Drawing Directed Graphs Clockwise

Christian Pich

Chair of Systems Design, ETH Zürich*
cpich@ethz.ch

Abstract. We present a method for clockwise drawings of directed cyclic graphs. It is based on the eigenvalue decomposition of a skew-symmetric matrix associated with the graph and draws edges clockwise around the center instead of downwards, as in the traditional hierarchical drawing style. The method does not require preprocessing for cycle removal or layering, which often involves computationally hard problems. We describe an efficient algorithm which produces optimal solutions, and we present some application examples.

1 Introduction

Directed graphs are usually drawn with the desire to have edges pointing in the same direction, say, downwards, assuming that there is a general trend or direction of flow in the graph. The most popular and thoroughly researched drawing method is the Sugiyama framework [15], which works well for directed graphs with no or only few cycles. After preprocessing, in which some edges are temporarily removed or reversed, the graph is acyclic, which allows all nodes to be assigned to layered in such a way that all edges point in the same direction.

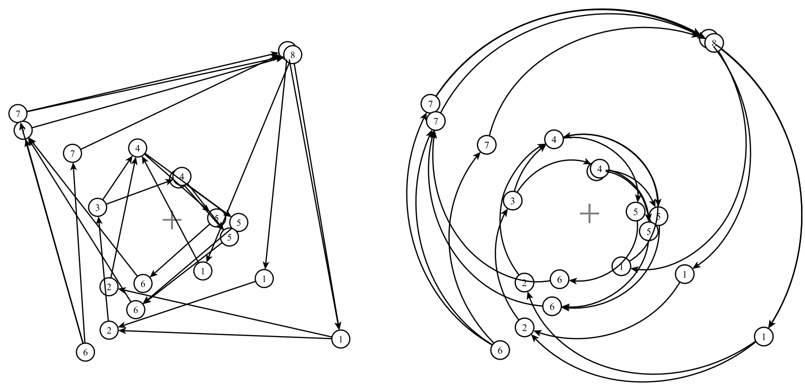
Instead of discrete levels, nodes may also be assigned continuous vertical coordinates. Carmel et al. [2] minimize a hierarchy energy in which every edge in a directed graph induces a target height difference between the two incident nodes; an iterative optimization process computes coordinates which attain these height differences as well as possible. Sometimes, however, it is not appropriate to assume that there is an overall linear trend of direction; cycles may not just be considered as "noise", but as essential information which should be highlighted and conveyed in a drawing.

An alternative to the traditional style of hierarchical layouts are *recurrent hierarchies* [15], which have long gone unnoticed until recently. Such drawings are read clockwise with respect to a distinguished point of origin. For constructing a drawing, a cyclic order on all nodes has to be found in which as many edges as possible point forward.

Sugiyama and Misue introduced a set of modifications of force-directed algorithms to get a cyclic orientation [14]. They use a concentric force field which rotates around the center and takes edges along, and report about satisfactory

* Part of this work was done while the author was at the University of Konstanz, Department of Computer & Information Science.

D. Eppstein and E.R. Gansner (Eds.): GD 2009, LNCS 5849, pp. 369–380, 2010.



(a) straight line edges (b) curved edges

Fig. 1. Clockwise drawings of the graph on the cover page of [8]. The crosses indicate the location of the origin, relative to which the configuration is oriented. The labels represent the layers in the original drawing.

experimental results for small example graphs. This method is intuitive and works for small graphs, but is, like many other force-directed methods, susceptible to local minima, sensitive to the choice of initial configurations, and not very scalable.

Bachmaier et al. extend the traditional Sugiyama approach by a cyclic level assignment [1]. The lowest level is considered to be on top of the highest level; this modification renders some of the involved optimization problems $\mathcal{NP}$-complete; for the combinatorial background of cyclic arrangements for directed graphs, see also [3,11]. The assignment of levels to nodes in such a cyclic setting is done with various heuristics.

We describe a novel approach for drawing directed graphs in a cyclic style, which does not require a discrete leveling, and gives direct and, in a sense to be specified later, optimal solutions; see Fig. 1 for an example. Positions are given by eigenvectors of a matrix associated with the graph, which is technically similar in style to other spectral layout methods [10], but conceptually different. We give the mathematical background and present a drawing algorithm which is efficient and easy to implement, together with some application results.

2 Preliminaries

In the following, let $G = (V, E)$ be a directed connected graph with directed edges $(u, v) \in E \subseteq V \times V$. The cardinalities of node and edge sets are denoted by $n = |V|, m = |E|$. When $(u, v) \in E$ we say that u *precedes* v and v *succeeds* u. The sets of predecessors and successors of a node $v \in V$ are denoted $N^-(v)$ and $N^+(v)$. Node coordinates are written as column vectors of the form $x = (x_v)_{v \in V} \in \mathbb{R}^n$, and the norms of vectors and matrices are denoted by $\|x\| = (\sum_{v \in V} x_v^2)^{1/2}$ and $\|A\| = (\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2)^{1/2}$.

3 Skew-Symmetry

Let $A = A(G) = (a_{uv})_{u,v \in V}$ denote the *adjacency matrix* of G, with entries

$$a_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We will assume in the following that between every pair of nodes $u, v \in V$ there is at most one directed edge, and that there are no self-edges (v, v).

From the adjacency matrix, which is asymmetric in general, a *skew-symmetric* matrix is derived. A square matrix $S = (s_{uv})_{u,v \in V}$ is skew-symmetric if and only if $s_{uv} = -s_{vu}$ for all $u, v \in V$, or equivalently, $S = -S^T$.

The *skew-symmetric adjacency matrix* $S(G)$ of a directed graph $G = (V, E)$ is connected to its adjacency matrix $A(G)$ by

$$S = S(G) = A(G) - A(G)^T \tag{2}$$

with entries

$$s_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E, (v,u) \notin E \\ -1 & \text{if } (v,u) \in E, (u,v) \notin E \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

for all $u, v \in V$.

We will now use the eigenvalues and eigenvectors of S to obtain positions for every node and thus a drawing of G. Without loss of generality, the *eigenvalue decomposition* of S may be written in the form

$$S = U \Phi U^T \ , \tag{4}$$

where $U \in \mathbb{R}^{n \times n}$ is an orthogonal matrix whose columns are real unit length eigenvectors $u_1, \ldots, u_n \in \mathbb{R}^n$, $\|u_i\| = 1$ for all $i \in \{1, \ldots, n\}$, and $\Phi \in \mathbb{C}^{n \times n}$ is a diagonal matrix of complex eigenvalues.

Since S is skew-symmetric, the complex eigenvalues of S are purely imaginary and occur in conjugated complex pairs

$$\pm\sqrt{-1}\phi_1, \pm\sqrt{-1}\phi_2, \ldots, \pm\sqrt{-1}\phi_{\lfloor n/2 \rfloor} \tag{5}$$

with an additional singleton zero eigenvalue if n is odd. We will refer to a pair of eigenvalues $\pm\sqrt{-1}\phi_i$ as *the eigenvalue* ϕ_i. Without loss of generality we assume that the eigenvalues in Φ are ordered non-increasingly by their absolute magnitude, $\phi_1 \geq \cdots \geq \phi_{\lfloor n/2 \rfloor} \geq 0$. With the i-th eigenvalue ϕ_i $(1 \leq i \leq \lfloor n/2 \rfloor)$ a pair of eigenvectors u_{2i-1}, u_{2i} is associated, which span a two-dimensional space frequently called *(i-th) bimension.*

Through orthogonal transformation, (4) can be brought into a slightly different form known as the *Gower decomposition* [5]

$$\left[\begin{array}{ccccc} u_1 & u_2 & \dots & u_{n-1} & u_n \end{array}\right] \left[\begin{array}{cccccc} 0 & \phi_1 & & & & \\ -\phi_1 & 0 & & & & \\ & & \ddots & & & \\ & & & & 0 & \phi_{\lfloor \frac{n}{2} \rfloor} \\ & & & & -\phi_{\lfloor \frac{n}{2} \rfloor} & 0 \end{array}\right] \left[\begin{array}{c} u_1^T \\ u_2^T \\ \vdots \\ u_{n-1}^T \\ u_n^T \end{array}\right] \tag{6}$$

which allows S to be written as a sum of $\lfloor n/2 \rfloor$ elementary rank-2 matrices

$$S = \sum_{i=1}^{\lfloor n/2 \rfloor} \phi_i \left(u_{2i} u_{2i-1}^T - u_{2i-1} u_{2i}^T \right), \tag{7}$$

all of which are skew-symmetric.

