) the four results are balanced by computing one x-coordinate for each vertex out of the four x-coordinates computed by the four runs. The only difference to the algorithm of Brandes and Köpf [3] is that we do not use the average median of the four x-coordinates for each vertex, since this can induce additional bends in the cyclic case. The reason is that on lines with different slopes the median changes at crossings, i. e., it is a non-linear function. Hence, we use the average of all four x-coordinates for each vertex.

Proposition 1. *Using the average of the x-coordinates of the four runs for each vertex does not change the ordering of the vertices on a level and preserves at least unit distance. Additional bends can occur since the blocks of the four runs may differ. However, the invariant of at most two bends per edge e in the final drawing still holds, as the y-coordinates of the bends located at the topmost and lowest dummy vertex of e are identical in each drawing.*

Note that it is possible that some vertices in one run belong to a block of an SCC although they do not belong to an SCC or even one block in another run. Thus, balancing can lead to more different slopes than in each of the four runs alone. See Fig. 2 for two different block graphs of two runs of the same graph.

4 Algorithm Analysis

Theorem 3. *Let $G = (V, E, \phi, \prec)$ be a proper ordered cyclic k-level graph. The width of the intermediate drawing of G is $\mathcal{O}(|V|^2/k)$ and the area is $\mathcal{O}(|V|^2)$. For the 3D drawing the same bounds hold. The 2D drawing has a width and height of $\mathcal{O}(|V|^2/k)$ and thus an area of $\mathcal{O}(|V|^4/k^2)$.*

Proof. Let $\mathcal{S} = \{S_1, \ldots, S_r\}$ be the set of all SCCs. Let N_i be the number of (dummy) vertices in S_i. The width of the compacted drawing of S_i is $\text{width}(S_i) \leq N_i$. Shearing the drawing of height at most N_i with slope $-\text{wind}(S)\cdot k/\text{width}(S_i)$ adds at most N_i^2/k to the width. Thus, the width of the drawing is in $\mathcal{O}(N_i^2/k)$. The width of the drawing of G is at most the sum of the widths of the drawings of all SCCs and thus in $\mathcal{O}(|V|^2/k)$. As the height is k, the area is in $\mathcal{O}(|V|^2)$.

The height and width of the 2D drawing is twice the width of the intermediate drawing and thus the area is $\mathcal{O}(|V|^4/k^2)$, however. □

Note that the width of the drawing reduces to $\mathcal{O}(|V|)$ if there are no complex SCCs in the graph. This reduces the area of the intermediate and 3D drawings to $\mathcal{O}(|V| \cdot k)$ and of the 2D drawing to $\mathcal{O}(|V|^2)$. Complex SCCs can always be avoided by using special crossing reduction and block building algorithms [5].

Theorem 4. *The layout algorithm described in Algorithm 1 has a time complexity of $\mathcal{O}(|V| + |E|)$ for a proper ordered cyclic k-level graph $G = (V, E, \phi, <)$.*

5 Summary

We presented the first coordinate assignment algorithm for cyclic level graphs. Like the established algorithm by Brandes and Köpf, which is the de facto standard coordinate assignment method for hierarchic level graphs, we ensure to have at most two bends per edge and try to align long edges and center parents over their children. These are the major aesthetic criteria for such drawings. We implemented a prototype of our algorithm within the Gravisto toolkit [2].

References

1. Bachmaier, C., Brandenburg, F.J., Brunner, W., Lovász, G.: Cyclic leveling of directed graphs. In: Tollis, I., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 348–359. Springer (2009)
2. Bachmaier, C., Brandenburg, F.J., Forster, M., Holleis, P., Raitner, M.: Gravisto: Graph visualization toolkit. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 502–503. Springer (2004), `http://gravisto.fim.uni-passau.de/`
3. Brandes, U., Köpf, B.: Fast and simple horizontal coordinate assignment. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 31–44. Springer (2001)
4. Garey, M.R., Johnson, D.S.: A Guide to the Theory of NP-Completeness. W. H. Freemann, New York (1979)
5. Hübner, F.: A Global Approach on Crossing Minimization in Hierarchical and Cyclic Layouts of Leveled Graphs. Master's thesis, University of Passau (2009)
6. Kaufmann, M., Wagner, D.: Drawing Graphs, LNCS, vol. 2025. Springer (2001)
7. Michal, G. (ed.): Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology. Wiley (1998)
8. Serafini, P., Ukovich, W.: A mathematical model for periodic scheduling problems. SIAM J. Discrete Math. 2(4), 550–581 (1989)
9. Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Trans. Syst., Man, Cybern. 11(2), 109–125 (1981)