

Embedding Stacked Polytopes on a Polynomial-Size Grid

Erik D. Demaine*

André Schulz†

Abstract

We show how to realize a stacked 3D polytope (formed by repeatedly stacking a tetrahedron onto a triangular face) by a strictly convex embedding with its n vertices on an integer grid of size $O(n^4) \times O(n^4) \times O(n^{18})$. We use a perturbation technique to construct an integral 2D embedding that lifts to a small 3D polytope, all in linear time. This result solves a question posed by Günter M. Ziegler, and is the first nontrivial subexponential upper bound on the long-standing open question of the grid size necessary to embed arbitrary convex polyhedra, that is, about efficient versions of Steinitz’s 1916 theorem. An immediate consequence of our result is that $O(\log n)$ -bit coordinates suffice for a greedy routing strategy in planar 3-trees.

1 Introduction

Steinitz’s Theorem [25, 33] states that the graphs of 3D polytopes¹ are exactly the planar 3-connected graphs. In particular, every planar 3-connected graph can be realized by a 3D polytope. The original proof is constructive, transforming the graph by a sequence of local operations down to a tetrahedron. Unfortunately, the resulting polytope construction requires exponentially many bits of accuracy for each vertex coordinate. Stated another way, this construction can place the n vertices on an integer grid, but that grid may have dimensions doubly exponential in n [15].

How large an integer grid do we need to embed a given planar 3-connected graph as a polytope? This question goes back at least fifteen years as Problem 4.16 in Günter M. Ziegler’s book [33]; he wrote that “it is quite possible that there is a quadratic upper bound” on the length of the maximum dimension. The best bound so far is exponential in n , namely $O(2^{7.21n})$ [5, 18]; see below for the long history. The central question is whether a polynomial grid suffices, that is, whether

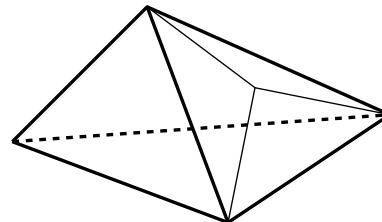


Figure 1: A stacked polytope obtained by two stacking operations.

Steinitz’s Theorem can be made efficient. For comparison, a planar graph can be embedded in the plane with strictly convex faces using a polynomial-size grid [3]. In this paper, we give the first nontrivial subexponential upper bound for a large class of polytopes.

Stacked polytopes. A *stacked polytope* is a 3D polytope that is constructed by a sequence of “stacking operations” applied to a triangle. A *stacking operation* glues a tetrahedron atop a triangular face f of the polytope, by identifying f with a face of the tetrahedron, while maintaining the convexity of the polytope. Thus a stacking operation removes one face f and adds three new faces having a new common vertex. We call this new vertex *stacked on f* . Stacked polytopes seem a natural class to study, because the stacking operation is a special case (perhaps the simplest) of the operations in Steinitz’s proof. Thus our solution for stacked polytopes has potential to be generalized to general 3D polytopes. The graphs of stacked polytopes are the planar 3-trees, that is, maximal planar graphs of treewidth 3.

Our results. We present an algorithm that realizes a stacked polytope on a grid whose dimensions are polynomial in n . Our main result is the following:

THEOREM 1. *Every planar 3-tree G can be realized as a stacked polytope whose coordinates are nonnegative integers. The largest x and y coordinates are $10n^4$ and all z coordinates are smaller than $224,000 n^{18}$. The grid embedding can be computed in linear time.*

Geometric routing. One application of this embedding is a geometric routing scheme that uses small coordinates: $O(\log n)$ bits per vertex.

To send a message over a network, an easy strategy is to send the message at each step to a node that is

*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, edemaine@mit.edu.

†Institut für Mathematische Logik und Grundlagenforschung, Universität Münster, andre.schulz@uni-muenster.de. Supported by the German Research Foundation (DFG) under grant SCHU 2458/1-1.

¹A *polytope* is a convex polyhedron, and its *graph* (also known as 1-skeleton) is formed by its vertices and edges.

closer to the target than the current node. This strategy is known as *greedy* or *geometric routing*. It suffices for every node to know the coordinates of its neighbors and the target. A *greedy embedding* of a graph guarantees that geometric routing works, that is, messages always eventually reach their destination.

In their seminal paper introducing geometric routing, Papadimitriou and Ratajczak [16] noticed that convex 3D polytopes are greedy embeddings of planar 3-connected graphs. Later Moitra and Leighton [14] found a way to compute a 2D greedy embedding for such graphs. Their embedding uses integer coordinates that are exponential in the number n of nodes, and hence the space to store the $\Theta(n)$ -bit coordinates for the neighbors and the target node is likely too big to justify geometric routing via these embeddings. Goodrich and Strash [10] extended the results of [14] and found a 2D greedy embedding with succinct representation of its coordinates. The 2D embeddings, however, have undesirable features; for example, they may have crossings and they are only determined by a spanning tree of the graph. By using an embedding of the graph as a 3D polytope, we obtain a more natural greedy embedding, and using our results, the coordinates of each vertex can be represented using $O(\log n)$ bits.

Related work. Several algorithms have been developed to realize a given graph as a 3D polytope. Most of these algorithms are based on the following two-stage approach. The first stage computes a plane 2D embedding. To extend the 2D drawing to a 3D polytope, the 2D drawing must fulfill a criterion which can be phrased as an “equilibrium stress condition”. Roughly speaking, replacing every edge of the graph with a spring, the resulting system of springs must be in a stable state for the 2D embedding. Plane drawings that fulfill this criterion for the interior vertices can be computed as barycentric embeddings, i.e., by Tutte’s method [28, 29]. The main difficulty is to guarantee the equilibrium condition for the boundary vertices as well, because in general this goal is achievable only for certain locations of the outer face. The second stage computes a 3D polytope by assigning every vertex a height expressed in terms of the spring constants of the system of springs.

The two-stage approach finds application in a series of algorithms [6, 8, 11, 15, 18, 19, 24]. The first result that improves the induced grid embedding of Steinitz’s construction is due to Onn and Sturmfels [15]; they achieved a grid size of $O(n^{160n^3})$. Richter-Gebert’s algorithm [19] uses a grid of size $O(2^{18n^2})$ for general polytopes, and a grid of size $O(2^{5.43n})$ if the graph of the polytope contains at least one triangle. These bounds were improved by Ribó [17] and later by Ribó, Rote, and Schulz [18]. The last paper expresses an

upper bound for the grid size in terms of the number of spanning trees of the graph. Using the recent bounds of Buchin and Schulz [5] on the number of spanning trees, this approach gives an upper bound on the grid size of $O(2^{7.21n})$ for general polytopes and $O(2^{4.83n})$ for polytopes with at least one triangular face. These bounds are the best known to date.

