CHAPTER 1

Why are Geometric Representations Interesting?

To represent a graph geometrically is a natural goal in itself, since it provides visual access to the abstract structure of the graph. But, in addition, it is an important tool in the study of various graph properties, including their algorithmic aspects. To illustrate what this means, let us describe a few examples of increasing complexity.

1.1. Edge coloring

We start with a very simple application of a well-chosen geometric image. How many colors are needed to color the edges of a complete graph so that no two edges incident with the same node get the same color? The answer depends on the parity of the number of nodes n. If n is odd, then at most $(n-1)/2$ edges can use the same color, so we need at least

$$\binom{n}{2}\Big/\frac{n-1}{2} = n$$

colors. To show that this is sufficient, we consider the nodes as the vertices of a regular polygon. An edge and all diagonals parallel to it can be colored with the same color. Rotating this set about the center of the polygon, we get n such sets of edges, which together cover all edges of the graph. So they form an edge-coloring with n colors (Figure 1.1, left).

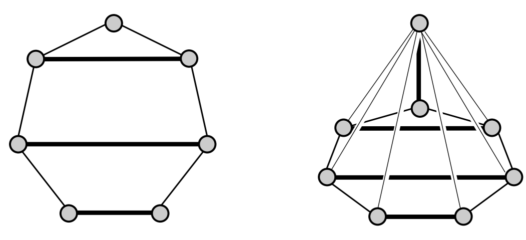

FIGURE 1.1. Optimal edge coloring of a complete graph with an odd and even number of nodes.

If n is odd, then the analogous computation gives a weaker lower bound of

$$\binom{n}{2}\Big/\frac{n}{2} = n-1$$

for the number of colors. Is this sufficient? To improve the construction by a similar use of rotations as for odd n, we step up to dimension 3. We represent the nodes as the vertices of a pyramid with an $(n-1)$-sided regular polygon as its base. For any edge e of the base, we can color the diagonals parallel to e as well as the edge connecting its apex to the base vertex opposite to e with the same color. Rotating this set again about the axis of the pyramid, we get a coloring of the edges of K_n with $n-1$ colors (Figure 1.1, right).

There are of course many other ways to solve this simple problem; but the above construction using geometry is perhaps the nicest.

1.2. Disjoint paths

Suppose that we have a 2-connected graph G and two specified nodes s and t. Two "requests" come in for two nodes x and y, and we have to find two disjoint paths connecting s and t to x and y (it does not matter which of x and y will be connected to s). This can be computed by one of zillions of flow or connectivity algorithms in reasonable time.

Now suppose that we have to compute such paths for many requests $\{x, y\}$. Do we have to repeat the computation each time? We can do much better if we use the following theorem: *Given a 2-connected graph and two specified nodes s and t, we can order all nodes so that s is first, t is last, and every other node v has a neighbor that comes earlier as well as a neighbor that comes later.* Such an ordering is called an *s-t numbering*. It is not hard to construct such a numbering (Exercise 1.2).

Once we know an s-t numbering, and a request $\{x, y\}$ comes in, it is trivial to find two disjoint paths: Let (say) x precede y in the ordering, then we can move from x to an earlier neighbor x', then to an even earlier neighbor x'' of x' etc. until we reach s. Similarly, we can move from y to a later neighbor y', then to an even later neighbor y'' of y' etc. until we reach t. This way we trace out two paths as requested.

The ordering can be thought of as representing the nodes of G by points on the line, and the easy procedure to find the two paths uses this geometric representation.

1.3. Hitting times

In this book, we will discuss some properties of random walks on a graph (as far as they are related to geometric representations). We'll also use the physical representation of a graph with the edges replaced by rubber bands. As a sampler, let us describe a very simple but nice connection.

Let a be a node of a connected graph G. We start a walk at $v^0 = a$. We select one of the neighbors v^1 of v^0 at random (every neighbor has the same probability of being selected), and we move to v^1. Then we move from v^1 to v^2 in the same way, etc. This way we get an infinite sequence of random nodes $(v^0, v^1, v^2, \dots)$, which we call a *random walk* on G.

Many important questions can be asked about a random walk (we are going to talk about some of them later). Perhaps the following is the simplest. Let b be another node of G. We define the *hitting time* $H(a, b)$ as the expected number of steps before a random walk, started at a, will reach b. (This number is known to be finite for a finite connected graph.)

There are many interesting questions you can ask about hitting times. To begin with, what are the hitting times on particular graphs like paths, cycles, or trees?

The following construction is very useful answering quite a few of these basic questions (of course, there are questions about hitting times whose answer requires a much more sophisticated approach). Consider the edges of the graph as rubber bands; these are ideal (or really high tech) rubber bands, contracting to zero length when not stretched, and not getting tangled. Attach a weight of $\deg(v)$ to each node v. Nail the node b to the wall and let the graph find its equilibrium (Figure 1.2).

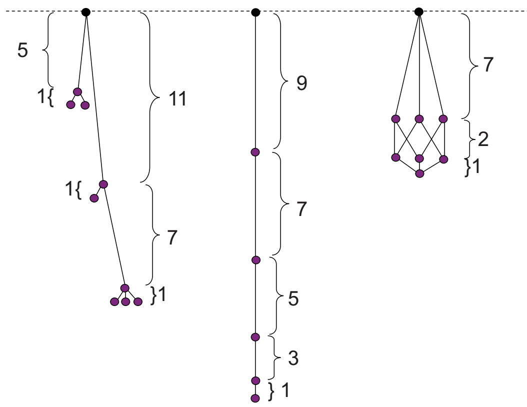

FIGURE 1.2. Hanging a tree, a path, and a 3-cube from a node. The graphs are horizontally distorted to show their structure. The stretching of the edges is easily computed using Hooke's Law.

The simple but useful fact about this construction is that *each node v will be at a distance of $H(v, b)$ below b.* This is not hard to prove noticing that the hitting times to b and the distances from b satisfy the same equations (see Exercise 1.4).

Using this geometric/physical picture, we can derive several interesting facts about hitting times. Applying the construction to a path with k nodes, we see that the k-th edge from the bottom is stretched to a length of $2k-1$ (this is the total weight of the nodes below it). Hence the total length of the path will be $1+3+\cdots+(2n-3) = (n-1)^2$; this is the hitting time from one endpoint of the path to the other. Or, using the symmetries of the skeleton of the 3-dimensional cube, we can see that the lowest edges are stretched to length 1 (3 edges carry a weight of 3), the next layer, to length 2, and the edges incident to b, to length 7. So the hitting time from a node to the opposite node is 10. Exercises 1.5–1.7 offer some further applications of this geometric construction.

1.4. Shannon capacity

The last two examples used just a single dimension; of course, one-dimensional geometry is not "really" geometry, and we better give a higher-dimensional example, which is substantially more involved. The following problem in information theory was raised by Claude Shannon, and it motivated the introduction of orthogonal representations [**Lovász 1979b**] and several of the results to be discussed in this book.

Consider a noisy channel through which we are sending messages composed of a finite alphabet V. There is an output alphabet U, and each $v \in V$, when transmitted through the channel, can come out as any element in a set $U_v \subseteq U$. Usually there is a probability distribution specified on each set U_v, telling us the probability with which v produces a given $u \in U_v$, but for the problem we want to discuss, these probabilities do not matter. As a matter of fact, the output alphabet will play no role either, except to tell us which pairs of input characters can be confused: those pairs (v, v') for which $U_v \cap U_{v'} \neq \emptyset$.

One way to model the problem is as follows: We consider V as the set of nodes of a graph, and connect two of them by an edge if they can be confused. This way we obtain a graph G, which we call the *confusion graph* of the alphabet. The maximum number of non-confusable messages of length 1 is the maximum number of nonadjacent nodes (the maximum size of a stable set) in the graph G, which we denote by $\alpha(G)$.

Now we consider longer messages, say of length k. We want to select as many of them as possible so that no two of them can possibly be confused. This means that for any two of these selected messages, there should be a position, where the two characters are not confusable. As we shall see, the number of words we can select grows as Θ^k for some $\Theta \geq 1$, which is called the *Shannon zero-error capacity* of the channel.

A simple and natural way to create such a set of words is to pick a non-confusable subset of the alphabet, and use only those words composed from this set. So if we have α non-confusable characters in our alphabet, then we can create α^k non-confusable messages of length k. But, as we shall see, making use of other characters in the alphabet we can create more! How much more, is the issue in this discussion.

Let us look at two simple examples (Figure 1.3).

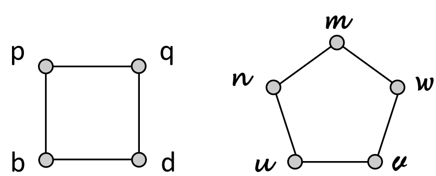

FIGURE 1.3. Two confusion graphs. In the alphabet $\{p, q, b, d\}$ two letters that are related by a reflection in a horizontal or vertical line are confusable, but not if they are related by two such reflection. The confusability graph of the alphabet $\{m, n, u, v, w\}$ is only convincing a little in handwriting, but this graph plays an important role in this book.

EXAMPLE 1.1. Consider the simple alphabet (p, q, d, b), where the pairs $\{p, q\}$, $\{q, d\}$, $\{d, b\}$ and $\{b, p\}$ are confusable (Figure 1.3, left). We can just keep p and d (which are not confusable), which allows us to create 2^k non-confusable messages of length k. On the other hand, if we use a word, then all the 2^k words obtained from it by replacing some occurrences of p and q by the other, as well as some occurrences of b and d by the other, are excluded. Hence the number of messages we can use is at most $4^k/2^k = 2^k$. ♦

EXAMPLE 1.2 (5-cycle). If we switch to alphabets with 5 characters, then we get a much more difficult problem. Let $V = \{m, n, u, v, w\}$ be our alphabet, with confusable pairs $\{m, n\}$, $\{n, u\}$, $\{u, v\}$, $\{v, w\}$ and $\{w, m\}$ (Figure 1.3, right; we refer to this example as the "pentagon"). Among any three characters there are two that can be confused, so we have only two non-confusable characters. Restricting the alphabet to two such characters (say, m and v), we get 2^k non-confusable messages of length k.

But we can do better: the following 5 messages of length two are non-confusable: mm, nu, uw, vn and wv. This takes some checking: for example, mm and nu cannot be confused, because their second characters, m and u, cannot be confused. If k is even, then we can construct $5^{k/2}$ non-confusable messages, by concatenating any $k/2$ of the above 5. This number grows like $(\sqrt{5})^k \approx 2.236^k$ instead of 2^k, a substantial gain! ♦

Can we do better by looking at longer messages (say, messages of length 10), and by some *ad hoc* method finding among them more that 5^5 non-confusable messages? We are going to show that we cannot, which means that the set of words composed of the above 5 messages of length 2 is optimal.

The trick is to represent the alphabet in a different way. Let us assign to each character $i \in V$ a vector $\mathbf{u}_i$ in some Euclidean space $\mathbb{R}^d$. If two characters are not confusable, then we represent them by orthogonal vectors.

If a subset of characters S is non-confusable, then the vectors $\mathbf{u}_i$ $(i \in S)$ are mutually orthogonal unit vectors, and hence for every unit vector $\mathbf{c}$,

$$\sum_{i\in S} (\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2 \le 1.$$

Hence $|S| \min_{i\in S}(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2 \le 1$, or

$$|S| \le \max_{i\in S} \frac{1}{(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2} \le \max_{i\in V} \frac{1}{(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2}.$$

So if we find a representation $\mathbf{u}$ and a unit vector $\mathbf{c}$ for which the squared products $(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2$ are all large (which means that the angels $\measuredangle(\mathbf{c}, \mathbf{u}_i)$ are all small), then we get a good upper bound on $|S|$.

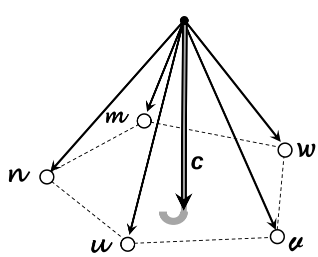

FIGURE 1.4. An umbrella representing the pentagon.

For the alphabet in Example 1.2 (the pentagon), we use the 3-dimensional vectors in Figure 1.4. To describe these, consider an umbrella with 5 ribs of unit length. Open it up to the point when nonconsecutive ribs are orthogonal. This way we get 5 unit vectors $\mathbf{u}_m, \mathbf{u}_n, \mathbf{u}_u, \mathbf{u}_v, \mathbf{u}_w$, assigned to the nodes of the pentagon so that each $\mathbf{u}_i$ forms the same angle with the "handle" $\mathbf{c}$ and any two nonadjacent nodes are labeled with orthogonal vectors. With some effort, one can compute that $(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2 = 1/\sqrt{5}$ for every i, and so we get that $|S| \le \sqrt{5}$ for every non-confusable set S. Since $|S|$ is an integer, this implies that $|S| \le 2$.

This is ridiculously much work to conclude that the 5-cycle does not contain 3 nonadjacent nodes! But the vector representation is very useful for handling longer messages. We define the *tensor product* of two vectors $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{R}^n$ and $\mathbf{v} = (v_1, \dots, v_m) \in \mathbb{R}^m$ as the vector

$$\mathbf{u} \circ \mathbf{v} = (u_1 v_1, \dots, u_1 v_m, u_2 v_1, \dots, u_2 v_m, \dots, u_n v_1, \dots, u_n v_m)^\mathsf{T} \in \mathbb{R}^{nm}.$$

It is easy to see that $|\mathbf{u} \circ \mathbf{v}| = |\mathbf{u}|\,|\mathbf{v}|$, and (more generally) if $\mathbf{u}, \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{v}, \mathbf{y} \in \mathbb{R}^m$, then $(\mathbf{u} \circ \mathbf{v})^\mathsf{T}(\mathbf{x} \circ \mathbf{y}) = (\mathbf{u}^\mathsf{T}\mathbf{x})(\mathbf{v}^\mathsf{T}\mathbf{y})$. For a $k \ge 1$, if we represent a message $i_1 \dots i_k$ by the vector $\mathbf{u}_{i_1} \circ \dots \circ \mathbf{u}_{i_k}$, then non-confusable messages will be represented by orthogonal vectors. Indeed, if $i_1 \dots i_k$ and $j_1 \dots j_k$ are not confusable, then there is at least one subscript r for which i_r and j_r are not confusable, hence $\mathbf{u}_{i_r}^\mathsf{T} \mathbf{u}_{j_r} = 0$, which implies that

$$(\mathbf{u}_{i_1} \circ \dots \circ \mathbf{u}_{i_k})^\mathsf{T}(\mathbf{u}_{j_1} \circ \dots \circ \mathbf{u}_{j_k}) = (\mathbf{u}_{i_1}^\mathsf{T} \mathbf{u}_{j_1}) \dots (\mathbf{u}_{i_k}^\mathsf{T} \mathbf{u}_{j_k}) = 0.$$

Using $\mathbf{c} \circ \dots \circ \mathbf{c}$ (k factors) as the "handle", we get that for any set S of non-confusable messages of length k,

$$|S| \le \max_{i_1,\dots,i_k} \frac{1}{\big((\mathbf{c} \circ \dots \circ \mathbf{c})^\mathsf{T}(\mathbf{u}_{i_1} \circ \dots \circ \mathbf{u}_{i_k})\big)^2} = \max_{i_1,\dots,i_k} \frac{1}{(\mathbf{c}^\mathsf{T}\mathbf{u}_{i_1})^2 \dots (\mathbf{c}^\mathsf{T}\mathbf{u}_{i_k})^2} = (\sqrt{5})^k.$$

So every set of non-confusable messages of length k has at most $(\sqrt{5})^k$ elements. We have seen that this bound can be attained, at least for even k. Thus we have established that *the Shannon zero-error capacity of the pentagon is* $\sqrt{5}$.

We will return to this topic in Sections 11.5.1 and 12.2, where the zero-error capacity problem will be discussed for general confusability graphs, in classical and quantum information theory.

1.5. Vector-labeled graphs or frameworks?

One of the most basic objects we study in this book is a graph whose nodes are labeled by vectors from a euclidean space $\mathbb{R}^d$. We can also think of these vectors as the positions of the nodes in $\mathbb{R}^d$. A mapping $\mathbf{u}: V \to \mathbb{R}^d$ can be thought of as a "drawing", or "embedding", or "geometric representation" of the set V in a Euclidean space. Most of the time, V will be the node set of a graph, and we should think of $\mathbf{u}_i$ as the position of node i; in this case, we think of the edges as straight line segments connecting the points in the corresponding position. The main point in this book is to relate geometric and graph-theoretic properties, so this way of visualizing is often very useful. On the other hand, all three phrases above are ambiguous, and we are going to use "vector labeling" in most of the formal statements. This is the computer science view: we have a graph and store additional information for each node (see Figure 1.5). A vector-labeled graph $(G, \mathbf{u})$ will also be called a *framework*, motivated by the important topic of rigidity.

The difference between "vector-labeled graphs" and "geometric representations" is a bit more than just different usage of words. In computer science (indeed, in any area that uses and analyzes data), one considers large tables where each row corresponds to an item of some sort, and the numbers in the row represent different data (for example, age, weight, height, income of a person, or frequencies of a word in various types of documents). Often the set of rows has a network structure (say, people who know each other or words with similar meaning can be considered "adjacent" in these networks).

Surprisingly, the geometry of the row vectors often contains important information about the data, and geometric manipulation can lead to a better handle on the data, even though these data had nothing to do with geometry. While obviously

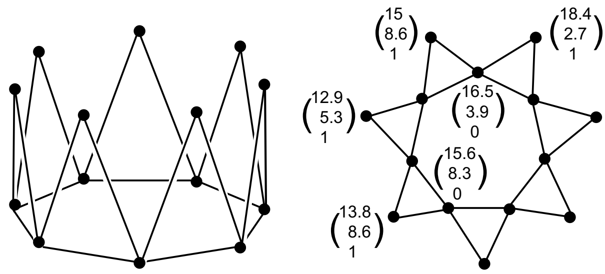

FIGURE 1.5. Two ways of looking at a graph with vector labeling (not all vector labels are shown).

closely related to our more mathematical material, we cannot treat in detail this emerging line of research.

While this is not strictly true, most of the time we can distinguish two kinds of vectors: those which are "true" vectors (best to think of them as geometric objects, like point positions or velocities), and those which are assignments of numbers, like weighting the nodes or edges of a graph. "True" vectors will be denoted by boldface characters. Mathematically, there is no difference of course.

EXERCISE 1.1. Prove that s-t numberings can be characterized as orderings of the nodes in which the nodes preceding any node induce a connected subgraph, and the same holds for the nodes following a given node.

EXERCISE 1.2. Design an algorithm that constructs an s-t numbering of a graph in polynomial time.

EXERCISE 1.3. Let us assign a real weight $f(e)$ to every edge of a transitive tournament on n nodes.
(a) For $v \in V$, let P_v be the set of all pairs (a,b) such that there is a directed path of length a leaving v along which the weights are monotone increasing, as well as a directed path of length b leaving v along which the weights are monotone non-increasing, starting with the same edge. Prove that the sets P_v are different.
(b) Prove that if $n \le \binom{p+q}{p}$, then either there is a directed path of length $p+1$ along which the weights are monotone increasing, or there is a directed path of length $q+1$ along which the weights are monotone non-increasing.[**Chvátal–Komlós 1971**]:

EXERCISE 1.4. (a) