Algorithms for Drawing Polyhedra from 3-Connected Planar Graphs

Alen Orbanić, Marko Boben, Gašper Jaklič and Tomaž Pisanski
IMFM, OTR,
Jadranska 19,
Ljubljana, Slovenia
Emails: *Alen.Orbanic@fmf.uni-lj.si*, *Marko.Boben@fmf.uni-lj.si*,
Gasper.Jaklic@fmf.uni-lj.si, *Tomaz.Pisanski@fmf.uni-lj.si*

Keywords: graph drawing, Tutte's drawing method, representation of a graph, polyhedral representations

Received: September 15, 2003

Two algorithms for producing polyhedral representations for 3-connected planar graphs are discussed in the paper. One of them uses Tutte's drawing algorithm [11] to produce a 2D drawing. Then the drawing is lifted into 3D space obtaining a polyhedral embedding. The other is a simple algorithm by G. Hart [4] for drawing canonical polyhedral representations. Some alternative aspects (physical model, Markov chain model) in algorithms for obtaining Tutte's drawings are presented and proved.

Povzetek: članek opisuje dva pristopa za grafičen prikaz planarnih grafov.

1 Introduction

A convex polyhedron can be viewed as a convex hull of its vertices and referred as $P = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$, $\mathbf{p}_i \in \mathbb{R}^3$, or as an intersection of the half-spaces defined by the supporting planes of the faces using the side of $\mathbb{R}^3$ that contains the polyhedron. These are two dual definitions. For more detailed and generalized definitions of polyhedra see [12].

Any graph mentioned in the article is 3-connected planar. Vertices and edges of a polyhedron P define a skeleton graph $G(P)$ in an obvious way. The skeleton graph is obviously planar. By Balinski's theorem [12] this graph is 3-connected. In 1922 Steinitz proved that 3-connected planar graphs are exactly the skeletons of the convex 3D polyhedra. According to Whitney [2], every 3-connected planar graph has a unique embedding in the plane and the faces of the embedding are exactly the non-separating induced cycles. These faces are exactly the faces of any polyhedron with the same skeleton graph.

Given a 3-connected graph G one would like to have an algorithm to obtain a polyhedron with its skeleton G. We will call such a representation a *polyhedral representation*. A graph is given as a combinatorial structure and our algorithms return 3D coordinates of the vertices of the polyhedron. There are infinitely many polyhedral representations for a given graph. According to Koebe [5], for each polyhedron there is the canonical form, which is determined up to a rotation in 3D space. The canonical representation is especially "nice" because it possesses maximal possible geometric symmetry. Each edge of such a polyhedron touches the unit sphere in exactly one point and the center of the gravity of these touching points is the origin. If a polyhedron P is in canonical form, then the polar polyhedron is also in canonical form and the dual edges have the same touching points with the unit sphere. For a given polyhedron P containing the origin of $\mathbb{R}^3$ a polar polyhedron P^* is well defined and unique:

$$P^* = \{\mathbf{y} \in \mathbb{R}^3 \mid \langle \mathbf{x}, \mathbf{y} \rangle \leq 1 \text{ for all } \mathbf{x} \in P\}.$$

A skeleton graph of P^*, $G^* := G(P^*)$ is exactly the dual graph of the skeleton graph $G := G(P)$.

We will present some methods together with references for more detailed information. [1]

2 Tutte's algorithm

The proofs and detailed descriptions can be found in [10] and [11].

In 1963 Tutte [11] invented an interesting rubber-band method for embedding a 3-connected planar graph G into the plane. By this method one face C (on vertices $k+1, \ldots, n$, referred as the *outer cycle*) of a graph is embedded into the plane as a fixed strictly convex polygon, while the other adjacent vertices are connected with rubber-bands. On the edge ij, where i, j not both in C, there is a strictly positive stretching coefficient ω_{ij}. Later we will also assign the weights ω_{ij} to edges $ij \in C$, but we will not use them in the Tutte's method. Let $\mathbf{p}_i$ represent a position of the vertex i in the plane. The positions $\mathbf{p_i} := \mathbf{p}_i^0$, $i = k+1, \ldots, n$ are fixed. The other positions are to be calculated and lie somewhere inside the polygon. The system consisting of a rigid polygon as a frame and rubber-bands is in *equilibrium* if the following system of equations holds:

$$\sum_{ij \in E(G)} \omega_{ij}(\mathbf{p}_j - \mathbf{p}_i) = 0, \qquad i \in V(G) - V(C), \quad (1)$$

$$\mathbf{p}_i = \mathbf{p}_i^0, \quad i \in V(C).$$

[1] *Supported in part by Ministrstvo za šolstvo, znanost in šport Republike Slovenije, grant J1-6161, J2-6193.*

