University of Ljubljana
Institute of Mathematics, Physics and Mechanics
Department of Mathematics
Jadranska 19, 1000 Ljubljana, Slovenia

Preprint series, Vol. 40 (2002), 829

DRAWING METHODS FOR 3-CONNECTED PLANAR GRAPHS

Alen Orbanić   Tomaž Pisanski
Marko Boben   Ante Graovac

ISSN 1318-4865

July 31, 2002

Ljubljana, July 31, 2002

Drawing Methods for 3-connected Planar Graphs

Alen Orbanić[*], Tomaž Pisanski[†], Marko Boben[‡]

University of Ljubljana, IMFM
Jadranska 19, SI-1000 Ljubljana, Slovenia

Ante Graovac[§]

Rudjer Bošković Institute
POB 180, HR-10002 Zagreb, Croatia

Abstract

According to Whitney, every 3-connected planar graph admits a unique embedding in the sphere or, equivalently, in the plane. Tutte showed that the combinatorial information is sufficient for providing a straight line planar embedding. In 1922 Steinitz proved that 3-connected planar graphs are exactly skeletons of convex 3D polyhedra. This means that the topological characterization of Whitney extends to 3D geometrical representation. Tutte's simple and efficient method can be lifted into 3D polyhedral drawing using methods involving Maxwell-Cremona stress theorem.

The Laplace method for graph drawing proved to be suitable for 3D representations of some classes of graphs. However, there are no guaranties that the faces will be planar. Recently it was proved (Lovász et al.) that the so-called Colin de Verdière generalization of Laplace matrix could give very good polyhedral representations of 3-connected planar graphs.

In these presentation algorithmic aspects of above approach to the problem is discussed. The algorithmic construction of convex polyhedron corresponding to planar 3-connected graphs has numerous applications in crystallography and chemistry.

[*] `Alen.Orbanic@fmf.uni-lj.si`

[†] `Tomaz.Pisanski@fmf.uni-lj.si` supported in part by Ministrstvo za šolstvo, znanost in šport Republike Slovenije, grant J1-6161, J2-6193.

[‡] `Marko.Boben@fmf.uni-lj.si`

[§] `graovac@rudjer.irb.hr`

1 Representations of Graphs

Graphs are combinatorial structures for representing several real-life structures, having connections between some pairs of nodes, for instance atoms and bonds. In a graph G nodes are called *vertices* and connections are called *edges*, represented by sets $V(G)$ and $E(G)$, respectively. Graphs in raw form contain only information about connectivity of nodes. One can assign additional information to edges. This information is most often numerical and is called a *weight*, for instance: edge ij between vertices i and j has weight ω_{ij}. Additional information can also be assigned to vertices.

Imagine that a graph G represents a molecule (vertices = atoms, edges = bonds). Having additional information in vertices (type of atom) and weights on edges (for instance: length of bond), one can try to find a visualization of the graph G in $\mathbb{R}^d$, $d = 2, 3$.

Graphs are visualized in $\mathbb{R}^d$ by means of a map $\rho : V(G) \cup E(G) \to \mathbb{R}^d$, $d = 2, 3$, which is called a *representation* of a graph in $\mathbb{R}^d$. If we regard vectors $\rho(u)$, $u \in V(G)$, as row vectors, we may represent ρ by $|V(G)| \times d$ matrix R with the images of the vertices of G as its rows. The representation ρ is *orthonormal* if $R^T R = I_m$. A representation ρ is called *balanced* if $\sum_{u \in VG} \rho(u) = 0$.

The idea of graph representation goes back at least to Tutte [11, 12], where it is stated primarily for 3-connected planar graphs.

The *energy* of the representation ρ is defined in general form to be the value:

$$\mathcal{E}(\rho) = \sum_{uv \in E(G)} \omega_{uv} ||\rho(u) - \rho(v)||^2 \qquad (1)$$

where $\omega : E(G) \to \mathbb{R}^+$ is a map defining an edge-weighted graph.