Zickfeld showed in his PhD thesis [32] that it is possible to embed very special cases of stacked polytopes on a grid polynomial in n . First, if each stacking operation takes place on one of the three faces that were just created by the previous stacking operation (what might be called *serpentine*), then there is an embedding on the $n \times n \times 3n^4$ grid. Second, if we perform the stacking in rounds, and in every round we stack on every face simultaneously (what we call the *balanced stacked polytope*), then there is an embedding on a $\frac{4}{3}n \times \frac{4}{3}n \times O(n^{2.47})$ grid. Zickfeld’s embedding algorithm for balanced stacked polytopes constructs a barycentric embedding. Because of the special structure of the underlying graph the 2D embedding remarkably fits on a small grid.

Every stacked polytope can be extended to a balanced stacked polytope at the expense of adding an exponential number of vertices. By doing so, Zickfeld’s grid embedding for the balanced case induces a $O(2^{3.91n})$ grid embedding for general stacked polytopes. From our experience, the most difficult-to-embed stacked polytopes are “almost balanced”. To construct such a polytope, we perform the stacking operations in rounds. In each round, we stack a vertex on two of the three faces that share a vertex that was introduced in the previous round.

Little is known about the lower bound of the grid size. An integral convex embedding of an n -gon in the plane needs an area of $\Omega(n^3)$ [1, 2, 27]. Therefore, realizing a 3D polytope with an $(n-1)$ -gonal face requires at least one dimension of size $\Omega(n^{3/2})$. For simplicial polytopes² (and hence stacked polytopes), this lower-bound argument does not apply. Because every orthogonal projection of a 3D polytope decomposes into two noncrossing drawings, the projections into the xy and yz plane have to contain a noncrossing drawing of at least $n/2$ vertices. Thus we know that the coordinates must be at least $\Omega(n)$.

The situation in higher dimensions is more complicated. Already in dimension 4, there are polytopes that cannot be realized with rational coordinates, and a 4-polytope that can be realized on the grid might require coordinates that are doubly exponential in the number of its vertices [33]. Moreover, it is NP-hard

²A 3D polytope is *simplicial* if its faces are all triangles.

even to decide whether a lattice is a face lattice of a 4-polytope [19, 20].

Alternative approaches for realizing general polytopes come from the original proof of Steinitz’s theorem, as well as the Koebe–Andreev–Thurston circle-packing theorem, which induces a particular polytope realization called the *canonical polytope* [33]. Das and Goodrich [7] essentially perform many inverse Steinitz operations on many independent vertices in one step, resulting in a singly exponential bound on the grid size. The proof of the Koebe–Andreev–Thurston circle-packing theorem relies on nonlinear methods and makes the features of the 3D embedding obtained from a circle packing intractable; see [23] for an overview. Lovasz [12] studied a method for realizing polytopes using a vector of the nullspace of a “Colin de Verdière matrix” of rank 3. It is easy to construct these matrices for stacked polytopes; however, without additional requirements, the computed grid embedding might again need an exponential-size grid.

Our approach. At a high level, we follow the popular two-stage approach: we compute a 2D embedding and then lift it to a 3D polytope. Usually in past work, the 2D embedding comes from as a barycentric embedding using Tutte’s method. This construction gives the equilibrium condition for the interior vertices for free. However, a barycentric embedding might require an exponential-size grid. To overcome this difficulty, we produce a special 2D embedding by mimicking the stacking operation in the plane and maintaining an equilibrium stress throughout the construction. The resulting embedding might require a large grid as well; however, we construct the triangulation G so that a small perturbation preserves the noncrossing property of the embedding. Moreover, the small perturbation changes the equilibrium stress only by a small amount. In order to make our approach work, we have to construct the (original, not perturbed) 2D embedding so that the smallest and largest equilibrium stresses are polynomially related.

To control the 2D embedding, we prescribe the areas of the faces. For a fixed outer face, these constraints determine a drawing of a planar 3-tree in the plane uniquely. Also the equilibrium stress can be expressed in terms of the face areas. Initially, all face areas are set to 1, but to prevent large stresses, we increase the areas in some faces to get a more “balanced” 2D embedding. To see which faces must be blown up, we make use of a decomposition technique from data structural analysis called *heavy-light edge decomposition* [26]. Based on this decomposition, we subdivide G into a hierarchy of simple stacked polytopes, which we use to define the face areas.

2 Stacked Polytopes

Let G be the graph of a stacked polytope P . We denote the vertex set of G by $V = \{v_1, v_2, \dots, v_n\}$ and its edge set by E . Because G is planar and 3-connected, its faces are uniquely determined [31]. We denote the triangle with which we started the stacking as the *outer face* f_0 . By convention, we label the vertices of f_0 as v_1, v_2, v_3 . During the construction of P , we may introduce a face that will later vanish after some stacking operation; we call such face a *subface*.

2.1 Tree representation. We associate every planar 3-tree G with a rooted ternary tree $\mathcal{T}(G)$ that encodes the sequence of stacking operations that led to G . The internal nodes of $\mathcal{T}(G)$ represent the subfaces of G , and the leaves of $\mathcal{T}(G)$ the faces of G . See Figure 2 for an example. The three children of an internal node represent the three (sub)faces obtained by stacking a vertex on that subface. Every subtree of $\mathcal{T}(G)$ corresponds to a subgraph of G that is also a planar 3-tree. Notice that a tree $\mathcal{T}(G)$ does not specify G uniquely. A unique specification can be achieved by ordering the tree edges such that the left/middle/right child is associated with a unique (sub)face. However, for the scope of this paper, a bijection between G and $\mathcal{T}(G)$ is not necessary, because we use $\mathcal{T}(G)$ as a simplified description of the structure of G .

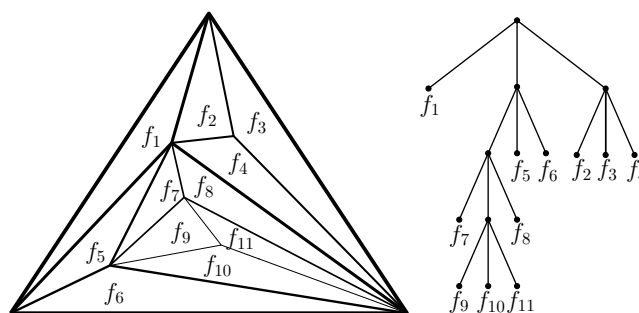


Figure 2: The graph G of a stacked polytope (on the left) and its associated ternary tree $\mathcal{T}(G)$ (on the right).

2.2 2D drawings of planar 3-trees. Let $w: F \rightarrow \mathbb{R}_+$ be a function that assigns a positive integer to every face of G . We aim at constructing a 2D drawing such that every face f of G is realized with area $w(f)$. We extend the notion of w to the subfaces s of G by $w(s) := \sum \{w(f) \mid \text{leaf } f \text{ is in the subtree rooted at } s\}$. A planar straight-line drawing is described by a function \mathbf{p} that assigns 2D coordinates to every vertex. For simplicity, we write $\mathbf{p}(v_i) =: \mathbf{p}_i = (x_i, y_i)$. The following

lemma also appears in [4], including an analysis of the induced resolution.

LEMMA 1. *Let G be a planar 3-tree and w a weight function for its faces. Then G admits a 2D drawing such that every face in G has area $w(f)$. The embedding can be computed in linear time.*

Proof. We construct the 2D drawing by first drawing the outer face and then performing the stacking operations step by step. During the construction we guarantee the invariant that every subface s , realized so far, has area $w(s)$. Each stacking operation inserts a vertex v inside the subface s and creates three new (sub)faces. We can always locate v inside s such that $w(s)$ will be partitioned as described by the weights of the new (sub)faces. The feasible location can be expressed in terms of barycentric coordinates and hence be computed in $O(1)$ time per stacked vertex. \square

2.3 Lifting a stacked polytope. Once we obtain a 2D drawing \mathbf{p} of G , we can assign every vertex v_i a suitable height h_i . The function $h: V \rightarrow \mathbb{R}$ that assigns every vertex v_i the height $h_i := h(v_i)$ is called a *lifting*. By construction, the vertices of the outer face will always remain in the $z = 0$ plane.

To study the space of liftings, we introduce the concept of “equilibrium stress”:

DEFINITION 1. *An assignment $\omega: E \rightarrow \mathbb{R}$ of scalars (denoted by $\omega(i, j) = \omega_{ij} = \omega_{ji}$) to the edges of G is called a stress. The stress ω is an equilibrium stress for an embedding \mathbf{p} if, for every vertex v_i , we have*

$$(2.1) \quad \sum_{j:(i,j) \in E} \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{0}.$$

In the 19th century, James Clerk Maxwell observed that there is a bijection between 2D drawings with equilibrium stress and projections of 3D polytopes [13], also known as the Maxwell–Cremona correspondence.

THEOREM 2. (MAXWELL, WHITELEY) *Let G be a planar 3-connected graph with 2D drawing \mathbf{p} and designated outer face f_0 . There exists a one-to-one correspondence between*

- A) equilibrium stresses ω for G at \mathbf{p} ; and
- B) liftings in \mathbb{R}^3 , where face f_0 remains in the $z = 0$ plane.

The proof that A induces B (which is the important direction for our purpose) is due to Walter Whiteley [30]. An easy way to compute the lifting from the stressed graph can be found in Richter-Gebert’s book [19, Chapter 13]. We review this method in Appendix A.

Every height assignment results in some spatial polyhedral surface on the lifted points, but only special heights lift \mathbf{p} to a convex surface. However, we can characterize all stresses that lift to a convex surface by their sign patterns.

LEMMA 2. *Let G be a planar 3-connected graph with noncrossing straight-line embedding \mathbf{p} that has an equilibrium stress ω . If $\omega_{ij} > 0$ for every interior edge (i, j) , then the lifting from the Maxwell–Cremona correspondence yields a convex 3D polytope.*

A proof can be found in [19].

Now let us study the space of equilibrium stresses for stacked polytopes with embedding \mathbf{p} . The smallest (full-dimensional) 3D stacked polytope is the tetrahedron. Its graph is K_4 , the complete graph on four vertices. For a fixed 2D drawing of K_4 (with boundary face v_1, v_2, v_3), we can assign the vertex v_4 any positive height to describe a convex lifting. Thus the space of liftings is 1-dimensional. As a consequence the equilibrium stresses for an embedding of K_4 are unique up to a multiplication with a positive scalar. The equilibrium stresses can be described in terms of the areas of the faces in the 2D drawing of K_4 . Let $[i, j, k]$ denote twice the (signed) area of the triangle $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k$:

$$[i, j, k] := \det \begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{pmatrix}.$$

The following definition follows the presentation of [21] with slight modifications:

DEFINITION 2. *Let $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l$ be four points in the plane in general position. Let G_4 be the complete graph realized on these points. For an edge (i, j) , we define the atomic equilibrium stress for the drawing of G_4 by*

$$\omega_{ij}^{kl} := [i, k, l] [j, k, l],$$

for $\{k, l\} = \{1, 2, 3, 4\} \setminus \{i, j\}$.

The atomic equilibrium stress is positive on the interior edges and negative on the boundary edges.

Every equilibrium stress on a 2D drawing of G can be expressed as a linear combination of the atomic stresses of several K_4 ’s. In fact, a stacking operation can be viewed as adding a K_4 with a scaled atomic stress to an already existing graph with equilibrium stress. More formally, we consider

$$(2.2) \quad \omega_{ij} = \sum_{\{i,j,k,l\} \in S} \alpha_{ijkl} \omega_{ij}^{kl},$$

where S lists the indices of the 4-tuples that occurred in the stacking operations during the construction of G .

The parameters α_{ijkl} are the coefficients of the linear combination and define the equilibrium stress uniquely. Throughout this paper, we consider only positive values for these coefficients.

For an interior edge (i, j) , ω_{ij} receives one positive charge from one atomic stress; all other atomic stresses contribute a negative charge. In order to define a stress that corresponds to a lifted convex polytope, we have to adjust the coefficients α so that the stress ω is positive on the interior edges.

3 The Embedding Algorithm

The embedding algorithm follows the following high-level approach. First we draw G with a noncrossing straight-line drawing. As a preprocessing step, we prescribe the desired face areas of the 2D drawing in order to get a “balanced” drawing and derive the coefficients α that specify the right equilibrium stress for the drawing. We then compute a second 2D embedding by carefully rounding the coordinates. The rounding will affect the atomic stresses, but not by much. Thus, after computing the induced lifting, we obtain a stacked polytope with small integer x and y coordinates. However, the heights are not yet integral. After scaling the coefficients α with a sufficiently large common factor, we round them to integers. This induces an integral equilibrium stress, and therefore integer heights in the lifting. It remains to show that the convexity along the lifted edges is preserved.

3.1 Balancing. The crucial step in the embedding algorithm is the preprocessing where we define the face areas $w(f)$ of the 2D embedding, but also obtain the coefficients α that will determine the lifting.

We use a tree decomposition of $\mathcal{T}(G)$ to define both face areas and α coefficients. Consider an interior node n_i in $\mathcal{T}(G)$. Let n_k be the child of n_i whose subtree has the largest number of leaves (compared to the subtrees of the other children), breaking ties arbitrarily. We call the edge (i, k) a *heavy edge*, and the edges to the other two children of n_i *light edges*. The heavy edges induce a decomposition of $\mathcal{T}(G)$ into paths, also called *heavy paths*; see Figure 3 on the left.

We extend each heavy path by re-attaching its child light edges, resulting in a graph known as a *caterpillar*. If a node of a caterpillar lies on a heavy path, we call it a *spine node*; otherwise, it is *tree node*. The *root* of the caterpillar is the spine node with shortest distance in $\mathcal{T}(G)$ to the root of $\mathcal{T}(G)$. We label the spine nodes s_1, s_2, \dots, s_\perp starting from the root of the caterpillar. We denote the tree nodes adjacent to s_i by t_{i+1} and t'_{i+1} . (It does not matter which of the two children gets which label.) The “last” spine node s_\perp has no adjacent tree

nodes. We store in every tree node t a pointer $\text{link}(t)$ to the caterpillar rooted at t . We call the described tree decomposition the *heavy caterpillar decomposition* of $\mathcal{T}(G)$. Figure 3 shows an example on the right.