Tutte proved that this system has a unique solution which gives a set of positions that induces a straight line embedding of G into the plane with convex faces. Instead of solving the system of the linear equations directly, two alternative approaches can be used.

In [8] the following algorithm called *Schlegel diagram* was introduced:

ALGORITHM 1 *Fix the positions of the vertices of the outer cycle C on a convex polygon in $\mathbb{R}^2$. Assign to the other vertices random positions inside the polygon. For those vertices repeat the following steps:*

- *for each vertex i calculate the resulting force $\mathbf{F}_i$ of all adjacent rubber-bands;*
- *move each vertex i for vector $\alpha\mathbf{F}_i$, where $\alpha > 0$ is a fixed real number;*

until the displacements of all vertices are sufficently small (depends on the prescribed precision of the drawing).

We will determine the values of α which guarantee the convergence of Algorithm 1.

A *weighted Laplacian matrix* Q_ω for a graph G on n vertices and weights $\omega = (\omega_{ij})_{ij \in E(G)}$ is a $n \times n$ matrix with $(Q_\omega)_{i,j} = (Q_\omega)_{j,i} = -\omega_{ij}$ when $ij \in E(G)$, $(Q_\omega)_{i,j} = 0$, when $i \neq j$, and $ij \notin E(G)$ and $(Q_\omega)_{i,i} = -\sum_{j=1, j\neq i}^{n} (Q_\omega)_{i,j}$. Let $K = \{1, \ldots, k\}$ and $Q_{\omega,K}$ be a matrix that consists of the first k rows of Q_ω. Let

$$A = \begin{pmatrix} & -Q_{\omega,K} & \\ 0_{k,n-k} & & -I_{n-k} \end{pmatrix}, \qquad (2)$$

where $0_{k,n-k}$ represents a $k \times (n-k)$ zero matrix and I_ℓ a $\ell \times \ell$ identity matrix. Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$, where $\mathbf{p}_i = (x_i, y_i)$. Let $\mathbf{p}_i := \mathbf{p}_i^0$ be fixed values for $i = k+1, \ldots, n$ and $\mathbf{p}_i$ be the pairs of variables for $i = 1, \ldots, k$. Let also $\mathbf{b} = (b_1, \ldots, b_n)$, $\mathbf{c} = (c_1, \ldots, c_n)$ and $(b_i, c_i) = (0, 0)$ for $i = 1, \ldots, k$ and $(b_i, c_i) = -\mathbf{p}_i^0$ for $i = k+1, \ldots, n$. The system of equations (1) can be written as:

$$A\mathbf{x} = \mathbf{b}, \qquad A\mathbf{y} = \mathbf{c}. \qquad (3)$$

The iteration procedure in the Algorithm 1 can be rewritten as:

$$\begin{aligned} \mathbf{x}^{n+1} &= \mathbf{x}^n + \alpha(A\mathbf{x}^n - \mathbf{b}), \qquad (4) \\ \mathbf{y}^{n+1} &= \mathbf{y}^n + \alpha(A\mathbf{y}^n - \mathbf{c}). \end{aligned}$$

Let us determine the bounds for α that ensure that the algorithm converges. It is sufficient to do this for the iteration for $\mathbf{x}$. Let us rewrite the iteration (4):

$$\mathbf{x}^{n+1} = (\alpha A + I)\mathbf{x}^n - \alpha\mathbf{b}. \qquad (5)$$

Let $\rho(M)$ denote a spectral radius of a matrix M: It is well known that an iteration of this type converges iff $\rho(\alpha A + I) < 1$, see [1]. We will show that for a sufficiently small $\alpha > 0$ it follows $\rho(\alpha A + I) < 1$.

If $\sigma(A)$ is the spectrum of the matrix A, then (with slight abuse of notation) $\sigma(\alpha A + I) = 1 + \sigma(\alpha A) = 1 + \alpha \cdot \sigma(A)$. Proving that $\sigma(A)$ is strictly negative would yield the existence of the appropriate α, such that $\rho(\alpha A + I) < 1$. The matrix A is block diagonal and upper triangular with two diagonal blocks $-Q_{\omega,(K,K)}$ and $-I_k$, where $Q_{\omega,(K,K)}$ is a matrix obtained from Q_ω taking the rows and the columns in the set K in induced order. It follows that $\sigma(A) = \sigma(-Q_{\omega,(K,K)}) \cup \sigma(-I_k)$. It suffices to show that $Q_{\omega,(K,K)}$ is positive definite.

The matrix $Q_{\omega,(K,K)}$ is a principal sub-matrix of Q_ω. We will prove that $\sigma(Q_\omega) \geq 0$ and that $Q_{\omega,(K,K)}$ is nonsingular. Cauchy's interlacing theorem [1] implies that $Q_{\omega,(K,K)}$ is also positive definite.

Using the Gershgorin's theorem (see [1]) it is easy to see that the eigenvalues of Laplacian matrix Q_ω are contained in $[0, 2 \cdot \max_{i=1,\ldots,n}\{(Q_\omega)_{i,i}\}]$, hence $\sigma(Q_\omega) \geq 0$. By Whitney's theorem $H = G - C$ is connected. Let $Q' := Q_\omega(H)$ be a weighted Laplacian matrix of H and $Q := Q_{\omega,(K,K)}$. Then: $Q = Q' + \Delta$, where Δ is a nonnegative diagonal matrix. Since G is connected, $\Delta \neq 0$. For an arbitrary vector $\mathbf{w} \in \mathbb{R}^k$:

$$\mathbf{w}^T Q \mathbf{w} = \mathbf{w}^T Q' \mathbf{w} + \mathbf{w}^T \Delta \mathbf{w}.$$

Since eigenvalues of Q' are by Gershgorin's theorem contained in the interval $[0, 2 \cdot \max_{i=1,\ldots,k} \{Q'_{i,i}\}]$, it follows that $\mathbf{w}^T Q' \mathbf{w} \geq 0$. Since H is connected, the multiplicity of the smallest eigenvalue is 1. But the smallest eigenvalue is 0 and its eigenvector is the all ones vector $\mathbf{1}$. Therefore $\mathbf{w}^T Q' \mathbf{w} = 0$ iff $\mathbf{w} = c\mathbf{1}$ for some $c \in \mathbb{R}$. Since Δ is a nonnegative diagonal matrix, it follows that $\mathbf{w}^T \Delta \mathbf{w} \geq 0$. But for $\mathbf{w} = \mathbf{1}$, $\mathbf{w}^T \Delta \mathbf{w} = c^2 \mathbf{1}^T \Delta \mathbf{1} > 0$, if $c \neq 0$. So $\mathbf{w}^T Q \mathbf{w} > 0$ for each $\mathbf{w} \neq 0$ and Q is positive definite. The following theorem holds:

THEOREM 1 *Let G be a 3-connected planar graph. If:*

$$0 < \alpha < \frac{2}{\max\left\{1, 2 \cdot \max_{i=1,\ldots,k} \left\{\sum_{ij \in E(G)} \omega_{ij}\right\}\right\}},$$

then Algorithm 1 converges to the solution of (1).

The other alternative approach comes from probability. Let us define a Markov chain $X_0, X_1, \ldots$ with transition matrix P using the graph G and the weights ω:

$$P_{ij} = \begin{cases} 0 & ij \notin E(G), \\ \frac{\omega_{ij}}{\sum_{ik \in E(G)} \omega_{ik}} & ij \in E(G) - E(C), \\ 1 & ij \in E(C). \end{cases} \qquad (6)$$

After rewriting the system (1) we get:

$$\begin{aligned} \mathbf{p}_i &= \frac{\sum_{ij \in E(G)} \omega_{ij} \cdot \mathbf{p}_j}{\sum_{ij \in E(G)} \omega_{ij}}, & i \in V(G) - V(C), \qquad (7) \\ \mathbf{p}_i &= \mathbf{p}_i^0, & i \in V(C). \end{aligned}$$

or

$$\begin{aligned} &P\mathbf{x} = \mathbf{x}, \\ &P\mathbf{y} = \mathbf{y}, \\ &\mathbf{p}_i = \mathbf{p}_i^0 = (x_i^0, y_i^0), \qquad i \in V(C). \end{aligned} \tag{8}$$

The vertices of the graph G are the states of the Markov chain. Let $A \subset V(G)$. The values:

$$h_i^A = \Pr(X_r \in A \text{ for some } r \geq 0 \mid X_0 = i), \tag{9}$$

are *hitting probabilities* for the set A when starting in the state i. We will write h_i^j for $h_i^{\{j\}}$. The following theorem holds (see [7]):

THEOREM 2 *A vector $\mathbf{h}^A = (h_1^A, \dots, h_n^A)$ is a solution of a system:*

1. *$h_i^A = 1$, for $i \in A$,*
2. *$\sum_{j=1}^n P_{ij} h_j^A = h_i^A$, for $i \notin A$,*
3. *$0 \leq h_i^A \leq 1$ for all i.*

If $\widetilde{\mathbf{h}}^A$ is any other solution then $h_i^A \leq \widetilde{h}_i^A$ (inequality on components). Therefore $\mathbf{h}^A$ is a minimal solution.

Let $\mathbf{h}^j = (h_i^j)_{i=1}^n$ be the vector of hitting probabilities. By Theorem 2, $Ph^j = h^j$. Let $J = [\mathbf{h}^{k+1}, \dots, \mathbf{h}^n]$ be a matrix with vectors $\mathbf{h}^j$ as columns. Hence $PJ = J$.

Let $\mathbf{x}^0 = (x_{k+1}^0, \dots, x_n^0)$ and $\mathbf{y}^0 = (y_{k+1}^0, \dots, y_n^0)$ be the coordinates of vertices on the outer polygon. Then:

$$\begin{aligned} PJ\mathbf{x}^0 &= J\mathbf{x}^0, \\ PJ\mathbf{y}^0 &= J\mathbf{y}^0, \end{aligned} \tag{10}$$

and

$$\begin{aligned} (J\mathbf{x}^0)_j &= x_j^0, \\ (J\mathbf{y}^0)_j &= y_j^0, \quad j = k+1, \dots, n. \end{aligned} \tag{11}$$

Thus $\mathbf{p}_i = ((J\mathbf{x}^0)_i, (J\mathbf{y}^0)_i)$ is a solution of the system (7) and it is unique by Tutte.

The following theorem holds:

THEOREM 3 *Let $G = (\{1, \dots, n\}, E(G))$ be a 3-connected planar graph, $C = (k+1, \dots, n)$ be one of its faces, and $\mathbf{p}_i^0$, $i = k+1, \dots, n$, be the fixed coordinates of the vertices of C forming a convex polygon in the cyclical order of C. Let P be a transition matrix for the Markov chain (X_r) and let h_i^j be the hitting probabilities for entering into vertices $j = k+1, \dots, n$, starting from the vertex $i \in V(G)$. Then the Tutte's drawing can be obtained in the following way:*

$$\begin{aligned} \mathbf{p}_i &= \sum_{j=k+1}^{n} h_i^j \mathbf{p}_j^0, & i \in V(G) - V(C), \\ \mathbf{p}_i &= \mathbf{p}_i^0, & i \in V(C). \end{aligned}$$

The theorem was originally conjectured by T. W. Tucker.

To obtain the hitting probabilities h_i^j one can make sufficently large number of the following experiments: start in i and make a series of steps until a state in C is reached. The frequency of the walks ending in j is an approximation for h_i^j.

It can be easily verified that

$$\lim_{n \to \infty} (P^n)_{ij} = h_i^j, \quad i \notin C, j \in C.$$

Hence the hitting probabilities can be obtained by calculating P^r, r large.