An intuitive interpretation of the decomposition (7) is that each of the (at most) $\lfloor n/2 \rfloor$ summands explains a share of the directional information expressed by S; the magnitude of the eigenvalue ϕ_i is equal to the share of the ith bimension. Note that a pair of eigenvectors u_{2i-1}, u_{2i} may be replaced by any orthogonal pair of vectors spanning the same two-dimensional space.

4 Clockwise Drawings

Each of the $\lfloor n/2 \rfloor$ bimensions of $S(G)$ may be used to obtain a two-dimensional drawing of a graph $G = (V, E)$. Since ϕ_1 is the largest eigenvalue, the information expressed by the edges of G is best captured in two-dimensions by using the corresponding eigenvectors u_1 and u_2 as follows.

Positions for every node $v \in V$ are simply obtained by setting

$$x = \sqrt{\phi_1} u_1, y = \sqrt{\phi_1} u_2 \tag{8}$$

and using the entries x_v, y_v as the coordinates of v in two-dimensional Euclidean space. In such a configuration, the particular skew-symmetry s_{uv} between two nodes u and v, which comes from the orientation (or absence) of the edge (u, v), is fitted by

$$s_{uv} \approx x_u y_v - x_v y_u. \tag{9}$$

This quantity is proportional to the *signed* area of the triangle of the positions (x_u, y_u) and (x_v, y_v) subtended by the origin, since

$$x_u y_u - (x_v y_v/2 + x_u y_v/2 + (x_u - x_v)(y_v - y_u)/2) = (x_u y_v - x_v y_u)/2$$

as illustrated in Fig. 2.

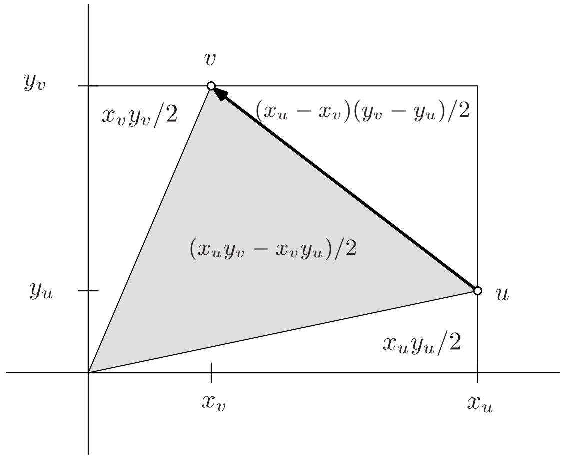


Fig. 2. The signed area of the triangle of the edge pointing from (x_u, y_u) to (x_v, y_v) subtended by the origin represents the amount and direction of the skew-symmetry between u and v

It can be shown that setting the positions (x_v, y_v) for each node $v \in V$ as in (8) minimizes the objective function

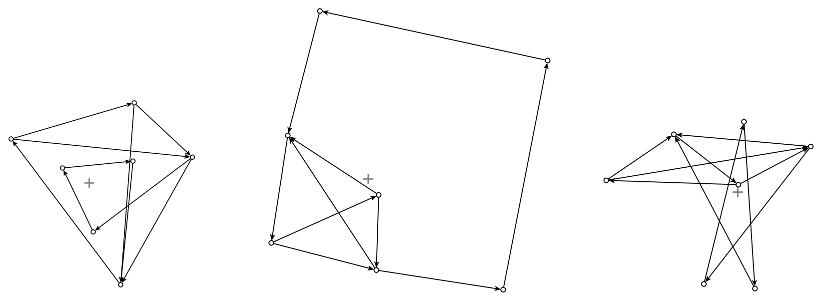
$$\sum_{(u,v)\in E} (x_u y_v - x_v y_u - 1)^2 + \sum_{(u,v),(v,u)\notin E} (x_u y_v - x_v y_u)^2 \qquad (10)$$

among all two-dimensional layouts $x, y \in \mathbb{R}^n$ [6]. Intuitively, minimizing (10) tries to represent all directed edges with a correspondingly oriented triangle having *positive unit area*, while all non-adjacent pairs of nodes should be located on a line through the origin, forming a triangle with area zero; this may be interpreted as a global repulsion energy for non-adjacent node pairs.

Unlike distance-based layout methods, the origin and the relation of nodes to the origin are crucial for reading the clockwise drawing. The angle of a node's position is determined largely by the angle of its predecessors and successors. Since two eigenvectors spanning a bimension share the same eigenvalue, axes are not meaningful, and the configuration may be freely rotated around the center without modifying triangle areas and signs. Furthermore, it is not determined whether the bimension blocks in the block-diagonal matrix in (6) are of the form

$$\begin{pmatrix} 0 & \phi_i \\ -\phi_i & 0 \end{pmatrix} \qquad \text{or} \qquad \begin{pmatrix} 0 & -\phi_i \\ \phi_i & 0 \end{pmatrix}$$

so that the orientation is made clockwise or counterclockwise, as desired, by reflecting it on any line through the origin. Note that some edges may point against the desired orientation because the associated triangle areas are negative. Depending on the context, these edges may be visually highlighted, or re-oriented by letting them follow the opposite, longer way around the origin.



(a) $\phi_1 = 2.65843$ (b) $\phi_2 = 1.55086$ (c) $\phi_3 = 0.72802$

Fig. 3. Drawings of an example graph ($n = 7, m = 10$) in all three possible bimensions. Note that $\phi_1^2 + \phi_2^2 + \phi_3^2 = m$, and the bimensions account for about $\phi_1^2/m \approx 70.6\%$, $\phi_2^2/m \approx 24.1\%$ and $\phi_3^2/m \approx 5.3\%$ of the skew-symmetry information.

Although the bimension for the largest eigenvalue is the best one can do with two dimensions in the sense of the criterion (10), drawings in other bimensions may also be helpful, since they visualize additional, less dominant parts of the directional information. A small graph and the layout in all possible bimensions is given in Fig. 3. The second bimension explains as much as possible of the skew-symmetry remaining after removing the contribution of the first bimension from the sum in (7), and so on.

5 Implementation

There are some dedicated algorithms for computing the complete spectral decomposition (4) of a skew-symmetric matrix [12,16]. Fortunately, we need only the largest eigenvalue and the two associated eigenvectors, and a simple power iteration is thus sufficient.

Since $S = -S^T$ and hence $-SS^T = S^2$, we can use the fact that the eigenvectors of S are identical to the eigenvectors of the symmetric matrix $SS^T = -S^2$, and power-iterate with SS^T, which is convenient to handle computationally.

An initial non-zero vector, which may be chosen randomly, is iteratively multiplied with SS^T by carrying out the multiplication step

$$x \leftarrow \frac{SS^T x}{\|SS^T x\|}. \tag{11}$$

over and over again; in general, x will converge to the desired eigenvector [4]. Instead of materializing the matrix SS^T, which would require $\mathcal{O}(n^3)$ real multiplication operations, the step (11) can be split into

$$\hat{x} \leftarrow \frac{S^T x}{\|S^T x\|} \tag{12}$$

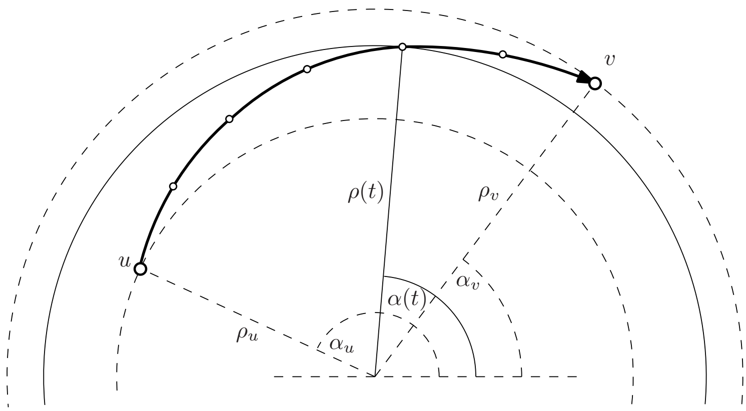


Fig. 4. Drawing curved edges using splines with control points

$$x \leftarrow \frac{S\hat{x}}{\|S\hat{x}\|} \tag{13}$$

which encompasses only $\mathcal{O}(n^2)$ operations, assuming a constant number of iterative steps to achieve convergence. Furthermore, sparsity of a graph G, i.e., when $S(G)$ has $o(n^2)$ non-zero entries, may be exploited to obtain a power iteration with linear time per step. The power iteration process in (12) and (13) becomes

$$\hat{x} \leftarrow \frac{Ax - A^T x}{\|Ax - A^T x\|} \tag{14}$$

$$x \leftarrow \frac{A^T \hat{x} - A\hat{x}}{\|A^T \hat{x} - A\hat{x}\|} \tag{15}$$

which is just a linear scan over all edges, since only positions of adjacent nodes need to be accumulated; this is reminiscent of hubs and authorities [9], where the eigenvectors of AA^T and $A^T A$ are computed.

The required second eigenvector of SS^T is computed similarly, but with orthogonalizing against the first eigenvector after each step. Pseudo-code of an algorithm with running time and space complexity in $\mathcal{O}(n + m)$ per iteration step is given in Alg. 1.

To avoid unnecessary crossings by straight lines, edges may be drawn as clockwise curves around the origin, e.g., using splines. The corresponding control points are determined in a linear interpolation between the angles α_u, α_v, and the radii ρ_u, ρ_v of the nodes u, v by

$$\rho(t) = (1 - t) \cdot \rho_u + t \cdot \rho_v \tag{16}$$

$$\alpha(t) = (1 - t) \cdot \alpha_u + t \cdot \alpha_v \tag{17}$$

Algorithm 1. Drawing a directed graph clockwise

Input: Directed graph $G = (V, E)$
Output: Coordinate vectors $x, y \in \mathbb{R}^n$ with positions for every $v \in V$
$x \leftarrow \text{random}, y \leftarrow \text{random}$
while *x and y change significantly* **do**
  $x \leftarrow x/\|x\|, y \leftarrow y/\|y\|$ // normalize
  $y = y - x^T y \cdot x$ // orthogonalize
  foreach $v \in V$ **do**
    $\hat{x}_v \leftarrow \sum_{u \in N^-(v)} x_u - \sum_{w \in N^+(v)} x_w$ // $\hat{x} \leftarrow (A - A^T) \cdot x$
    $\hat{y}_v \leftarrow \sum_{u \in N^-(v)} y_u - \sum_{w \in N^+(v)} y_w$ // $\hat{y} \leftarrow (A - A^T) \cdot y$
  foreach $v \in V$ **do**
    $x_v \leftarrow \sum_{w \in N^+(v)} \hat{x}_w - \sum_{u \in N^-(v)} \hat{x}_u$ // $x \leftarrow (A^T - A) \cdot \hat{x}$
    $y_v \leftarrow \sum_{w \in N^+(v)} \hat{y}_w - \sum_{u \in N^-(v)} \hat{y}_u$ // $y \leftarrow (A^T - A) \cdot \hat{y}$
  $\phi \leftarrow \sqrt{\|x\|}$ // estimate for largest eigenvalue
$x \leftarrow x/\phi^{3/2}, y \leftarrow y/\phi^{3/2}$ // scale eigenvectors to have length $\sqrt{\phi}$

where $0 \leq t \leq 1$; when k control points are used, $t \in \{0, \frac{1}{k}, \frac{2}{k}, \ldots, \frac{k-1}{k}, 1\}$. Note that when $|\alpha_u - \alpha_v| > \pi$, this interpolation results in the edge (u, v) winding around the center with an angle greater than π; the shorter counterpart of that curve is obtained by adding 2π to the smaller of α_u, α_v.

6 An Application

A special class of directed graphs is called *tournaments* [7,13]. A tournament $G = (V, E)$ on n nodes is an orientation of the complete undirected graph on n nodes. Tournaments are a model for round-robin competitions in which everybody competes with everybody else, and every competition $\{u, v\}$ for $u, v \in V, u \neq v$ has a winner u and a loser v, say, which is represented by the orientation (u, v).

Here we use a variant of tournaments, in which the underlying undirected graph is almost complete, but some edges are allowed to be missing because there are situations in which no winner can be determined. The method of clockwise drawing is applied to results of international football leagues in England, Germany, Italy, and Spain, in the seasons ending in 2006, 2007, and 2008. In every season, between every possible pair of teams two matches are carried out, each team being the home team once. The tournament graph contains an edge $(u, v) \in E$ when u dominates v, i.e., u has won more matches against v than v against u; ties are not considered.

Fig. 5 shows drawings of all 12 tournaments, as given by the positions in the bimension of the largest eigenvalue. A cyclic structure is displayed in some of

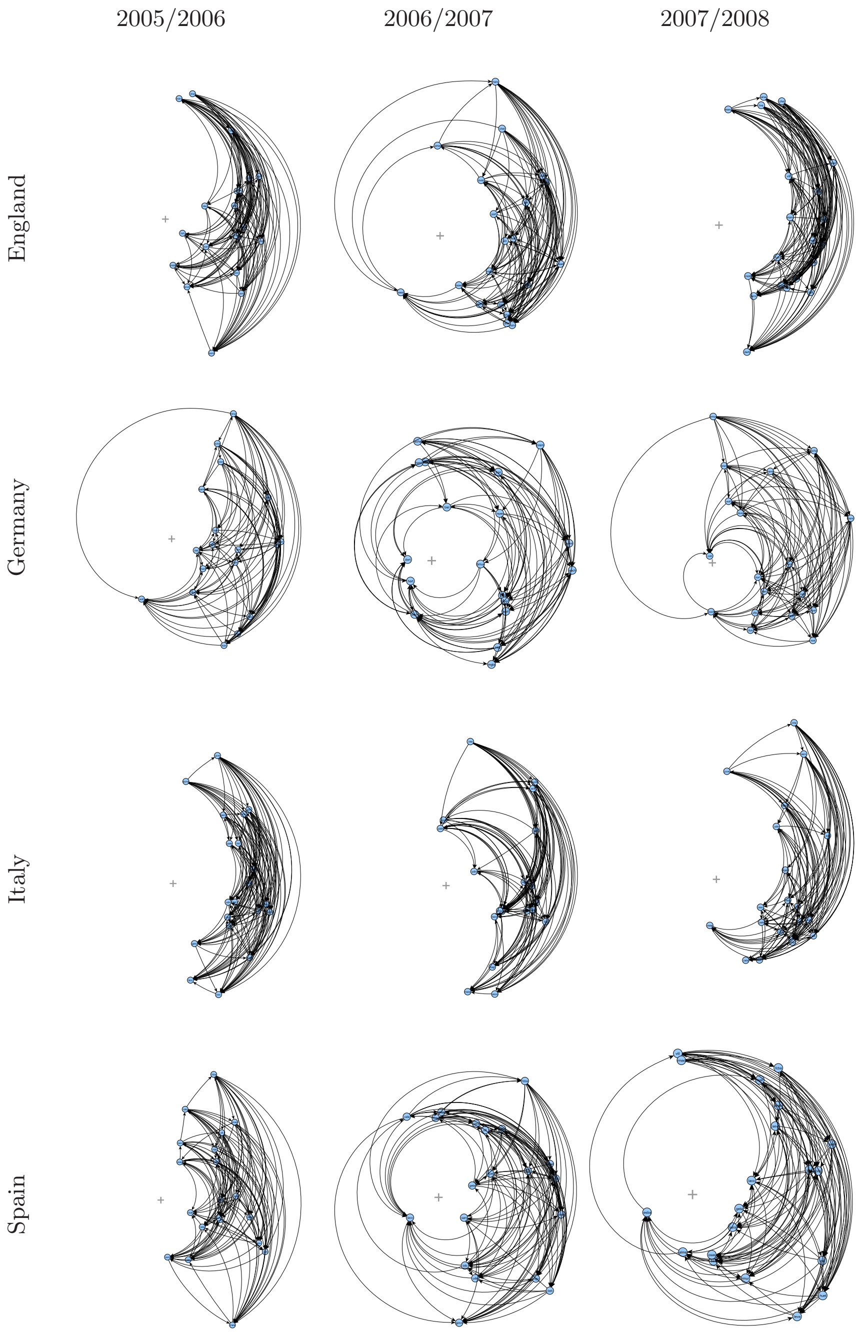
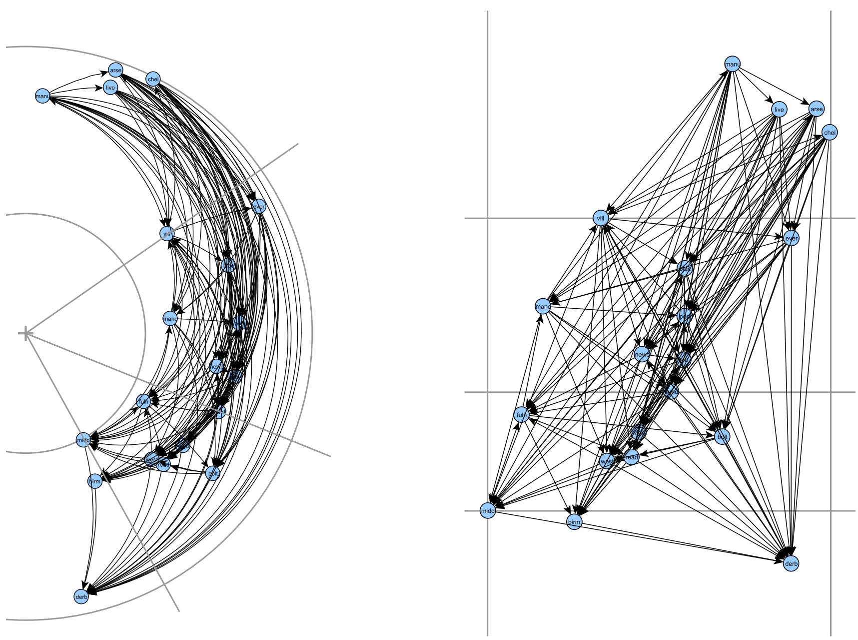


