2.5.2 The Shift-Algorithm

Let $(v_1, \ldots, v_n)$ be a canonical ordering. The general plan is to construct an embedding by inserting vertices in this order, starting from the triangle $P(v_1) = (0, 0)$, $P(v_3) = (1, 1)$, $P(v_2) = (2, 0)$; see Figure 2.20.

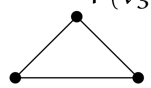

Figure 2.20: *Initialization of the shift algorithm.*

At each step, some vertices will be shifted to the right, making room for the edges to the freshly inserted vertex. For each vertex v_i already embedded, maintain a set $L(v_i)$ of vertices that move rigidly together with v_i. Initially $L(v_i) = \{v_i\}$, for $1 \leqslant i \leqslant 3$.

Ensure that the following invariants hold after Step k (that is, after v_k has been inserted):

(i) $P(v_1) = (0, 0)$, $P(v_2) = (2k - 4, 0)$;

(ii) The x-coordinates of the points on $C_\circ(G_k) = (w_1, \ldots, w_t)$, where $w_1 = v_1$ and $w_t = v_2$, are strictly increasing (in this order)[5];

(iii) each edge of $C_\circ(G_k)$ is drawn as a straight-line segment with slope ± 1.

Clearly these invariants hold for G_3, embedded as described above. Invariant (i) implies that after Step n we have $P(v_2) = (2n - 4, 0)$, while (iii) implies that the Manhattan distance[6] between any two points on $C_\circ(G_k)$ is even.

Idea: put v_{k+1} at $\mu(w_p, w_q)$, where $w_p, \ldots, w_q$ are its neighbors on $C_\circ(G_k)$ and

$$\mu((x_p, y_p), (x_q, y_q)) = \frac{1}{2}(x_p - y_p + x_q + y_q, -x_p + y_p + x_q + y_q)$$

is the point of intersection between the line $\ell_1 : y = x - x_p + y_p$ of slope 1 through $w_p = (x_p, y_p)$ and the line $\ell_2 : y = x_q - x + y_q$ of slope -1 through $w_q = (x_q, y_q)$.

Proposition 2.51. *If the Manhattan distance between w_p and w_q is even, then $\mu(w_p, w_q)$ is on the integer grid.*

Proof. By Invariant (ii) we know that $x_p < x_q$. Suppose without loss of generality that $y_p \leqslant y_q$. The Manhattan distance d of w_p and w_q is $x_q - x_p + y_q - y_p$, which by

[5] The notation is a bit sloppy here because both t and the w_i in general depend on k. So in principle we should write w_i^k instead of w_i. But as the k would just make a constant appearance throughout, we omit it to avoid index clutter.

[6] The *Manhattan distance* of two points (x_1, y_1) and (x_2, y_2) is $|x_2 - x_1| + |y_2 - y_1|$.

assumption is an even number. Adding the even number $2x_p$ to d yields the even number $x_q + x_p + y_q - y_p$, half of which is the x-coordinate of $\mu((x_p, y_p), (x_q, y_q))$. Adding the even number $2y_p$ to d yields the even number $x_q - x_p + y_q + y_p$, half of which is the y-coordinate of $\mu((x_p, y_p), (x_q, y_q))$. □

After Step n we have $P(v_n) = (n-2, n-2)$ because v_n is a neighbor of both v_1 and v_2. However, $P(v_{k+1})$ may not "see" all of $w_p, \ldots, w_q$, in case that the slope of $w_p w_{p+1}$ is 1 and/or the slope of $w_{q-1} w_q$ is -1 (Figure 2.21).

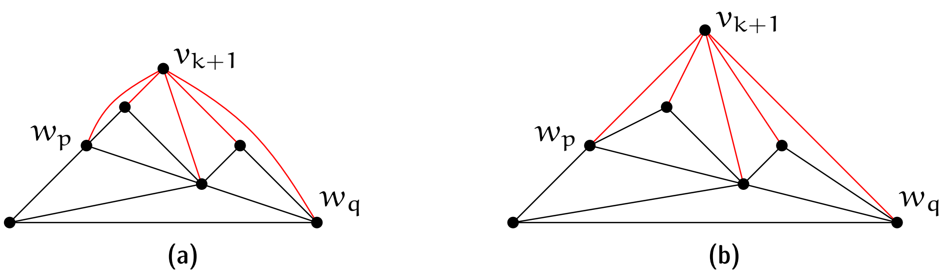

Figure 2.21: *(a) The new vertex v_{k+1} is adjacent to all of $w_p, \ldots, w_q$. If we place v_{k+1} at $\mu(w_p, w_q)$, then some edges may overlap, in case that w_{p+1} lies on the line of slope 1 through w_p or w_{q-1} lies on the line of slope -1 through w_q; (b) shifting $w_{p+1}, \ldots, w_{q-1}$ by one and $w_q, \ldots, w_t$ by two units to the right solves the problem.*