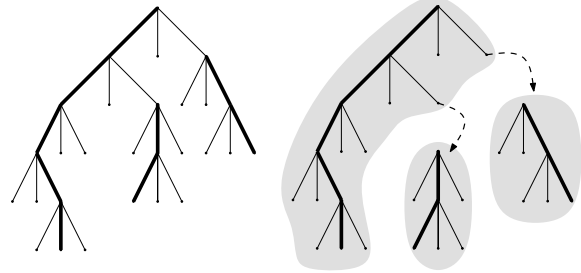


Figure 3: The tree $\mathcal{T}(G)$ and its heavy edges (drawn bold) on the left. On the right we depicted the induced heavy caterpillar decomposition

Every caterpillar C in the heavy caterpillar decomposition is a rooted ternary tree, which is a subgraph of $\mathcal{T}(G)$. A *2D drawing* of C is a drawing of a stacked polytope as a triangulation whose tree representation coincides with C , and whose combinatorial structure respects the face structure of G restricted to the subfaces of C . Furthermore, the drawing of C must respect the face areas specified by w . The *level* of a caterpillar is the number of light edges traversed by walking in $\mathcal{T}(G)$ from the caterpillar’s root to the root of $\mathcal{T}(G)$. The level of a caterpillar is at most $\lg n$.

The tree nodes of the caterpillars define rooted subtrees of $\mathcal{T}(G)$. Every node in $\mathcal{T}(G)$ corresponds to a (sub)face f . The face areas of the 2D drawing are computed with the help of the tree decomposition. With each node n_i in $\mathcal{T}(G)$, we associate a weight $w(n_i)$ that corresponds to the weight of the subface represented by n_i . By definition, $w(s_{i-1}) = w(s_i) + w(t_i) + w(t'_i)$, and $w(s_i) \geq w(t_i), w(t'_i)$. For each spine node n_s , we store the α coefficient that corresponds to the K_4 that was inserted by stacking a vertex on the subface represented by n_s .

To simplify further constructions, we show how to adjust the weights w such that, for all tree vertices, we have $w(t_i) = w(t'_i)$. We call this process *balancing*. We start by assigning all faces of G a weight of 1, and update the accumulated weights of the subfaces appropriately. We then update the weights recursively. Assume that we process caterpillar C and $w(t_i) = w(t'_i)$ holds for all caterpillars “below” C . For every pair of tree nodes t_i, t'_i in C , we increase the weight of the smaller tree, say t_i , such that both weights are equal. To maintain consistency, we have to increase the weight in one of the (sub)faces of $\text{link}(t_i)$. To avoid recursive updates,

we pick the face s_\perp in $\text{link}(t_i)$ and increase its weight. We also have to update the weight of the tree node t with $\text{link}(t) = C$. Further updating is not necessary because we process level by level.

Algorithm 1 gives pseudocode for this algorithm. To compute the weights w , we execute $\text{BALANCE}(C_0)$, where C_0 is the caterpillar at level 0.

Algorithm 1 $\text{BALANCE}(C)$.

Input: A caterpillar C from the heavy caterpillar decomposition of $\mathcal{T}(G)$.

Output: Weights for the nodes of $\mathcal{T}(G)$.

```

1: for all  $t_i, t'_i$  in  $C$  do
2:    $\text{BALANCE}(\text{link}(t_i))$ 
3:    $\text{BALANCE}(\text{link}(t'_i))$ 
4:   if  $w(t_i) > w(t'_i)$  then
5:      $\text{relab } t_i \leftrightarrow t'_i$ 
6:   end if
7:    $w(t_i) = w(t'_i)$ 
8:   add  $(w(t'_i) - w(t_i))$  to the weight of  $s_\perp$  in  $\text{link}(t_i)$ 

9:   add  $(w(t'_i) - w(t_i))$  to the weight of  $\text{link}^{-1}(C)$ 
10: end for

```

How much does calling $\text{BALANCE}(C_0)$ increase the weight of the outer face? Let C be a caterpillar of level l and assume we have balanced all caterpillars with larger level. By adjusting the weights in C , we at most double the current weight of the caterpillar. The caterpillars in each level behave independently. Hence, for every level, the sum of the weights at most doubles. Therefore the total increased weight is

$$(3.3) \quad w(f_0) < \underbrace{2 \cdot 2 \cdot \dots \cdot 2}_{\lg n \text{ times}} n = n^2.$$

For convenience, we increase the weight of s^\perp in the topmost caterpillar such that $w(f_0)$ is exactly n^2 .

The coefficients α are defined as follows. Let v_i be the vertex that is stacked on the face $v_j v_k v_l$. We have to specify the coefficient α_{ijkl} that describes the amount of the stress on the K_4 realized by these four vertices. Let s_u represent the spine node of the subface $v_j v_k v_l$ in the caterpillar C . We define

$$\alpha_{ijkl} := \frac{1}{w(t_u)} = \frac{1}{w(t'_u)}, \quad \text{for } t_u, t'_u \text{ in } C.$$

3.2 A quantitative analysis of the stress. We show in this subsection that the stress ω defined by our choice of coefficients α and 2D drawing is positive on every interior edge and hence induces a lifting to a convex polytope. More importantly, we show that the stresses are within a polynomial range.

The stress on an interior edge (i, j) consists of a positive part ω_{ij}^+ and a negative part $\omega_{ij}^- := \omega_{ij} - \omega_{ij}^+$. The positive part comes from a single (weighted) atomic stress; the negative part might be a sum of several atomic stresses. We assume by symmetry that the vertex v_i was stacked after v_j . The positive part ω_{ij}^+ comes from an atomic stress defined from by stacking v_i inside a subface that contains v_j .

LEMMA 3. *Let v_i be introduced by stacking v_i to the subface f spanned by v_j, v_x , and v_y . Then*

$$\omega_{ij}^+ \geq w(f).$$

Proof. The stacking of v_i is recorded in some caterpillar, say C_i , in the heavy caterpillar decomposition of $\mathcal{T}(G)$. Let v_i be the common vertex of subfaces represented by s_k, t_k, t'_k in C_i . We call an edge in the 2D drawing of a caterpillar a *spine edge* if its two incident faces are represented as tree nodes in the caterpillar.

Case 1: (i, j) is a spine edge in C_i . In this case

$$\begin{aligned} \omega_{ij}^+ &= \alpha_{ijxy} \omega_{ij}^{xy} = \alpha_{ijxy} [i, x, y] [j, x, y] \\ &= \frac{w(s_k) w(s_{k-1})}{w(t_k)} \geq w(s_{k-1}) = w(f). \end{aligned}$$

Case 2: (i, j) is a not spine edge in C_i . In this case

$$\begin{aligned} \omega_{ij}^+ &= \alpha_{ijxy} \omega_{ij}^{xy} = \alpha_{ijxy} [i, x, y] [j, x, y] \\ &= \frac{w(s_{k-1}) w(t_k)}{w(t_k)} = w(s_{k-1}) = w(f). \end{aligned}$$

□

The next lemma bounds the influence of all atomic stresses that are defined by stacking operations within a subface.

LEMMA 4. *Let f be a subface in G with boundary vertices v_i, v_j, v_k . Let ω_{ij}^f be the (negative) stress on the boundary edge (i, j) induced by the linear combination of the atomic equilibrium stresses within f . Further let f^- be the face contained in f with $w(f^-)$ minimal. Then*

$$-\omega_{ij}^f < w(f) - w(f^-).$$

Proof. Every atomic stress “within” f that contributes a negative charge is defined by a K_4 that contains v_i, v_j , and two other vertices. Because the 2D drawing of G is planar, the K_4 ’s defining these charges are nested; see Figure 4. We name two remaining vertices of the K_4 that appears in the l th position of the nesting v_{k_l} and $v_{k_{l+1}}$ such that $v_{k_1} = v_k$. To determine ω_{ij}^f , we have to sum up all $\alpha_{ijk_l k_{l+1}} \omega_{ij}^{k_l, k_{l+1}}$. Let $\alpha_{ijxy} \omega_{ij}^{xy}$ be one of the summands ($x = k_l, y = k_{l+1}$). By Definition 2,

$$\omega_{ij}^{xy} = [i, x, y] [j, x, y].$$

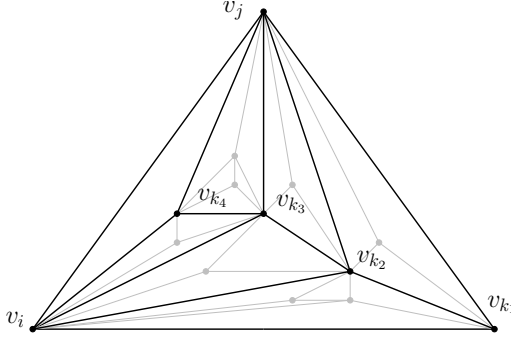


Figure 4: The part of G that contributes to ω_{ij}^f , drawn as dark edges.

The value of α_{ijxy} depends on the heavy caterpillar decomposition. We know that the stacking of v_y is recorded in some caterpillar C , which also defines the corresponding coefficient α . In particular, C has a spine node, say s_{q-1} , that represents the subface spanned by v_i, v_j, v_x . The other three faces of the K_4 are represented in C by t_q, t'_q , and s_q , but we do not know which face belongs to which node in C . We distinguish three cases; refer to Figure 5.

Case 1: s_q represents the face spanned by v_i, v_j, v_y . In this case we have that $\alpha_{ijxy} = |1/[ixy]|$, and therefore $\alpha_{ijxy}\omega_{ij}^{ky} = -|[jky]|$.

Case 2: s_q represents the face spanned by v_i, v_x, v_y . In this case we have that $\alpha_{ijxy} = |1/[jxy]|$, and therefore $\alpha_{ijxy}\omega_{ij}^{ky} = -|[ixy]|$.

Case 3: s_q represents the face spanned by v_j, v_x, v_y . In this case we have that $\alpha_{ijxy} = |1/[ixy]|$, and therefore $\alpha_{ijxy}\omega_{ij}^{ky} = -|[jxy]|$.

In all three cases, the negative charge equals the area of some subface in the 2D drawing. Because the K_4 's are nested, the interior of all subfaces defining these charges are disjoint. Hence, in total, we cannot exceed $[i, j, k]$. Moreover, at least two faces are missing in the sum and thus $-\omega_{ij}^f < w(f) - w(f^-)$. \square

We can now combine the results of Lemmas 3 and 4:

LEMMA 5. *Let ω be the equilibrium stress for G defined for our choice of the 2D drawing and coefficients α . Further let f^- be the face in G with $w(f^-)$ minimal.*

1. For every interior edge (i, j) , we have that $w(f_0) > \omega_{ij} > 3w(f^-)$.
2. For every boundary edge (i, j) , we have that $-\omega_{ij} < w(f_0)$.

Proof. To get an estimate for ω_{ij} on an interior edge (i, j) , we add $\omega_{ij}^+ + \omega_{ij}^-$. Assume by symmetry that

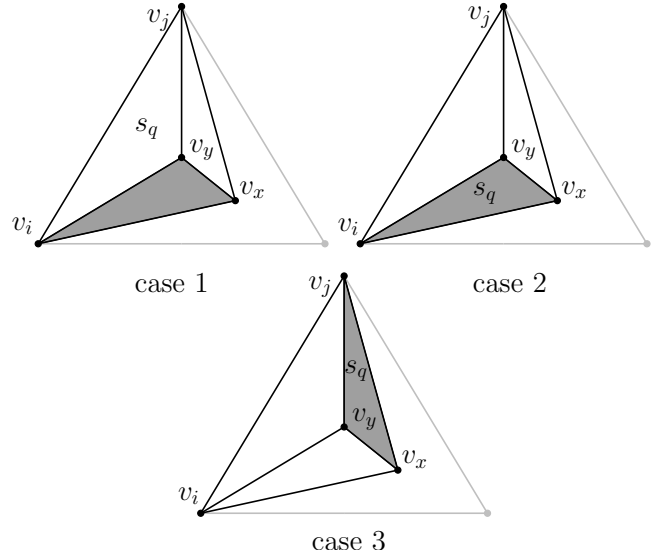


Figure 5: The three cases discussed in the proof of Lemma 4. The shaded area indicates the negative charge contributed to ω_{ij}^f .

we stacked v_i after v_j , and created the subfaces f_1, f_2 , and f_3 by stacking v_i . By Lemma 3, ω_{ij}^+ is larger than $w(f_1) + w(f_2) + w(f_3)$. Suppose that f_1 and f_2 are incident to (i, j) . All faces contained in f_1, f_2 that are incident to (i, j) contribute something to ω_{ij}^- . By Lemma 4, $-\omega_{ij}^- < w(f_1) + w(f_2) - 2w(f^-)$ and therefore $\omega_{ij} > 2w(f^-) + w(f_3) \geq 3w(f^-)$. By Lemma 3, no stress on an interior edge is larger than $w(f_0)$. Lemma 4 induces a bound on the stress on the boundary, namely $w(f_0)$. \square

3.3 Perturbation.

3.3.1 The general idea. Let us pause to recap what we have achieved so far. We defined an area assignment and an equilibrium stress. The prescribed areas uniquely define a 2D embedding once we fix the location of the boundary face. The equilibrium stress is independent of our choice of the boundary face. Lemma 5 showed two properties. (1) The stress is not too small on the interior edges. This implies that the “creasing” is not too small, so the edges in the lifting remain convex after a small perturbation. (2) The absolute stress on the boundary is not too large, so the dihedral angle between the outer face and its adjacent interior face is small. This indicates that the lifting has small heights.

For convenience, we multiply all prescribed face areas by $1/2$. Suppose we realize the boundary face

at $\mathbf{p}_1 = (0, 0)$, $\mathbf{p}_2 = (n, 0)$, and $\mathbf{p}_3 = (0, n)$. Thus the area of the boundary face has the right area, namely $n^2/2$. The smallest face has area at least $1/2$. Therefore, the triangular faces in the 2D embedding are not too skinny. In particular, the longest edge is at most $\sqrt{2}n$, and thus the smallest height in a triangular face is at most $1/(\sqrt{2}n)$. As a consequence, a small perturbation (i.e., adding a sufficiently small polynomially related vector to each point) cannot make cause faces to “flip over”, and hence such a perturbation will introduce no crossings. Moreover, the triangle areas will change only by a small polynomial, which implies that the atomic stresses change also only by a small polynomial. Instead of rounding to a fine grid, we select an appropriately large boundary face (including scaling the prescribed face areas) and then round to integer grid points.

So far we have obtained integer 2D coordinates but not necessarily integer heights in the lifting. We use the following idea to construct integer heights. The coefficients α are rational numbers between 0 and 1. All atomic stresses are a product of two integers, if the 2D coordinates are integers. If furthermore the coefficients α are integral, then the induced equilibrium stress is integral as well. An integral equilibrium stress with integer grid embedding yields integer heights, because the computation of the heights boils down to adding and multiplying integers (see Appendix A). Our plan is to round the coefficients α . In order to preserve the properties of the induced equilibrium stress, we have to scale all coefficients α_{ijkl} by a common factor. For every interior edge (i, j) , only one atomic stress is positive on (i, j) ; all other atomic stresses are negative. However, the positive part dominates the negative part. By scaling the coefficients α and rounding them down, we might decrease the positive atomic stress. By picking a sufficiently large scaling factor, the decrement will not affect the properties of the lifting, and preserve the sign pattern of the induced equilibrium stress.

3.3.2 Perturbation by rounding in detail. We locate the outer face at $\mathbf{p}_1 = (0, 0)$, $\mathbf{p}_2 = (10n^4, 0)$, and $\mathbf{p}_3 = (0, 10n^4)$. Thus the outer face has area $w(f_0) = 50n^8$. This implies that we have scaled the original area assignment by a factor of $50n^6$. As a consequence, the smallest face f^- has area at least $w(f^-) = 50n^6$. On the other hand, the longest edge in the drawing is at most $\ell_{\max} := 10\sqrt{2}n^4$. From now on we refer to our particular choice of the (scaled) drawing as \mathbf{p} .

To obtain small coordinates, we round every vertex down. Thus we obtain a new drawing \mathbf{r} , defined by $\mathbf{r}_i := (\lfloor x_i \rfloor, \lfloor y_i \rfloor)$. The new drawing changes the areas of all subfaces and hence the atomic equilibrium stresses.

As a consequence, the stress ω defined by the coefficients α will change as well. Let $\bar{\omega}$ denote the modified stress (defined by the same coefficients α).

LEMMA 6. *Let Δ be a triangle with area A and longest side length at most ℓ . Suppose we perturb each corner of Δ , by adding a vector whose Euclidean norm is at most 1. For the area A' of the perturbed triangle, we have*

$$|A - A'| \leq \frac{3}{2}(\ell + 1).$$

Proof. Let $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ be the corners of Δ , with ℓ realized between $\mathbf{p}_1, \mathbf{p}_2$. We perturb the points one after another. Let h_i be the height of \mathbf{p}_i relative to its opposing triangle edge g_i . By perturbing only one point \mathbf{p}_i , we change h_i by at most ± 1 , while leaving g_i unaltered. Hence the new area \tilde{A} is bounded by

$$A - \frac{\ell}{2} \leq A - \frac{g_i}{2} \leq \tilde{A} \leq A + \frac{g_i}{2} \leq A + \frac{\ell}{2}.$$

Now consider the effect of perturbing all three points. First we perturb \mathbf{p}_3 and increase or decrease the area A by at most $\ell/2$. Perturbing \mathbf{p}_3 will also change g_2 and g_3 , but only by an absolute value of at most 1 (by the triangle inequality). As a consequence, the longest triangle edge length after perturbing the first vertex is at most $\ell + 1$. The perturbation of \mathbf{p}_2 causes then an absolute change of the area by at most $(\ell + 1)/2$. Finally we perturb \mathbf{p}_1 . The longest edge after perturbing the second vertex is at most $\ell + 2$. Hence, the absolute change of the area is at most $(\ell + 2)/2$, and the total change of A is bounded by

$$A' = A \pm \frac{\ell}{2} \pm \frac{\ell + 1}{2} \pm \frac{\ell + 2}{2} = A \pm \frac{3}{2}(\ell + 1).$$

□

A direct implication of Lemma 6 is the following:

LEMMA 7. *The drawing \mathbf{r} of G is crossing-free.*

Proof. In order to introduce a crossing, one of the triangles would have to flip over. Such a flip could happen only if there were a perturbation, by vectors of Euclidean length less than 1, such that the area of one triangle becomes zero. The area of the smallest triangle is $50n^6$, which is (for $n \geq 1$) larger than the “loss of area” caused by the perturbation, by Lemma 6. □

LEMMA 8. *Let $\bar{\omega}$ be the stress obtained by the atomic equilibrium stresses on \mathbf{r} and the coefficients α . For every interior edge, we have that*

$$\bar{\omega}_{ij} > 14n^6.$$

Proof. Let us first compute the “relative error” introduced by the rounding for the triangle areas. Let A be the area of a triangle in the drawing \mathbf{p} , and A' be the area of the same triangle in \mathbf{r} . Define

$$(3.4) \quad q := \frac{3}{7n^2}.$$

For any $n \geq 3$, we have $q > \frac{3(\ell_{\max}+1)}{2w(f^-)}$, and hence by Lemma 6,

$$(1 - q)A \leq A' \leq (1 + q)A.$$

The rounding of \mathbf{p} affects the atomic equilibrium stresses. Let $\bar{\omega}_{ij}^{kl}$ denote the stress ω_{ij}^{kl} after rounding. Because every atomic stress is a product of two triangle areas, we obtain the estimate

$$(1 - q)^2 \omega_{ij}^{kl} \leq \bar{\omega}_{ij}^{kl} \leq (1 + q)^2 \omega_{ij}^{kl}.$$

The stress ω is defined with help of the coefficients α and the atomic stresses. We mimic the proof of Lemma 5 and express the stress ω_{ij} as $\omega_{ij}^+ + \omega_{ij}^-$ and do the same for $\bar{\omega}_{ij}^{kl}$. Recall that $w(f^-) \geq 50n^6$. We use the notation of Lemma 5 and denote by f_1 and f_2 the two subfaces incident to (i, j) that were introduced by stacking v_i . The value of $\bar{\omega}_{ij}$ is bounded by

$$\begin{aligned} \bar{\omega}_{ij}^+ + \bar{\omega}_{ij}^- &> (1 - q)^2 \omega_{ij}^+ + (1 + q)^2 \omega_{ij}^- \\ &> (1 - q)^2 (w(f_1) + w(f_2)) \\ &\quad - (1 + q)^2 (w(f_1) + w(f_2) - 2w(f^-)) \\ &= -4q (w(f_1) + w(f_2)) + (1 + q)^2 2w(f^-) \\ &> -4q w(f_0) + 2w(f^-) \\ &> -85.8n^6 + 100n^6 \\ &> 14n^6. \end{aligned}$$

□

Because we have integer 2D coordinates, the atomic stresses are integral. Hence, integral coefficients α yield integer heights in the lifting. The coefficients α are by definition between 0 and 1. Therefore rounding the coefficients is not possible. However, when scaling all coefficients by a common scalar Y , we make the contributions from the atomic stresses big enough to afford a small perturbation of the coefficients.