2 Drawing Methods

Methods for drawing graphs can be divided into two categories:

- Iterative methods: spring-embedders such as `NiceGraph` in VEGA [7].
- Exact methods: Laplace method [9], Tutte method [12], Lovász method [5] Stress based methods [2], [3]

2.1 NiceGraph

NiceGraph algorithm is a typical spring-embedder algorithm in $\mathbb{R}^d$, $d = 2, 3$. The vertices of a graph G are represented by points in $\mathbb{R}^d$, the edges between vertices are represented by straight lines between corresponding two vertices and operate as springs producing attractive force between vertices. Between each two non-adjacent vertices there is a repulsive force. In each iteration we calculate resultant forces of all other vertices and springs to each vertex. According to the direction and magnitude of resultant force vector, each vertex is moved towards direction of the force vector by the distance that is some function f of magnitude of the vector.

The method is easy to implement and is relatively fast. The drawings produced by the method are "nice" in general. One cannot use this algorithm alone to draw 3-connected planar graphs in $\mathbb{R}^3$ as skeletons of convex polyhedra since there is no guarantee that the resulting drawing would be anyhow similar to some kind of skeleton of convex polyhedra. Remarkably, the drawings obtained by this method are nearly convex polyhedral. The problems are with convexity and facets that are in general not planar.

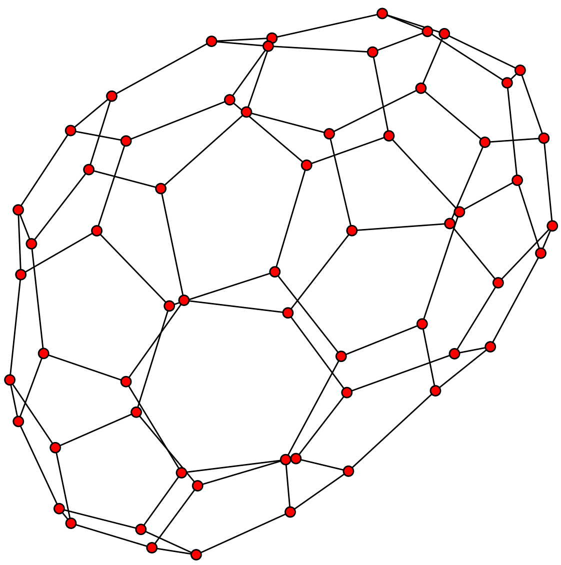

Figure 1: Fullerene C_{60} using `NiceGraph`.

2.2 The Laplace Method

The matrix Q with the elements $q_{uv} = -\omega_{uv}, uv \in EG, q_{uv} = 0, uv \notin EG, q_{uu} = -\sum_{uv \in EG} q_{uv}$ is called the *generalized Laplacian* of an edge-weighted graph G.

Theorem 2.1 ([9, 6]). *Let G be an edge-weighted graph with edge-weights ω and the generalized Laplacian Q. Assume that the eigenvalues of Q are $\lambda_1 \leq \cdots \leq \lambda_n$ and that $\lambda_2 > 0$. The minimum energy of a balanced orthonormal representation of G in $\mathbb{R}^m$ equals $\sum_{i=2}^{m+1} \lambda_i$.*

Note that the orthonormal representation ρ of the above Theorem is given by the matrix $[x_2, \ldots, x_{m+1}]$ composed of orthonormal eigenvectors corresponding to $\lambda_2, \ldots, \lambda_{m+1}$. For $m = 2$ and $m = 3$ we get a graph drawing in $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively. Examples of such drawings are shown in Figure 2. Any procedure that obtains a representation of a graph by solving the eigenvalue and eigenvector problem will be called the *eigenvector method.* In the above case, Theorem 2.1 guarantees that the eigenvector method produces a representation that minimizes the energy given by (1).

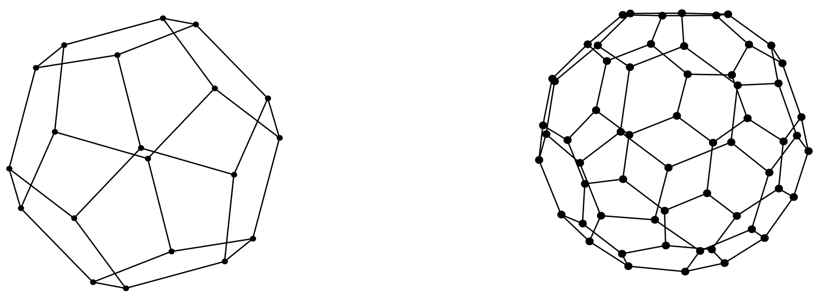

Figure 2: A $\mathbb{R}^2$ representation of the dodecahedron using second and third eigenvectors of the Laplace matrix Q and a $\mathbb{R}^3$ representation of the C_{60} fullerene, using second, third and fourth eigenvectors.

2.3 The Tutte method.

A cycle C of G is called *peripheral* if no edge not in C joins two vertices in C and $G \setminus C$ is connected. For example, any face of a 3-connected planar graph can be shown to be a peripheral cycle.

We say, that a representation ρ of G is *barycentric* relative to a subset S of VG if for each $u \notin S$ the vector $\rho(u)$ is the barycenter of the images of neighbors of u.

Theorem 2.2 (Tutte). *Let C be a peripheral cycle in a connected graph G. Let σ be a mapping from VC to the vertices of a convex $|VC|$-gon in $\mathbb{R}^2$ such that adjacent vertices in C are adjacent in the polygon. The unique barycentric representation determines a drawing of G in $\mathbb{R}^2$. This drawing has no crossings if and only if the graph is planar.*

A barycentric drawing based on this theorem is obtained by solving the system of equations

$$\rho(v) = \frac{1}{\deg(v)} \sum_{u \in N(v)} \rho(u), \qquad v \in VG \setminus S. \tag{2}$$

It is sometimes called the *Tutte drawing* of a graph.

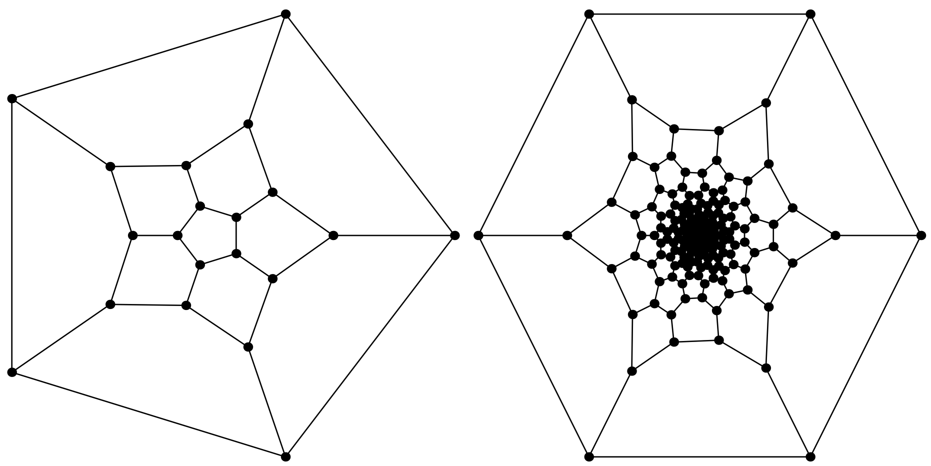

Figure 3: A Tutte $\mathbb{R}^2$ representation of the dodecahedron and $Le(C_{60})$.

The Tutte method can be generalized if weights ω_{ij} are used in (2) instead of $\frac{1}{\deg v}$. Compare [8].

2.4 The Lovász method

The Lovász method is based on null-space representation of "almost" Colin de Verdière matrices.

Let G be a 3-connected graph with the vertex-set $V(G)$ and the edge-set $E(G)$ and M be a $|V(G)| \times |V(G)|$ matrix such that

1. $M_{ij} \begin{cases} < 0 & \text{if } ij \in E(G) \\ = 0 & \text{if } ij \notin E(G) \text{ and } i \neq j \\ = \text{arbitrary} & \text{if } i = j \end{cases}$

2. M has exactly one negative eigenvalue (of multiplicity one).

3. $\operatorname{corank}(M) = 3$.

It is known [4] that corank of M corresponding to 3-connected planar graph equals 3. Let x_1, x_2, x_3 be a base of the null-space of M, where $x_i = (u_{i1}, \ldots, u_{in})$, $n = |V(G)|$. The null-space representation of G is defined as mapping:

$$\rho : V(G) \to \mathbb{R}^3, \qquad \rho : j \to \frac{u_j}{|u_j|},$$

where $u_j = (u_{1j}, u_{2j}, u_{3j})$ is a vector consisting of the j-th components of vectors x_i. In [4] it was proved that ρ induces an embedding of G into a unit sphere. Edges are represented as geodesics between the images of their endpoints.

Furthermore, in [5] it is shown that a so-called properly scaled Colin de Verdière matrix and a mapping

$$\begin{aligned} &\rho' : V(G) \cup E(G) \to \mathbb{R}^3, \\ &\rho' : j \to u_j, \\ &\rho' : ij \to \text{straight line between } \rho'(i) \text{ in } \rho'(j). \end{aligned}$$

yield a polyhedral representation of G in $\mathbb{R}^3$, i.e. $\rho'(G)$ represents a skeleton of a convex polyhedron given by G. Properly scaled matrix can be easily obtained from matrix M satisfying conditions above.

Conversely, having a polyhedral representation of G, one can obtain a properly-scaled Colin de Verdière matrix.

Properties of the method are:

- Properly scaled Colin de Verdière matrix of G is a special case of generalized Laplacian of G.
- Once we have a properly-scaled Colin de Verdière matrix we can easily obtain a polyhedral drawing.
- Colin de Verdière matrix is, in general, difficult to obtain.

2.5 Stress based methods

Stressed graph is a weighted graph drawn in the plane such that the edges are drawn as straight line segments. Weight ω_{ij} on each edge ij is called a *stress*. Let $\omega_{ij} = 0$ if $ij \notin E(G)$ and $\omega_{ij} = \omega_{ji}$. Graph is in equilibrium stress if

$$0 = \sum_{j=1}^{n} \omega_{ij}(\rho(j) - \rho(i)), \quad i = 1, \ldots, n.$$

Let ρ be an embedding of graph G in the plane such that all faces are convex. Let F be the outer face of the embedding. The edges on F are called *exterior* edges and the other edges are called *interior* edges. We call stress ω_{ij} an *internally convex stress* if $\omega_{ij} > 0$ and ij is interior edge. If internal stresses are positive and external stresses are negative, then the stress is called a *convex stress*. If the face F is a triangle then any internally convex stress can be converted to convex equilibrium stress [1] just by solving an equilibrium stress equations for three remaining exterior stresses for edges of F.

Let ρ be a Tutte embedding of G with triangle F in plane $z = 1$. As it is known from Tutte method, all internal stresses are 1. So one can compute the stresses on triangle F and obtain convex equilibrium stress. To each face r we associate a linear function defined on the face.

$$f_r(x, y) = a_r x + b_r y + c_r. \tag{3}$$

According to [1] any solution of equations below for set of functions f_r, r is a face of $\rho(G)$, maps the faces of $\rho(G)$ to facets of corresponding polyhedra:

$$\begin{aligned} \omega_{ij} &= \frac{\delta(r,s)(f_s(x_*, y_*) - f_r(x_*, y_*))}{[\rho(i), \rho(j), p^*]} \\ f_r(x, y) &= f_s(x, y) \quad \text{for each point } (x, y) \text{ such that } (x, y, 1) \in \rho(ij), \\ & \qquad\qquad ij \text{ edge separating faces } r, s. \end{aligned} \tag{4}$$

Face F remains on the plane $z = 1$, while other faces lift to corresponding facets defined by graphs of corresponding functions f_r. Parameters in equations (3) and (4) are:

- $p^* = (x_*, y_*, 1)$ – some point in the plane $z = 1$ that is not collinear with any image of edge $\rho(ij)$.
- $[\rho(i), \rho(j), p^*] = \det(\rho(i), \rho(j), p^*)$ – usual triple product in $\mathbb{R}^3$.
- $\delta(r, s)$ orientation coefficient. $\delta(r, s) = 1$, if in the counterclockwise ordering of vertices around face r vertex i precedes vertex j. Otherwise $\delta(r, s) = -1$.

The algorithm [2]

Input: A combinatorial planar embedding of a 3-connected graph

Output: A graph drawing in $\mathbb{R}^3$ representing a skeleton of corresponding polyhedron.

1. If G does not contain a triangle, then replace G by dual G^*.
2. Choose a triangle F and draw it as an equilateral triangle on the plane $z = 1$.
3. Using Tutte's algorithm draw the remaining vertices and edges of G producing a planar drawing that has an internally convex stress. Calculate the remaining three stresses on F and obtain convex equilibrium stress on embedded graph G.
4. Find a solution of system of equations above obtaining set of linear functions f_r for each face r.
5. Using mappings f_r find the images of vertices from Tutte embedding and draw a polyhedron P.
6. If necessary, replace P with its reciprocal polyhedron P^*.

The good thing about this method is that one can use it for drawing Steinitz representations of 3-connected planar graphs. The resulting polyhedra may not be "nicely" shaped. If one tries to draw for instance a skeleton of fullerene the obtained drawing may not be satisfactory.

References

[1] J.E. Hopcroft, P.J.Kahn. *A Paradigm for Robust Geometric Algorithms.* Algorithmica (1992) 7:339–380

[2] P. Eades, P. Garvan. *Drawing Stressed Planar Graphs in Three Dimensions,* Graph Drawing '95, (1995)

[3] M. Chorbak, M.T. Goodrich, R. Tammasia. *On the Volume and Resolution of 3-Dimensional Convex Graph Drawings.*

[4] L. Lovász, A. Schrijver. On the Null Space of a Colin de Verdiére Matrix, Annal. Instit. Fourier 49 (1999) 1017-1026

[5] L. Lovász. Steinitz Representations of Polyhedra and the Colin de Verdiére Number, Journal of Comb. Theory, Series B 82,(2001) 223-236

[6] C. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, New York, 2001.

[7] T. Pisanski & coworkers, VEGA, a programming tool for manipulating discrete mathematical structures, see `http://vega.ijp.si`.

[8] T. Pisanski, M. Boben, A. Žitnik, *Interactive conjecturing in Vega*, submitted.

[9] T. Pisanski and J. Shawe-Taylor, *Characterizing Graph Drawing with Eigenvectors*, J. Chem. Inf. Comput. Sci. 40 (2000) 567–571.

[10] B. Plestenjak, *An Algorithm for Drawing Planar Graphs*, Software – Practice and Experience 29 (1999) 973–984.

[11] W. T. Tutte, *Convex representations of graphs*, Proc. London Math. Soc. 10 (1960) 304–320.

[12] W. T. Tutte, *How to draw a graph*, Proc. London Math. Soc. 13 (1963) 743–767.

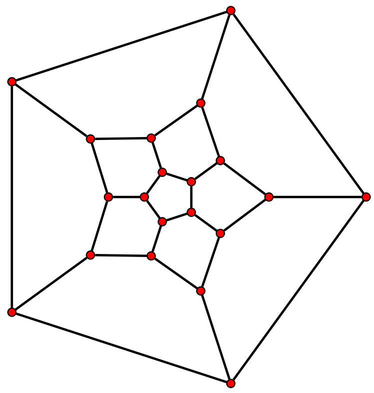

Figure 4: Dodecahedron.

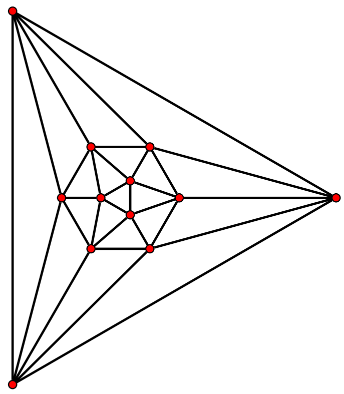

Figure 5: Dual of dodecahedron drawn by the Tutte method

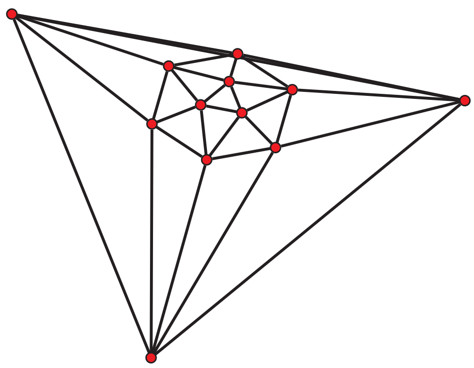

Figure 6: Lift of the plan graph of Figure 5 to an icosahedron in 3D.

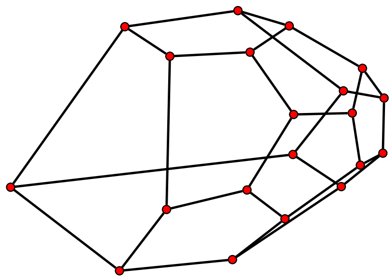

Figure 7: The reciprocal polyhedron of icosahedron of Figure 6 is dodecahedron with planar faces.