In order to resolve these problems we shift some points around so that after the shift w_{p+1} does not lie on the line of slope 1 through w_p and w_{q-1} does not lie on the line of slope -1 through w_q. The process of inserting v_{k+1} then looks as follows.

1. Shift $\bigcup_{i=p+1}^{q-1} L(w_i)$ to the right by one unit.
2. Shift $\bigcup_{i=q}^{t} L(w_i)$ to the right by two units.
3. $P(v_{k+1}) := \mu(w_p, w_q)$.
4. $L(v_{k+1}) := \{v_{k+1}\} \cup \bigcup_{i=p+1}^{q-1} L(w_i)$.

Observe that the Manhattan distance between w_p and w_q remains even because the shift increases their x-difference by two and leaves the y-coordinates unchanged. Therefore by Proposition 2.51 the vertex v_{k+1} is embedded on the integer grid.

The slopes of the edges $w_p w_{p+1}$ and $w_{q-1} w_q$ (might be just a single edge, in case that $p+1 = q$) become < 1 in absolute value, whereas the slopes of all other edges along the outer cycle remain ± 1. As all edges from v_{k+1} to $w_{p+1}, \ldots, w_{q-1}$ have slope > 1 in absolute value, and the edges $v_{k+1} w_p$ and $v_{k+1} w_q$ have slope ± 1, each edge $v_{k+1} w_i$, for $i \in \{p, \ldots, q\}$ intersects the outer cycle in exactly one point, which is w_i. In other words, adding all edges from v_{k+1} to its neighbors in G_k as straight-line segments results in a plane drawing.

Next we argue that the invariants (i)–(iii) are maintained. For (i) note that we start shifting with w_{p+1} only so that even in case that v_1 is a neighbor of v_{k+1}, it is never shifted. On the other hand, v_2 is always shifted by two because we shift every vertex starting from (and including) w_q. Clearly both the shifts and the insertion of v_{k+1} maintain the strict order along the outer cycle, and so (ii) continues to hold. Finally, regarding (iii) note that the edges $w_p w_{p+1}$ and $w_{q-1} w_q$ (possibly this is just a single edge) are the only edges on the outer cycle whose slope is changed by the shift. But these edges do not appear on $C_\circ(G_{k+1})$ anymore. The two edges $v_{k+1}w_p$ and $v_{k+1}w_q$ incident to the new vertex v_{k+1} that appear on $C_\circ(G_{k+1})$ have slope 1 and -1, respectively. So all of (i)–(iii) are invariants of the algorithm, indeed.

So far we have argued about the shift with respect to vertices on the outer cycle of G_k only. To complete the proof of Theorem 2.42 it remains to show that the drawing remains plane under shifts also in its interior part.

Lemma 2.52. *Let G_k, $k \geqslant 3$, be straight-line grid embedded as described, $C_\circ(G_k) = (w_1, \dots, w_t)$, and let $\delta_1 \leqslant \dots \leqslant \delta_t$ be nonnegative integers. If for each i, we shift $L(w_i)$ by δ_i to the right, then the resulting straight-line drawing is plane.*

Proof. Induction on k. For G_3 this is obvious. Let $v_k = w_\ell$, for some $1 < \ell < t$. Construct a delta sequence Δ for G_{k-1} as follows. If v_k has only two neighbors in G_k, then $C_\circ(G_{k-1}) = (w_1, \dots, w_{\ell-1}, w_{\ell+1}, \dots, w_t)$ and we set $\Delta = \delta_1, \dots, \delta_{\ell-1}, \delta_{\ell+1}, \dots, \delta_t$. Otherwise, $C_\circ(G_{k-1}) = (w_1, \dots, w_{\ell-1} = u_1, \dots, u_m = w_{\ell+1}, \dots, w_t)$, where $u_1, \dots, u_m$ are the $m \geqslant 3$ neighbors of v_k in G_k. In this case we set

$$\Delta = \delta_1, \dots, \delta_{\ell-1}, \underbrace{\delta_\ell, \dots, \delta_\ell}_{m-2 \text{ times}}, \delta_{\ell+1}, \dots, \delta_t \,.$$

