Regular Edge Labelings and Drawings of Planar Graphs

Xin He[1] and Ming-Yang Kao[2]

[1] Department of Computer Science, State University of New York at Buffalo, Buffalo, NY 14260. Research supported in part by NSF grant CCR-9205982.
[2] Department of Computer Science, Duke University, Durham, NC 27706. Research supported in part by NSF grant CCR-9101385.

Abstract. The problems of nicely drawing planar graphs have received increasing attention due to their broad applications [5]. A technique, *regular edge labeling*, was successfully used in solving several planar graph drawing problems, including *visibility representation, straight-line embedding*, and *rectangular dual* problems. A regular edge labeling of a plane graph G labels the edges of G so that the edge labels around any vertex show certain regular pattern. The drawing of G is obtained by using the combinatorial structures resulting from the edge labeling. In this paper, we survey these drawing algorithms and discuss some open problems.

1 Visibility Representation

Given a planar graph $G = (V, E)$, a *visibility representation* (VR) of G maps each vertex of G into a horizontal line segment and each edge into a vertical line segment that only touches the two horizontal line segments representing its end vertices. This representation has been used in several applications for representing electrical diagrams and schemes [28]. Linear time algorithms for constructing a VR has been independently discovered by Rosenstiehl and Tarjan [20] and Tamassia and Tollis [29]. Their algorithms are based on:

Definition 1: Let G be a connected plane graph and s, t be two vertices on the exterior face of G. An *st-labeling* of G is an orientation of edges such that:
1. All edges incident to s are leaving s; all edges incident to t are entering t.
2. For each vertex $v \neq s, t$, the edges incident to v are partitioned into two subsets, each of which is consecutive around v in the embedding. The edges in the first subset are leaving v; the edges in the second subset are entering v.

Such a labeling is also called an *st-numbering*, *st-orientation*, or *st-planar graph*. Its properties has been extensively studied [18, 20, 25, 26]. A VR of G can be obtained from an *st*-labeling by the following linear time algorithm [20].

Algorithm 1: Visibility Representation

1. Construct an *st*-labeling of G. Let $\bar{G}$ be the resulting directed graph and G^* the directed dual.
2. For each vertex v, compute $d(v)$, the length of the longest path from the unique source s of $\bar{G}$ to v in $\bar{G}$.

3. For each face F of G, compute $d^*(F)$, the length of the longest path from the unique source of G^* to F in G^*.
4. For each vertex v of G Do:
   If $v \neq s, t$, draw horizontal line between $(d^*(\mathit{left}(v)), d(v))$ and $(d^*(\mathit{right}(v)) - 1, d(v))$. ($\mathit{left}(v)$ denotes the face incident to v that separates the edges entering v and the edges leaving v in clockwise direction. $\mathit{right}(v)$ is defined analogously.)
   If $v = s$ or t, draw horizontal line between $(0, d(v))$ and $(D, d(v))$, where D is the length of the longest path between the source and the sink in G^*.
5. For each edge (u, v) of G do:
   Draw vertical line between $(d^*(\mathit{left}(u, v)), d(u))$ and $(d^*(\mathit{left}(u, v)), d(v))$. ($\mathit{left}(e)$ denotes the face on the left of the edge e.)

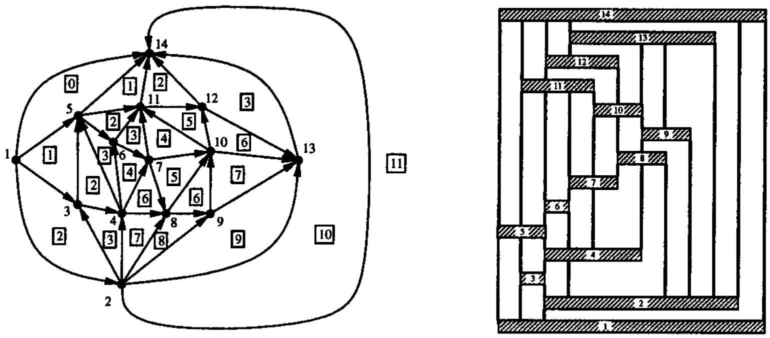

Fig. 1. A st-labeling of a plane graph and its visibility representation.