Fig. 5. Clockwise drawings (bimension of the largest eigenvalue) of the tournament graphs in European football leagues in three consecutive seasons. The span of edges around the center is emphasized by edges curving around the center.

the configurations, such as England 2006/2007, Germany 2006/2007, and Spain 2006/2007, 2007/2008, which leads to the conjecture that these seasons were quite balanced, with no clear dominator. In these tournaments, some otherwise weak teams, which are dominated by most others, won against otherwise strong teams. For example, in the 2006/2007 season of the English Premier League, West Ham United (node on the lower left) closed the season on rank 15 of 20 teams, but dominated the champions Manchester United and fourth-ranked Arsenal FC.

In contrast, it is interesting to observe that the drawings of some other tournaments appear to be rather non-cyclic, especially England 2005/2006 and 2007/2008, all three seasons in Italy, and Spain 2005/2006. Since all nodes are on the same side of a line through the origin, the signed triangle areas do not allow for cyclic node triples in this bimension. Thus, most of the dominance structure in the skew-symmetric adjacency matrix is intrinsically rather non-cyclic, and suggests that the classical hierarchical approach is actually more appropriate than the cyclic one. In the context of football matches, there is a clear tendency for strong teams to consistently dominate weaker teams and weak teams to be consistently dominated by stronger teams, with no or only few exceptions.



(a) half-circular (b) hierarchical

Fig. 6. Football tournament graph in England 2007/2008 (bimension of the largest eigenvalue). The graph exhibits a substantially hierarchical structure, which justifies the transformation from a polar into a cartesian domain.

In fact, a polar transformation easily transforms the half-circular arrangement into the traditional hierarchical drawing style. A natural ranking is given by the total order of angles of all nodes with respect to the origin, as they are given by their angular representation. The coordinates for node v after this polar transformation are given by

$$\rho_v = \sqrt{x_v^2 + y_v^2}, \quad \alpha_v = \text{atan2}(y_v, x_v) \tag{18}$$

where $\text{atan2}(\cdot, \cdot)\colon \mathbb{R}^2 \to [0, 2\pi]$ denotes the two-argument inverse of the tangent function implemented in most modern programming languages. ρ_v represents the transformed clockwise rank of v and α_v the amount of skew-symmetry of v with all other nodes. An example of such a half-clockwise configuration and its polar transform is given in Fig. 6.