Clearly, Δ is monotonely increasing and by the inductive assumption a correspondingly shifted drawing of G_{k-1} is plane. When adding v_k and its incident edges back, the drawing remains plane: All vertices $u_2, \dots, u_{m-1}$ (possibly none) move rigidly with (by exactly the same amount as) v_k by construction. Stretching the edges of the chain $w_{\ell-1}, w_\ell, w_{\ell+1}$ by moving $w_{\ell-1}$ to the left and/or $w_{\ell+1}$ to the right cannot create any crossings. □

Linear time. The challenge in implementing the shift algorithm efficiently lies in the eponymous shift operations, which modify the x-coordinates of potentially many vertices. In fact, it is not hard to see that a naive implementation—which keeps track of all coordinates explicitly—may use quadratic time. De Fraysseix et al. described an implementation of the shift algorithm that uses $O(n \log n)$ time. Then Chrobak and Payne [11] observed how to improve the runtime to linear, using the following ideas.

Recall that $P(v_{k+1}) = (x_{k+1}, y_{k+1})$, where

$$\begin{aligned} x_{k+1} &= \frac{1}{2}(x_p - y_p + x_q + y_q) \text{ and} \\ y_{k+1} &= \frac{1}{2}(-x_p + y_p + x_q + y_q) = \frac{1}{2}((x_q - x_p) + y_p + y_q)\,. \end{aligned} \tag{2.53}$$

Thus,

$$x_{k+1} - x_p = \frac{1}{2}((x_q - x_p) + y_q - y_p). \qquad (2.54)$$

In other words, we need the y-coordinates of w_p and w_q together with the relative x-position (offset) of w_p and w_q only to determine the y-coordinate of v_{k+1} and its offset to w_p.

Maintain the outer cycle as a rooted binary tree T, with root v_1. For each node v of T, the *left child* is the first vertex covered by insertion of v (if any), that is, w_{p+1} in the terminology from above (if $p+1 \neq q$), whereas the *right child* of v is the next node along the outer cycle (if any; either along the current outer cycle or along the one at the point where both points were covered together). See Figure 2.22 for an example.

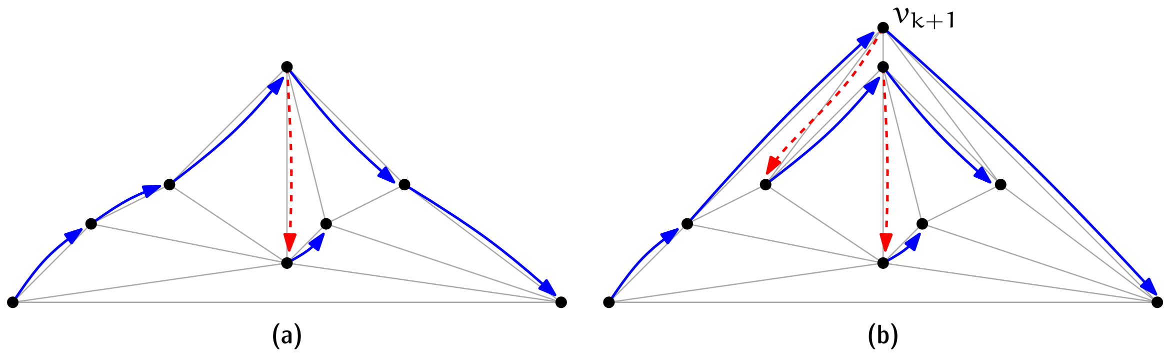

Figure 2.22: *Maintaining a binary tree representation to keep track of the x-coordinates when inserting a new vertex v_{k+1}. Red (dashed) arrows point to left children, blue (solid) arrows point to right children.*

At each node v of T we also store its x-offset $\mathrm{dx}(v)$ with respect to the parent node. For the root v_1 of the tree set $\mathrm{dx}(v_1) = 0$. In this way, a whole subtree (and, thus, a whole set $L(\cdot)$) can be shifted by changing a single offset entry at its root.

Initially, $\mathrm{dx}(v_1) = 0$, $\mathrm{dx}(v_2) = \mathrm{dx}(v_3) = 1$, $y(v_1) = y(v_2) = 0$, $y(v_3) = 1$, $\mathrm{left}(v_1) = \mathrm{left}(v_2) = \mathrm{left}(v_3) = 0$, $\mathrm{right}(v_1) = v_3$, $\mathrm{right}(v_2) = 0$, and $\mathrm{right}(v_3) = v_2$.

Inserting a vertex v_{k+1} works as follows. As before, let $w_1, \dots, w_t$ denote the vertices on the outer cycle $C_\circ(G_k)$ and $w_p, \dots, w_q$ be the neighbors of v_{k+1}.

1. Increment $\mathrm{dx}(w_{p+1})$ and $\mathrm{dx}(w_q)$ by one. *(This implements the shift.)*

2. Compute $\Delta_{pq} = \sum_{i=p+1}^{q} \mathrm{dx}(w_i)$. *(This is the total offset between w_p and w_q.)*

3. Set $\mathrm{dx}(v_{k+1}) \leftarrow \frac{1}{2}(\Delta_{pq} + y(w_q) - y(w_p))$ and $y(v_{k+1}) \leftarrow \frac{1}{2}(\Delta_{pq} + y(w_q) + y(w_p))$. *(This is exactly what we derived in (2.53) and (2.54).)*

4. Set $\mathrm{right}(w_p) \leftarrow v_{k+1}$ and $\mathrm{right}(v_{k+1}) \leftarrow w_q$. *(Update the current outer cycle.)*

5. If $p + 1 = q$, then set left$(v_{k+1}) \leftarrow 0$;
   else set left$(v_{k+1}) \leftarrow w_{p+1}$ and right$(w_{q-1}) \leftarrow 0$.
   (Update $L(v_{k+1})$, the part that is covered by insertion of v_{k+1}.)

6. Set $dx(w_q) \leftarrow \Delta_{pq} - dx(v_{k+1})$;
   if $p + 1 \neq q$, then set $dx(w_{p+1}) \leftarrow dx(w_{p+1}) - dx(v_{k+1})$.
   (Update the offsets according to the changes in the previous two steps.)

Observe that the only step that possibly cannot be executed in constant time is Step 2. To analyze Step 2, note that all vertices but the last vertex w_q for which we sum the offsets are covered by the insertion of v_{k+1}. As every vertex can be covered at most once, the overall complexity of this step during the algorithm is linear. Therefore, this first phase of the algorithm can be completed in linear time.

In a second phase, the final x-coordinates can be computed from the offsets by a single recursive pre-order traversal of the tree. The (pseudo–)code given below is to be called with the root vertex v_1 and an offset of zero. Clearly this yields a linear time algorithm overall.

```
compute_coordinate(Vertex v, Offset d) {
  if (v == 0) return;
  x(v) = dx(v) + d;
  compute_coordinate(left(v), x(v));
  compute_coordinate(right(v), x(v));
}
```

2.5.3 Remarks and Open Problems

From a geometric complexity point of view, Theorem 2.42 provides very good news for planar graphs in a similar way that the Euler Formula does from a combinatorial complexity point of view. Euler's Formula tells us that we can obtain a combinatorial representation (for instance, as a DCEL) of any plane graph using $O(n)$ space, where n is the number of vertices.

Now the shift algorithm tells us that for any planar graph we can even find a geometric plane (straight-line) representation using $O(n)$ space. In addition to the combinatorial information, we only have to store $2n$ numbers from the range $\{0, 1, \ldots, 2n - 4\}$.

When we make such claims regarding space complexity we implicitly assume the so-called *word RAM model*. In this model each address in memory contains a *word* of b bits, which means that it can be used to represent any integer from $\{0, \ldots, 2^b - 1\}$. One also assumes that b is sufficiently large, for instance, in our case $b \geqslant \log n$.

There are also different models such as the *bit complexity model*, where one is charged for every bit used to store information. In our case that would already incur an additional factor of $\log n$ for the combinatorial representation: for instance, for each halfedge we store its endpoint, which is an index from $\{1, \ldots, n\}$.

Edge lengths. Theorem 2.42 shows that planar graphs admit a plane straight-line drawing where all vertices have integer coordinates. It is an open problem whether a similar statement can be made for edge lengths.