LEMMA 9. Let $\dot{\omega}$ be defined by the atomic stresses $\bar{\omega}_{ij}^{xy}$ as

$$\dot{\omega}_{ij} = \sum_{\{i,j,k,l\} \in S} [Y\alpha_{ijkl}] \bar{\omega}_{ij}^{kl},$$

with S defined as for (2.2), and let $Y := 4n^2$. Then the stress $\dot{\omega}$ is a positive integer on every interior edge of G in \mathbf{r} .

Proof. Let (i, j) be an interior edge of G in \mathbf{r} . The stress $\bar{\omega}_{ij}$ is a linear combination of atomic stresses. One of the summands of the linear combination is positive, and the others are negative. Let $\alpha_{ijxy} \bar{\omega}_{ij}^{xy}$ be the positive summand. By rounding the coefficients α down, we decrease $\alpha_{ijxy} \bar{\omega}_{ij}^{xy}$. The effect for the negative atomic stresses can be ignored because these stresses will be altered in our favor. Thus we can estimate

$$\dot{\omega}_{ij} \geq (Y\alpha_{ijxy} - 1) \bar{\omega}_{ij}^{xy} + \sum_{\substack{\{i,j,k,l\} \in S \\ \{k,l\} \neq \{x,y\}}} Y\alpha_{ijkl} \bar{\omega}_{ij}^{kl} = Y\bar{\omega}_{ij} - \bar{\omega}_{ij}^{xy}.$$

For $n \geq 3$, we have by Lemma 5(1) that

$$(3.5) \quad \bar{\omega}_{ij}^{xy} \leq (1 + q)^2 \omega_{ij}^{xy} \leq (1 + q)^2 w(f_0) \leq 56n^8,$$

with q defined as in (3.4). Hence we have to pick Y such that $Y\bar{\omega}_{ij} > 56n^8$ holds. Because $\bar{\omega}_{ij}$ (after rounding for \mathbf{r}) is larger than $14n^6$ (by Lemma 8) it suffices to pick $Y = 4n^2$. □

We are now ready to prove the main theorem.

Proof of Theorem 1: The coordinates of the 2D drawing are nonzero integers and by construction smaller than $10n^4$. We compute the lifting induced by $\dot{\omega}$ following the method presented in Appendix A. Because $-\bar{\omega}_{12} \leq -(1 + q)^2 w(f_0) < -56n^8$, which follows from (3.5) and Lemma 5(2), we have

$$-\dot{\omega}_{12} < -4n^2 \bar{\omega}_{12} < 224n^{10}.$$

Let f_1 be the face incident to (v_1, v_2) that is not the outer face and let \mathbf{q}_i be the homogenized point for \mathbf{r}_i . The face f_1 is contained in a plane H_1 that sandwiches the lifting with the $z = 0$ plane. We describe H_1 by a function g_1 that assigns every homogenized 2D point \mathbf{t} its height on H_1 . The function g_1 is given as

$$g_1(\mathbf{t}) = \langle \mathbf{a}_1, \mathbf{t} \rangle = \dot{\omega}_{12} \langle \mathbf{q}_1 \times \mathbf{q}_2, \mathbf{t} \rangle.$$

By our choice of the boundary face, we have $\mathbf{q}_1 \times \mathbf{q}_2 = (y_1 - y_2, x_2 - x_1, 0)$. The largest z coordinate of the lifting is smaller than $g_1(\mathbf{q}_3)$, which is

$$g_1(\mathbf{q}_3) = \dot{\omega}_{12} \cdot 10n^4 \cdot 10n^4 < 224,000n^{18}.$$

□

4 Outlook

Let us briefly review some possible directions for further research. Our main theorem shows that a nontrivial class of 3D polytopes can be realized on a polynomial-size grid. It would be interesting to see whether the same is true for a larger class of polytopes. The most interesting class are the simplicial polytopes. These

polytopes have the property that a small perturbation of any geometric realization preserves the combinatorial structure of the polytope. (The same is not true for general polytopes.) Hence, it seems plausible that an algorithm that first realizes the polytope, and then rounds to grid points, could lead to an embedding algorithm with small coordinates. Notice that several aspects in our approach are more difficult for simplicial polytopes. First of all, it is in general not possible to create a 2D embedding that respects a given assignment of face areas; see, for example, [4]. Moreover, the space of equilibrium stresses is more complicated for general triangulations. However, if we could find a way to construct a 2D embedding, whose triangles are not too skinny, and whose minimum and maximum stresses are polynomially related, our *perturbation approach* should be applicable. It would also be interesting to see how the treewidth of G and the necessary grid size for the polytope embedding are related.

Another direction for further research addresses the problem of whether higher-dimensional stacked polytopes admit an embedding on a grid of polynomial volume. Unfortunately, the Maxwell–Cremona correspondence has no easy equivalence in higher dimensions [22]. On the other hand, our approach does not rely on the Maxwell–Cremona correspondence. To avoid the concept of equilibrium stresses, we could try to keep track of the creasing during the stacking operations more directly. This approach, however, makes the analysis more complicated. We have not checked whether such an analysis works out and whether it can be generalized to higher dimensions.

Our goal in this paper was to show that stacked polytopes can be embedded on a polynomial-size grid, preferring a simple presentation over the best possible grid size. By constructing integral coefficients α , we could easily show that the heights in the lifting are integers. However, a rounding scheme similar to the one we applied for the 2D drawing yields smaller coordinates. The analysis that the smaller perturbation along the z axis preserves convexity is more tedious. We also leave it for further study to exploit the dependencies between the size of the three different coordinate axes in the grid embedding.

Acknowledgements

We thank Günter Rote for suggesting this fascinating problem (originally posed by Günter M. Ziegler) to us.