Fig 1 shows an example of this algorithm. The figure on the left is a graph G with $s = 1$ and $t = 14$. The numbers in small boxes denote the d^*-value of the faces. The figure on the right is the resulting VR. It is easy to show the size of the drawing is at most $(2n - 5) \times (n - 2)$. By using a carefully constructed st-labeling, the drawing size can be reduced to $(n - 1) \times (n - 1)$ for 4-connected planar graphs [15], and to $(\lfloor \frac{3}{2}n \rfloor - 3) \times (n - 1)$ for general planar graphs [14].

2 Straight-Line Grid Embedding

A *straight-line grid embedding* of a planar graph G is a drawing where the vertices are located at grid points, and each edge is represented by a straight line segment. Such embeddings on reasonably small grids are very useful in visualizing planar graphs on graphic screens and have wide applications in CAD/CAM and computer graphics [5]. Wagner [31], Fáry [8], and Stein [24] showed that every planar graph has a straight-line embedding. Many embedding algorithms have been reported [3, 19, 30]. However, these algorithms all suffer two serious

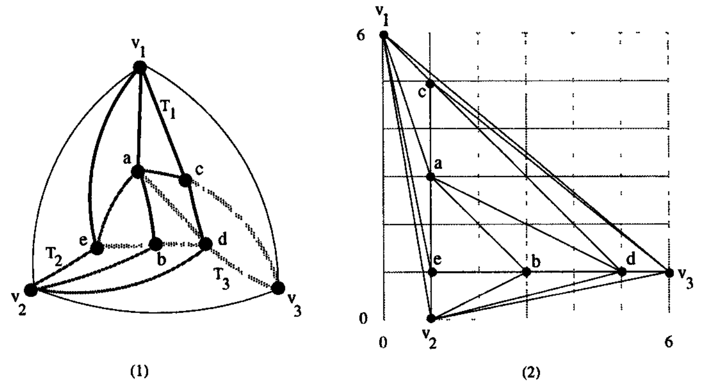

Fig. 2. A realizer of a plane graph and its straight-line grid drawing.

drawbacks. First, they require high-precision real arithmetic relative to the size of the graph. Second, in the drawings produced by them, the ratio of the smallest distance to the largest distance between vertices are often so small that it is very difficult to view those drawings on a graphic screen.

In view of these drawbacks, Rosenstiehl and Tarjan [20] posed the problem of computing a straight-line embedding on a grid of polynomial size. Fraysseix et. al. showed that a straight-line embedding on a grid of size $(2n-4) \times (n-2)$ can be computed in $O(n \log n)$ time [6, 7], which was improved to $O(n)$ in [4]. By using regular edge labeling, which he calls *realizer*, Schnyder proved the existence of an embedding on a smaller grid of size $(n-2) \times (n-2)$ and gave an elegant linear time embedding algorithm [21, 22, 23]. Schnyder's algorithm can be implemented in parallel in $O(\log n \log\log n)$ time with optimally many processors [9]. For this problem, without loss of generality, we can consider only plane triangulated graphs. Schnyder's algorithm is based on the following concept [21, 22]:

Definition 2: Let G be a plane triangulated graph with three exterior vertices v_1, v_2, v_3 in clockwise order. A *realizer* of G is a partition of interior edges into three sets T_1, T_2, T_3 and an orientation of interior edges such that:
1. For $i = 1, 2, 3$, all interior edges incident to v_i are in T_i and entering v_i.
2. For each interior vertex u of G, the edges incident to u appear around u clockwise in the following pattern:
 $*$ one edge in T_1 leaves u; a set (maybe empty) of edges in T_3 enters u;
 $*$ one edge in T_2 leaves u; a set (maybe empty) of edges in T_1 enters u;
 $*$ one edge in T_3 leaves u; a set (maybe empty) of edges in T_2 enters u.

A plane triangulated graph G and a realizer of G is shown in Fig 2(1). It is shown in [23] that every plane triangulated graph has a realizer, and for each $i \in \{1, 2, 3\}$, T_i forms a tree rooted at v_i consisting of all interior vertices and one exterior vertex v_i. For each $i \in \{1, 2, 3\}$ and for each vertex u in T_i:

- let $P_i(u)$ be the tree path in T_i from u to the root v_i of T_i;
- let $p_i(u)$ be the number of vertices in the path $P_i(u)$;

- for each interior vertex u, let $r_i(u)$ be the number of vertices (including the vertices on the boundary) in the triangular region bounded by the paths $P_{i-1}(u)$, $P_{i+1}(u)$, and the exterior edge (v_{i-1}, v_{i+1}).