3 Lifting of a Tutte's Drawing

A detailed procedure with proofs can be found in [10].

A weighted embedded 3-connected planar graph G with weights ω and positions $\mathbf{p}_1, \dots, \mathbf{p}_n$ of the vertices is in equilibrium if:

$$\sum_{ij \in E(G)} \omega_{ij}(\mathbf{p}_j - \mathbf{p}_i) = 0, \quad \text{for all } i \in V(G). \tag{12}$$

The corresponding weight ω is called an *equilibrium weight*. Maxwell-Cremona's theorem states that if for an embedded 3-connected graph with convex faces there exists an equilibrium weight with strictly negative weights on the edges of the outer face and strictly positive weights on the inner edges, then the drawing can be lifted to a polyhedron. A Tutte's drawing with the corresponding weights on the edges that are not in the outer face has all the vertices not on the outer face in an equilibrium. To use a Maxwell-Cremona's theorem there should be negative equilibrium weights on the edges of the outer face. A simplified version of the theorem will be used to solve that problem.

PROPOSITION 4 *Let G be a 3-connected planar graph embedded with one outer face f_1 on a convex polygon with all other faces ($f_2, \dots, f_m$) as convex polygons inside the outer polygon. Let G^* be the geometric dual of G. Let $e = ij \in E(G)$ and let $e^* = fg$ be the corresponding dual edge, such that if one traverses from i to j in the embedding of G, the face f is on the left side. Let (i, j, f, g) denote an* oriented patch *and let $\mathbf{q}_{f_1} = (0, 0, 0)$. Then for $2 \leq k \leq m$ the unique assignments $\mathbf{q}_{f_k}$ exist, such that:*

$$\mathbf{q}_f - \mathbf{q}_g = \omega_{ij}(\mathbf{p}_i \times \mathbf{p}_j),$$

where ω is an equilibrium weight and × denotes a cross product.

For the proof see [10].

Using vectors $\mathbf{q}$ one can define a function on the interior of the outer polygon:

$$z(\mathbf{x}) = \langle \mathbf{x}, \mathbf{q}_{f_i} \rangle, \quad \text{for } \mathbf{x} \in f_i \subset \mathbb{R}^2. \tag{13}$$

It can be proved that the function $z(\mathbf{x})$ is linear, continuous, convex, and the image of each face lies on some plane. It can be seen that if we have a Tutte's drawing with a triangle as the outer face, then using (12) on the Tutte's drawing, one can calculate the remaining negative weights on the edges of the triangle. The surface defined by the graph of the function $z(\mathbf{x})$ together with a patch in the place of the triangle determines a hull of the polyhedral representation of G. Using the fact that if G is 3-connected planar then either G or G^* has a triangle as a face (see [6]). This implies the following algorithm:

ALGORITHM 2 *1. Determine the faces of G (some planarity algorithm).*

2. If one of the faces is a triangle, use G otherwise use G^*.

3. Draw a Tutte's drawing.

4. From the Tutte's drawing and using Proposition 4 determine vectors $\mathbf{q}$.

5. Determine the vertices of the polyhedron using the function $z(\mathbf{x})$.

6. If G was used, we have a polyhedral embedding. Otherwise move the polyhedron so that the center of the gravity of the vertices lies in the origin. Calculate the polar polyhedron.

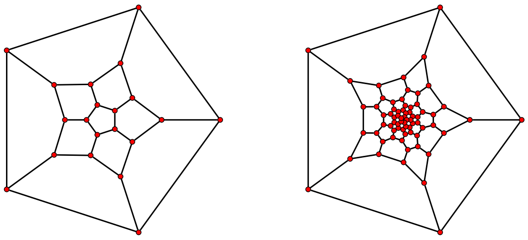

Figure 1: Dodecahedron and Fullerene C60 drawn by Tutte's method.