7 Conclusion

The decomposition of the skew-symmetric adjacency matrix yields a method for drawing directed graphs in a cyclic fashion and provides direct and unique solutions. The drawing area is oriented either clockwise or counterclockwise around a distingiuished center point; if necessary, the sense of rotation is inverted by reflecting one axis.

The algorithm is easy to implement because it requires only essential array operations and no sophisticated data structures. Since no cycle removal or level assignment is required, some of the computationally hard problems are avoided. The sparsity of the skew-symmetric adjacency matrix can be used to obtain a power iteration algorithm which runs in linear time per step and requires linear space.

When discrete levels or radial level assignments are required, they may be obtained from the continuous coordinates by a quantization scheme. The clockwise configurations can be combined with the force-directed methods in [1,14]. A straightforward extension would be to use non-uniform edge lengths. For strongly connected graphs, all distances are finite, and the analysis is also applicable to the corresponding skew-symmetric distance matrix.

While there is no space here for a detailed discussion of quantitative measures to characterize the cyclicity of a dominance relation, we would like to point out that clockwise oriented drawings and the distribution of the involved eigenvalues are useful for testing hypotheses about the cyclic or hierarchical structure of directed graphs.

Beyond the graph drawing application, we expect that the presented method is also useful for generating initial solutions to heuristic methods for general $\mathcal{NP}$-hard arrangement problems [3,11].

References

1. Bachmaier, C., Brandenburg, F.J., Brunner, W., Lovász, G.: Cyclic leveling of directed graphs. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 348–359. Springer, Heidelberg (2009)

2. Carmel, L., Harel, D., Koren, Y.: Combining hierarchy and energy for drawing directed graphs. IEEE Transactions on Visualization and Computer Graphics 10(1), 46–57 (2004)
3. Ganapathy, M.K., Lodha, S.P.: On minimum circular arrangement. In: Diekert, V., Habib, M. (eds.) STACS 2004. LNCS, vol. 2996, pp. 394–405. Springer, Heidelberg (2004)
4. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
5. Gower, J.C.: The analysis of asymmetry and orthogonality. In: Recent Developments in Statistics, pp. 109–123 (1977)
6. Gower, J.C., Constantine, A.G.: Graphical representation of asymmetric matrices. Applied Statistics 27, 297–304 (1978)
7. Harary, F., Moser, L.: The theory of round robin tournaments. Amer. Math. Monthly 73, 231–246 (1966)
8. Kaufmann, M., Wagner, D. (eds.): Drawing Graphs. LNCS, vol. 2025. Springer, Heidelberg (2001)
9. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
10. Koren, Y.: On spectral graph drawing. In: Warnow, T.J., Zhu, B. (eds.) COCOON 2003. LNCS, vol. 2697, pp. 496–508. Springer, Heidelberg (2003)
11. Liberatore, V.: Circular arrangements and cyclic broadcast scheduling. Journal of Algorithms 51(2), 185–215 (2004)
12. Paardekooper, M.H.C.: An eigenvalue algorithm for skew-symmetric matrices. Numerische Mathematik 17(3), 189–202 (1971)
13. Reid, K.B., Beineke, L.W.: Tournaments. In: Beineke, L.W., Wilson, R.J. (eds.) Selected Topics in Graph Theory, pp. 169–204. Academic Press, London (1978)
14. Sugiyama, K., Misue, K.: A simple and unified method for drawing graphs: Magnetic-spring algorithm. In: Tamassia, R., Tollis, I.G. (eds.) GD 1994. LNCS, vol. 894, pp. 364–375. Springer, Heidelberg (1995)
15. Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics 11(2), 109–125 (1981)
16. Ward, R.C., Gray, L.C.: Ward and Leonard C. Gray. Eigensystem compuation for skew-symmetric matrices and a class of symmetric matrices. ACM Transactions on Mathematical Software 4(3), 278–285 (1978)