From a realizer of G, the values defined above can be computed in linear time. A straight-line grid embedding of G can be obtained by using the following:

Theorem [23]: A straight-line embedding of G on a $(n-2) \times (n-2)$ grid is given by assigning each interior vertex u to the grid point $(r_3(u) - p_2(u), r_1(u) - p_3(u))$ and assigning v_1, v_2, v_3 to $(0, n-2), (1, 0), (n-2, 1)$, respectively.

Fig 2 shows an example embedding obtained from this theorem.

3 Rectangular Dual

Let R be a rectangle. A *rectangular subdivision system* of R is a partition of R into a set $\Phi = \{R_1, R_2, \ldots, R_n\}$ of non-intersecting smaller rectangles such that no four rectangles in Φ meet at the same point. A *rectangular dual* of a graph $G = (V, E)$ is a rectangular subdivision system Φ and a one-to-one mapping $f : V \rightarrow \Phi$ such that two vertices u and v are adjacent in G iff their corresponding rectangles $f(u)$ and $f(v)$ share a common boundary. Fig 3(1) and 3(2) show a graph G and a rectangular dual of G. If G has a rectangular dual, it is clear that G must be planar and all its interior faces must be triangles.

The rectangular dual finds applications in the floor planning of electronic chips and in architectural design [12]. The rectangular dual is related to the *tessellation representation* of plane graphs [26, 27], which maps the vertices, edges and faces of G to the rectangles of the plane such that the incidence relations of G correspond to the geometric adjacencies between the rectangles.

The problem of finding rectangular duals has been studied in [1, 2, 16, 17]. A linear time algorithm was given in [2]. This algorithm is complicated and requires real arithmetic for the coordinates of the rectangular dual.

Consider a plane graph $H = (V, E)$. We seek a rectangular dual of H. To simplify the problem, we modify H as follows: Add four new vertices v_N, v_W, v_S, v_E and connect each of them to a subpath on the exterior face of H. Then add four new edges $(v_N, v_W), (v_W, v_S), (v_S, v_E), (v_E, v_N)$. Let G be the resulting graph (see Fig 3(1)). Clearly H has a rectangular dual iff G has a rectangular dual R with exactly four rectangles on the boundary of R. Without loss of generality, we will only discuss plane graphs with triangular interior faces and exactly four vertices on the exterior face. In [1, 2, 16], it was shown that such a graph G has a rectangular dual iff G has no *separating triangles*. We will call these graphs *proper triangulated plane* (PTP) graphs. By using regular edge labeling, a simple linear time algorithm for constructing rectangular dual was found in [10, 15]. The coordinates of the rectangular dual constructed are integers and closely related to the structure of the graph. This algorithm can also be implemented on PRAM in $O(\log^2 n)$ time with $O(n)$ processors [11]. It was shown in [15] that this technique is related to the *lmc*-ordering of planar graphs, which is very useful in solving planar graph drawing problems [13].

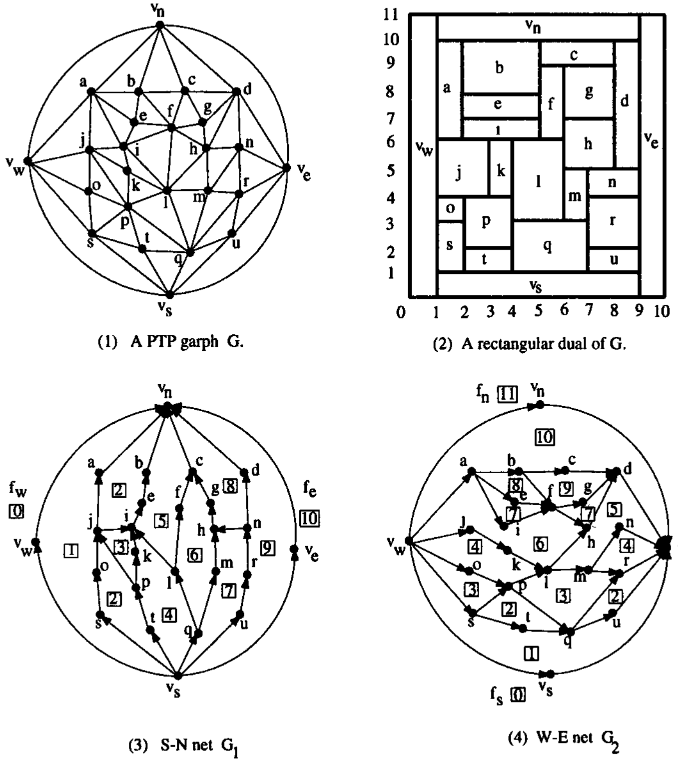

Fig. 3. A PTP graph G and a rectangular dual of G.