4 Canonical polyhedra

To produce the canonical polyhedral representation one can use methods for determining primal-dual circle packing (PDCP) of 3-connected planar graph. Having PDCP one can easily obtain the canonical polyhedral representation using the inverse of the stereographic projection on the unit sphere. One of the algorithms is due to Mohar [6] (circle packing in the plane or on the sphere). The other, which works in 3D and on more or less small graphs, is due to G. Hart [4]. We present some slight improvements for the latter algorithm. The Hart's algorithm reads:

ALGORITHM 3 *1. Start with a "good approximation" of a polyhedron that has edges as near as possible to the unit sphere. For instance, get the representation from the Algorithm 2, move the center of gravity into the origin, and project vertices from the origin to the unit sphere.*

2. Repeat the following procedure until the positions of the vertices are precise enough:

(a) for each edge e calculate the point $\mathbf{p}_e$, which is the closest to the origin. If the edge is not at the distance 1 to the origin, move the endpoints of e for a vector $\alpha(1 - \|\mathbf{p}_e\|)\mathbf{p}_e$, where $0 < \alpha < 1$.

(b) for each face calculate the approximate plane of the vertices on the face. If some vertex v is at the distance $d(v)$ from the plane, move v towards the plane in the direction of the normal for a magnitude $\beta d(v)$, $0 < \beta < 1$.

Algorithm 3 works well on small graphs. It seems to work well on cubic graphs and perhaps on graphs with lower degrees of the vertices, but the convergence might be poor, especially on the larger graphs. It behaves very poorly (does not even converge) if degrees are rather high (≥ 4). More or less we tested the performance on cubic graphs. If the cubic graph is large the convergence is very poor. To improve the convergence on cubic graphs we first use Algorithm 3 for a few iterations. Then at each iteration we calculate the vertices of the approximate polar (from approximate faces) and the dual edges. Then we check if the current dual edges are perpendicular to the original edges. If not, the vertices are moved in a manner of rotation for some small magnitude that depends on the difference between the current angle and $\pi/2$. This brings a small improvement to the convergence of the algorithm.

Both methods significantly depend on α, β and similar parameters in the improved algorithm. Unfortunately we were not able to prove the convergence for any set of the parameters.

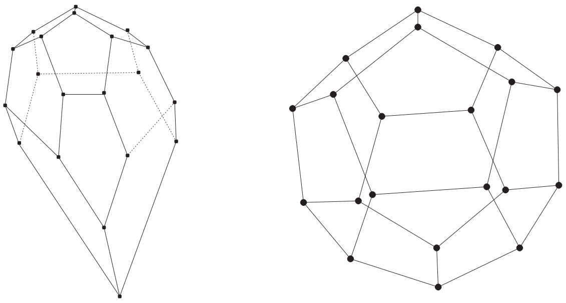

Figure 2: A dodecahedron obtained by lifting Tutte's drawing of its dual graph (left) and a dodecahedron in the canonical form (right).

References

[1] J. W. Demmel, *Applied numerical linear algebra*, SIAM Society for Industrial and Applied Mathematics, 1997.

[2] R. Diestel, *Graph Theory*, Springer, 1997.

[3] C. D. Godsil, G. F. Royle, *Algebraic Graph Theory*, Graduate Texts in Mathematics, 207, Springer-Verlag, 2001.

[4] G. W. Hart, *Calculating Canonical Polyhedra*, Mathematica in Education and Research, Vol 6 No. 3, Summer 1997, pp. 5–10.

[5] P. Koebe, *Kontaktprobleme der Konformen Abbildung*, Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl., **88** (1936), 141–164.

[6] B. Mohar, *A polynomial time circle packing algorithm*, Discrete Math. **117** (1993), 257–263.

[7] J. Norris, Markov Chains, Cambridge University Press, 1999.

[8] T. Pisanski, B. Plestenjak, A. Graovac: *NiceGraph Program and its Applications In Chemistry*, Croat. Chem. Acta **68** (1995), 283–292.

[9] T. Pisanski et al., VEGA *program*, `http://vega.ijp.si/Htmldoc/Vega03.html`

[10] J. Richter-Gebert, *Realization Spaces of Polytopes*, Lecture Notes in Mathematics, Vol. 1643, Springer-Verlag, Berlin, 1996.

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

[12] G. M. Ziegler, *Lectures on Polytopes*, Graduate Texts in Mathematics 152, Springer-Verlag, New York 1995, Revised edition 1998.