Problem 2.55 (Harborth's Conjecture [17]). Every planar graph admits a plane straight-line drawing where all Euclidean edge lengths are integral.

Without the planarity restriction such a drawing is possible because for every $n \in \mathbb{N}$ one can find a set of n points in the plane, not all collinear, such that their distances are all integral. In fact, such a set of points can be constructed to lie on a circle of integral radius [2]. When mapping the vertices of K_n onto such a point set, all edge lengths are integral. In the same paper it is also shown that there exists no infinite set of points in the plane so that all distances are integral, unless all of these points are collinear. Unfortunately, collinear point sets are not very useful for drawing graphs. The existence of a dense subset of the plane where all distances are rational would resolve Harborth's Conjecture. However, it is not known whether such a set exists, and in fact the suspected answer is "no".

Problem 2.56 (Erdős–Ulam Conjecture [12]). There is no dense set of points in the plane whose Euclidean distances are all rational.

Generalizing the Fáry-Wagner Theorem. As discussed above, not every planar graph on n vertices admits a plane straight-line embedding on every set of n points. But Theorem 2.39 states that for every planar graph G on n vertices there *exists* a set P of n points in the plane so that G admits a plane straight-line embedding on P (that is, so that the vertices of G are mapped bijectively to the points in P). It is an open problem whether this statement can be generalized to hold for several graphs, in the following sense.

Problem 2.57. What is the largest number $k \in \mathbb{N}$ for which the following statement holds? For every collection of k planar graphs $G_1, \ldots, G_k$ on n vertices each, there exists a set P of n points so that G_i admits a plane straight-line embedding on P, for every $i \in \{1, \ldots, k\}$.

By Theorem 2.39 we know that the statement holds for $k = 1$. Already for $k = 2$ it is not known whether the statement holds. However, it is known that k is finite [8]. Specifically, there exists a collection of 49 planar graphs on 11 vertices each so that for every set P of 11 points in the plane at least one of these graphs does not admit a plane straight-line embedding on P [29]. Therefore we have $k \leqslant 49$.

Questions

1. *What is an embedding? What is a planar/plane graph?* Give the definitions and explain the difference between planar and plane.

2. *How many edges can a planar graph have? What is the average vertex degree in a planar graph?* Explain Euler's formula and derive your answers from it.
3. *How can plane graphs be represented on a computer?* Explain the DCEL data structure and how to work with it.
4. *How can a given plane graph be (topologically) triangulated efficiently?* Explain what it is, including the difference between topological and geometric triangulation. Give a linear time algorithm, for instance, as in Theorem 2.31.
5. *What is a combinatorial embedding? When are two combinatorial embeddings equivalent? Which graphs have a unique combinatorial plane embedding?* Give the definitions, explain and prove Whitney's Theorem.
6. *What is a canonical ordering and which graphs admit such an ordering? For a given graph, how can one find a canonical ordering efficiently?* Give the definition. State and prove Theorem 2.44.
7. *Which graphs admit a plane embedding using straight line edges? Can one bound the size of the coordinates in such a representation?* State and prove Theorem 2.42.

References

[1] Oswin Aichholzer, Thomas Hackl, Matias Korman, Marc van Kreveld, Maarten Löffler, Alexander Pilz, Bettina Speckmann, and Emo Welzl, Packing plane spanning trees and paths in complete geometric graphs. *Inform. Process. Lett.*, 124, (2017), 35–41.

[2] Norman H. Anning and Paul Erdős, Integral distances. *Bull. Amer. Math. Soc.*, 51/8, (1945), 598–600.

[3] Bruce G. Baumgart, A polyhedron representation for computer vision. In *Proc. AFIPS Natl. Comput. Conf.*, vol. 44, pp. 589–596, AFIPS Press, Alrington, Va., 1975.

[4] Ahmad Biniaz and Alfredo García, Packing plane spanning trees into a point set. *Comput. Geom. Theory Appl.*, 90, (2020), 101653.

[5] Prosenjit Bose, Ferran Hurtado, Eduardo Rivera-Campo, and David R. Wood, Partitions of complete geometric graphs into plane trees. *Comput. Geom. Theory Appl.*, 34/2, (2006), 116–125.

[6] John M. Boyer and Wendy J. Myrvold, On the cutting edge: simplified O(n) planarity by edge addition. *J. Graph Algorithms Appl.*, 8/3, (2004), 241–273.

[7] Sergio Cabello, Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. *J. Graph Algorithms Appl.*, 10/2, (2006), 353–363.

[8] Jean Cardinal, Michael Hoffmann, and Vincent Kusters, On universal point sets for planar graphs. *J. Graph Algorithms Appl.*, 19/1, (2015), 529–547.

[9] Bernard Chazelle, Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6/5, (1991), 485–524.

[10] Norishige Chiba and Takao Nishizeki, The Hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *J. Algorithms*, 10/2, (1989), 187–211.

[11] Marek Chrobak and Thomas H. Payne, A linear-time algorithm for drawing a planar graph on a grid. *Inform. Process. Lett.*, 54, (1995), 241–246.

[12] Paul Erdős, Ulam, the man and the mathematician. *J. Graph Theory*, 9/4, (1985), 445–449.

[13] István Fáry, On straight lines representation of planar graphs. *Acta Sci. Math. Szeged*, 11/4, (1948), 229–233.

[14] Hubert de Fraysseix, Patrice Ossona de Mendez, and Pierre Rosenstiehl, Trémaux trees and planarity. *Internat. J. Found. Comput. Sci.*, 17/5, (2006), 1017—-1030.

[15] Hubert de Fraysseix, János Pach, and Richard Pollack, How to draw a planar graph on a grid. *Combinatorica*, 10/1, (1990), 41–51.

[16] Leonidas J. Guibas and Jorge Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4/2, (1985), 74–123.

[17] Heiko Harborth and Arnfried Kemnitz, Plane integral drawings of planar graphs. *Discrete Math.*, 236/1–3, (2001), 191–195.

[18] Dawei He, Yan Wang, and Xingxing Yu, The Kelmans-Seymour conjecture IV: A proof. *J. Combin. Theory Ser. B*, 144, (2020), 309–358.

[19] John Hopcroft and Robert E. Tarjan, Efficient planarity testing. *J. ACM*, 21/4, (1974), 549–568.

[20] Ken ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed, The disjoint paths problem in quadratic time. *J. Combin. Theory Ser. B*, 102/2, (2012), 424–435.

[21] Lutz Kettner, *Software design in computational geometry and contour-edge based polyhedron visualization*. Ph.D. thesis, ETH Zürich, Zürich, Switzerland, 1999.

[22] Kazimierz Kuratowski, Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15/1, (1930), 271–283.

[23] László Lovász, Graph minor theory. *Bull. Amer. Math. Soc.*, 43/1, (2006), 75–86.

[24] Bojan Mohar and Carsten Thomassen, *Graphs on surfaces*, Johns Hopkins University Press, Baltimore, 2001.

[25] David E. Muller and Franco P. Preparata, Finding the intersection of two convex polyhedra. *Theoret. Comput. Sci.*, 7, (1978), 217–236.

[26] Johannes Obenaus and Joachim Orthaber, Edge Partitions of Complete Geometric Graphs (Part 1). *CoRR*, abs/2108.05159.

[27] János Pach and Rephael Wenger, Embedding planar graphs at fixed vertex locations. *Graphs Combin.*, 17, (2001), 717–728.

[28] Neil Robertson and Paul Seymour, Graph Minors. XX. Wagner's Conjecture. *J. Combin. Theory Ser. B*, 92/2, (2004), 325–357.

[29] Manfred Scheucher, Hendrik Schrezenmaier, and Raphael Steiner, A Note on Universal Point Sets for Planar Graphs. *J. Graph Algorithms Appl.*, 24/3, (2020), 247–267.

[30] Walter Schnyder, Planar graphs and poset dimension. *Order*, 5, (1989), 323–343.

[31] Carsten Thomassen, Kuratowski's Theorem. *J. Graph Theory*, 5/3, (1981), 225–241.

[32] William T. Tutte, A theorem on planar graphs. *Trans. Amer. Math. Soc.*, 82/1, (1956), 99–116.

[33] Klaus Wagner, Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46, (1936), 26–32.

[34] Klaus Wagner, Über eine Eigenschaft der ebenen Komplexe. *Math. Ann.*, 114/1, (1937), 570–590.

[35] Kevin Weiler, Edge-based data structures for solid modeling in a curved surface environment. *IEEE Comput. Graph. Appl.*, 5/1, (1985), 21–40.

[36] Hassler Whitney, Congruent graphs and the connectivity of graphs. *Amer. J. Math.*, 54/1, (1932), 150–168.