Definition 3: A *regular edge labeling* (REL) of a PTP graph $G = (V, E)$ is a partition of the interior edges of G into two subsets $\{T_1, T_2\}$ and an orientation of the interior edges of G such that:

1. For each interior vertex v, the edges incident to v appear in counterclockwise order around v as follows: a set of edges in T_1 leaving v; a set of edges in T_2 entering v; a set of edges in T_1 entering v; a set of edges in T_2 leaving v.
2. All interior edges incident to v_N are in T_1 and enter v_N. All interior edges incident to v_W are in T_2 and leave v_W. All interior edges incident to v_S are in T_1 and leave v_S. All interior edges incident to v_E are in T_2 and enter v_E.

Let $G = (V, E)$ be a PTP graph and $\{T_1, T_2\}$ be a REL of G. Let G_1 be the directed subgraph of G induced by T_1 and the four exterior edges directed as $v_S \longrightarrow v_W$; $v_W \longrightarrow v_N$; $v_S \longrightarrow v_E$; $v_E \longrightarrow v_N$. Let G_2 be the directed subgraph of G analogously defined for T_2. We will call G_1 the *S-N net* and G_2 the *W-E net* of G derived from the REL $\{T_1, T_2\}$. Fig 3(1) shows a PTP graph G. An S-N net G_1 and the corresponding W-E net G_2 are shown in Figs 3(3) and 3(4). Both G_1 and G_2 are *s-t* planar graphs [10].

Consider the S-N net G_1 of G. For each edge e of G_1, let $left(e)$ ($right(e)$, resp.) denote the face of G_1 on the left (right, resp.) of e. Define the ***dual graph***, denoted by G_1^*, of G_1 as follows. The node set of G_1^* is the set of the interior faces of G_1 plus two exterior faces f_W and f_E. For each edge e of G_1, there is a corresponding arc e^* in G_1^* directed from the face $left(e)$ to the face $right(e)$. Since G_1 is an s-t planar graph, G_1^* is also an s-t planar graph. Namely G_1^* is a directed acyclic plane graph with f_W as the only source and f_E as the only sink. The dual graph G_2^* of G_2 is defined analogously. For each face f of G_1, let $F_1(f)$ be the length of the longest path in G_1^* from f_W to f (with $F_1(f_W) = 0$). For each face g of G_2, let $F_2(g)$ be the length of the longest path in G_2^* from f_S to g (with $F_2(f_S) = 0$). The following linear time algorithm computes a $k_1 \times k_2$ rectangular dual of G [10, 15]. An example of this algorithm is shown in Fig 3.

Algorithm 2: DUAL (Input: A PTP graph $G = (V, E)$)
(1) Find a REL $\{T_1, T_2\}$ of G.
(2a) Construct the S-N net G_1 derived from $\{T_1, T_2\}$ and its dual graph G_1^*.
(2b) Compute the function $F_1(f)$ for G_1^*. Let $k_1 = F_1(f_E)$.
(2c) For each vertex $v \in V$ other than v_S and v_N, let $f_1 = left(v)$ and $f_2 = right(v)$ in G_1. Let $x_1(v) = F_1(f_1)$ and $x_2(v) = F_1(f_2)$. Define $x_1(v_N) = x_1(v_S) = 1$ and $x_2(v_N) = x_2(v_S) = k_1 - 1$.
(3a) Construct the W-E net G_2 derived from $\{T_1, T_2\}$ and its dual graph G_2^*.
(3b) Compute the function $F_2(g)$ for G_2^*. Let $k_2 = F_2(f_N)$.
(3c) For each vertex $v \in V$, let $g_1 = below(v)$ and $g_2 = above(v)$ in G_2. Let $y_1(v) = F_2(g_1)$ and $y_2(v) = F_2(g_2)$.
(4) For each vertex $v \in V$, assign v a rectangle $R(v)$ bounded by x-coordinates $x_1(v)$, $x_2(v)$ and y-coordinates $y_1(v)$, $y_2(v)$.

4 Open Problems

For each problem discussed above, there is a corresponding optimization problem, which is usually hard to solve. As we have seen, the algorithms that are based on regular edge labeling techniques closely relate the drawing of the graph to the combinatorial properties of the graph. It is hopeful that this technique might be useful in solving these optimization problems.

For example, consider the visibility representation problem. We want to find a representation R of a given planar graph G such that the height $h(R)$, or the width $w(R)$, or the area $h(R) \times w(R)$ of R is minimized. This problem was first posed in [20]. To simplify the discussion, let us consider the simpler problem: Find a representation R with minimum height $h(R)$. From Algorithm 1, it is easy to see the problem is equivalent to:

Problem 1: Given a plane graph $G = (V, E)$ and two vertices s, t on the exterior face of G, find an st-labeling of G such that the length of the longest path from s to t in the resulting directed graph $\bar{G}$ is minimized.

To our knowledge, this problem is not known to be in P nor NP-complete.

We can also consider the problem from a different angle. Let G be a plane graph and G^* be its dual. Any st-orientation $\bar{G}$ of G induces an st-orientation

$\vec{G}^*$ of G^*. $\vec{G}^*$ is an acyclic directed graph with s^* as the only source and t^* as the only think. We can assign each edge of $\vec{G}^*$ a flow value such that each edge of $\vec{G}^*$ has at least one unit flow and the flow conservation requirement is satisfied. This results in an $s^* - t^*$ flow in $\vec{G}^*$. Under this correspondence, it is easy to show that the length of the longest path from s to t in $\vec{G}$ is equal to the total flow from s^* to t^* in $\vec{G}^*$. The problem 1 then is equivalent to the following:

Problem 1': Given an *undirected* plane graph G^* and two vertices s^* and t^* on the exterior face, find an $s^* - t^*$ flow in G^* such that the following hold: (a) There is no circulation. (b) Each edge of G^* has at least one unit flow. (c) The total flow value from s^* to t^* is *minimized.*

Although this flow problem has a "natural looking", no polynomial time algorithm for it is known. The reason is that we are dealing with an undirected graph with lower bounds on edge capacity. The traditional techniques for solving flow problems do not work here. It is intersting to see if Problem 1 and Problem 1' are in P or NP-complete. If they are indeed NP-complete, it is also interesting to find approximation algorithms for solving them.

For the rectangular dual problem, we can ask similar questions: How to find a rectangular dual R of a given graph G such that $h(R)$, or $w(R)$, or $h(R) \times w(R)$ is minimized? The practical importance of these optimization problems is obvious. Their combinatorial structure is very similar to that of VR problem. The study of the properties of the regular edge labeling might be useful in either finding polynomial time algorithms or approximation algorithms for solving them.

We have discussed three planar graph drawing problems. Although they appear very different, it is interesting to note that the algorithms for solving them are remarkably similar: They all require to label the edge set of the input graph G such that certain regular properties around each vertex of G are satisfied. Then the drawing problem is easily solved by using the properties of the resulting combinatorial structures. It is interesting to see if there are other regular edge labelings that can be used to solve interesting planar graph drawing problems.

References

1. J. Bhasker and S. Sahni, A linear algorithm to check for the existence of a rectangular dual of a planar triangulated graph, Networks 17, 1987, pp. 307-317.
2. J. Bhasker and S. Sahni, A linear algorithm to find a rectangular dual of a planar triangulated graph, *Algorithmica* 3, 1988, pp. 247–278.
3. N. Chiba, T. Yamanouchi, and T. Nishizeki, Linear algorithms for convex drawings of planar graphs, in *Progress in Graph Theory,* 1982, pp. 153–173.
4. M. Chrobak and T. H. Payne, A linear time algorithm for drawing planar graphs on a grid, TR. UCR-CS-90-2, Dept. of Math. and CS., Univ. of Calif. at Riverside.
5. P. Eades and R. Tamassia, Algorithms for automatic graph drawing: an annotated bibliography, TR., Dept. of Comp. Sci., Brown Univ., 1993.
6. H. de Fraysseix, J. Pach and R. Pollack, How to draw a planar graph on a grid, *Combinatorica* 10 (1990), pp. 41–51.
7. H. de Fraysseix, J. Pach and R. Pollack, Small sets supporting Fáry embeddings of planar graphs, Proc. of the 20th ACM STOC, 1988, pp. 426–433.

8. I. Fáry, On straight line representation of planar graphs, Acta Sci Math Szeged, 11 (1948), pp. 229–233.
9. M. Fürer, X. He, M. Y. Kao, and B. Raghavachari, $O(n \log \log n)$-work parallel algorithms for straight-line grid embeddings of planar graphs, Proc. of 4th ACM SPAA, 1992, pp. 410–419. Improved version to appear in SIAM J. Discrete Math.
10. X. He, On finding the rectangular duals of planar triangulated graphs, *SIAM J. Comput.* 22 (6), 1993, pp. 1218–1226.
11. X. He, An efficient parallel algorithm for finding rectangular duals of planar triangular graphs, *Proc. 3rd International Conference for Young Computer Scientists*, Beijing, 1993, pp. 6.33–6.37. Complete version to appear in *Algorithmica.*
12. W. R. Heller, G. Sorkin and K. Mailing, The planar package planner for system designers, *Proc. 19th IEEE Design Automation Conf.*, 1982, pp. 253–260.
13. G. Kant, Drawing planar graphs using the *lmc*-ordering, in *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science*, Pittsburgh, 1992, pp. 101-110.
14. G. Kant, A more compact visibility representation, in: J. van Leeuwen (Ed.), *Proc. 19th Intern. Workshop on Graph-Theoretic Concepts in Comp. Science (WG'93).*
15. G. Kant and X. He, Two algorithms for finding rectangular duals of planar graphs, in: J. van Leeuwen (Ed.), *Proc. 19th Intern. Workshop on Graph-Theoretic Concepts in Comp. Science (WG'93)*, *LNCS*, Springer-Verlag, 1994, to appear.
16. K. Koźmiński and E. Kinnen, Rectangular dual of planar graphs, *Networks* 15, 1985, pp. 145–157.
17. Y.-T. Lai and S. M. Leinwand, A theory of rectangular dual graphs, *Algorithmica* 5, 1990, pp. 467–483.
18. F. P. Preparata and R. Tamassia, Fully dynamic techniques for point location and transitive closure in planar structures, *Proc. 29th FOCS*, 1988, pp. 558-567.
19. R. C. Read, A new method for drawing a planar graph given the cyclic order of the edges at each vertex, *Congressus Numerantium* 56 (1987), pp. 31–44.
20. P. Rosenstiehl and R. E. Tarjan, Rectilinear planar layouts and bipolar orientations of planar graphs, *Discr. and Comp. Geometry* 1, 1986, pp. 343–353.
21. W. Schnyder, Embedding planar graphs on the grid, Abstracts of the Amer. Math. Soc., 9 (1988), p. 268.
22. W. Schnyder, Planar graphs and poset dimension, *Orders* 5 (1989), pp. 323–343.
23. W. Schnyder, Embedding planar graphs on the grid, in *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms*, 1990, pp. 138–148.
24. S. K. Stein, Convex maps, in *Proc Amer Math Soc*, vol. 2, 1951, pp. 464–466.
25. R. Tamassia and J. S. Vitter, Parallel transitive closure and point location in planar structures, *SIAM J. Comput.* 20 (2), 1991, pp. 708-725.
26. R. Tamassia, Drawing Algorithms for Planar *s-t* Graphs, *Australasian Journal of Combinatorics*, Vol. 2, 1990, pp. 217-236.
27. R. Tamassia and I. Tollis, Tessellation Representations of Planar Graphs, in *Proc. 27th Allerton Conference*, 1989, pp. 48-57.
28. R. Tamassia and I. G. Tollis, A unified approach to visibility representations of planar graphs, *Discr. and Comp. Geometry* 1, 1986, pp. 321–341.
29. R. Tamassia and I. G. Tollis, Planar grid embedding in linear time, *IEEE Trans. Circuits and Systems* 36, 1989, pp. 1230–1234.
30. W. Tutte, How to draw a graph, *Proc. London Math. Soc.* 13, 1963, pp. 743–768.
31. K. Wagner, Bemerkungen zum Vierfarbenproblem, Jahresbericht Deutsch Math-Verein, 46 (1936), pp. 26–32.