References

- [1] D. M. Acketa and J. D. Žunić. On the maximal number of edges of convex digital polygons included into an $m \times m$ -grid. *J. Comb. Theory Ser. A*, 69(2):358–368, 1995.
- [2] G. E. Andrews. A lower bound for the volume of strictly convex bodies with many boundary lattice points. *Trans. Amer. Math. Soc.*, 99:272–277, 1961.
- [3] I. Bárány and G. Rote. Strictly convex drawings of planar graphs. *Documenta Math.*, 11:369–391, 2006.
- [4] T. C. Biedl and L. E. R. Velázquez. Drawing planar 3-trees with given face-areas. In Eppstein and Gansner [9], pages 316–322.
- [5] K. Buchin and A. Schulz. On the number of spanning trees a planar graph can have. In M. de Berg and U. Meyer, editors, *ESA (1)*, volume 6346 of *Lecture Notes in Computer Science*, pages 110–121. Springer, 2010.
- [6] M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions (preliminary version). In *12th Symposium on Computational Geometry*, pages 319–328, 1996.
- [7] G. Das and M. T. Goodrich. On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Computational Geometry: Theory and Applications*, 8(3):123–137, 1997.
- [8] P. Eades and P. Garvan. Drawing stressed planar graphs in three dimensions. In F.-J. Brandenburg, editor, *Graph Drawing*, volume 1027 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 1995.
- [9] D. Eppstein and E. R. Gansner, editors. *Graph Drawing, 17th International Symposium, GD 2009, Chicago, IL, USA, September 22–25, 2009. Revised Papers*, volume 5849 of *Lecture Notes in Computer Science*. Springer, 2010.
- [10] M. T. Goodrich and D. Strash. Succinct greedy geometric routing in the euclidean plane. In *ISAAC '09: Proceedings of the 20th International Symposium on Algorithms and Computation*, pages 781–791, Berlin, Heidelberg, 2009. Springer-Verlag.
- [11] J. E. Hopcroft and P. J. Kahn. A paradigm for robust geometric algorithms. *Algorithmica*, 7(4):339–380, 1992.
- [12] L. Lovasz. Steinitz representations of polyhedra and the colin de verdière number. *J. Comb. Theory, Ser. B*, 82:223–236, 2000.
- [13] J. C. Maxwell. On reciprocal figures and diagrams of forces. *Phil. Mag. Ser.*, 27:250–261, 1864.
- [14] A. Moitra and T. Leighton. Some results on greedy embeddings in metric spaces. In *FOCS*, pages 337–346. IEEE Computer Society, 2008.
- [15] S. Onn and B. Sturmfels. A quantitative Steinitz’ theorem. In *Beiträge zur Algebra und Geometrie*, volume 35, pages 125–129, 1994.
- [16] C. H. Papadimitriou and D. Ratajczak. On a conjecture related to geometric routing. *Theor. Comput. Sci.*, 344(1):3–14, 2005.
- [17] A. Ribó Mor. *Realization and Counting Problems for Planar Structures: Trees and Linkages, Polytopes and Polyominoes*. PhD thesis, Freie Universität Berlin, 2006.

- [18] A. Ribó Mor, G. Rote, and A. Schulz. Small grid embeddings of 3-polytopes. *Discrete and Computational Geometry*, 2010. accepted for publication, preprint <http://arxiv.org/abs/0908.0488>.
- [19] J. Richter-Gebert. *Realization Spaces of Polytopes*, volume 1643 of *Lecture Notes in Mathematics*. Springer, 1996.
- [20] J. Richter-Gebert and G. M. Ziegler. Realization spaces of 4-polytopes are universal. *Bull. Amer. Math. Soc.*, 32:403, 1995.
- [21] G. Rote, F. Santos, and I. Streinu. Expansive motions and the polytope of pointed pseudo-triangulations. *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, 25:699–736, 2003.
- [22] K. A. Rybnikov. Stresses and liftings of cell-complexes. *Discrete & Computational Geometry*, 21(4):481–517, 1999.
- [23] O. Schramm. Existence and uniqueness of packings with specified combinatorics. *Israel J. Math.*, 73:321–341, 1991.
- [24] A. Schulz. Drawing 3-polytopes with good vertex resolution. In Eppstein and Gansner [9], pages 33–44.
- [25] E. Steinitz. Polyeder und Raumeinteilungen. In *Encyclopädie der mathematischen Wissenschaften*, volume 3-1-2 (Geometrie), chapter 12, pages 1–139. B. G. Teubner, Leipzig, 1916.
- [26] R. E. Tarjan. Linking and cutting trees. In *Data Structures and Network Algorithms*, chapter 5, pages 59–70. Society for Industrial and Applied Mathematics, 1983.
- [27] T. Thiele. Extremalprobleme für Punktmengen. Master’s thesis, Freie Universität Berlin, 1991.
- [28] W. T. Tutte. Convex representations of graphs. *Proceedings London Mathematical Society*, 10(38):304–320, 1960.
- [29] W. T. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 13(52):743–768, 1963.
- [30] W. Whiteley. Motion and stresses of projected polyhedra. *Structural Topology*, 7:13–38, 1982.
- [31] H. Whitney. Congruent graphs and the connectivity of graphs. *Amer. J. Math.*, 54:150–168, 1932.
- [32] F. Zickfeld. *Geometric and Combinatorial Structures on Graphs*. PhD thesis, Technical University Berlin, December 2007.
- [33] G. M. Ziegler. *Lectures on Polytopes*. Springer, 1995.

Appendix A: Computing a Lifting from an Equilibrium Stress

Let G be a 3-connected planar graph with equilibrium stress ω . We follow the presentation of [19] and describe the induced lifting by defining a plane H_i for every face f_i . We define the *homogenized* coordinates for \mathbf{p}_i by $\mathbf{q}_i := (x_i, y_i, 1)^T$. The plane H_i is specified by the function $g(\mathbf{t})$ that assigns a height to every (homogenized) 2D point. Hence we can compute the height of a vertex v_i lying on face f_l by $h_i := g_l(\mathbf{q}_i)$. We define g_i as an inner product with a vector \mathbf{a}_i in \mathbb{R}^3 :

$$g_i: \mathbf{t} \mapsto \langle \mathbf{t}, \mathbf{a}_i \rangle.$$

The parameters \mathbf{a}_i can be computed by the following iterative method. We set $\mathbf{a}_0 = (0, 0, 0)^T$. Then we compute the remaining parameters face by face. This is achieved by selecting a face f_l that is adjacent to a face f_r for which we have already determined \mathbf{a}_r . Let (i, j) be the common edge of f_l and f_r . Assume that, in the 2D drawing, the face f_l lies left of the directed edge ij , and f_r lies right of it. The parameter for g_l can be computed by

$$\mathbf{a}_l = \omega_{ij}(\mathbf{q}_i \times \mathbf{q}_j) + \mathbf{a}_r.$$

If the scalars ω_{ij} define an equilibrium stress, the definition of the functions g_i is consistent. Furthermore, $g_r(\mathbf{q}_i) = g_l(\mathbf{q}_i)$ if v_i is incident to the faces f_l and f_r . A proof can be found in Richter-Gebert’s book [19, Chapter 13].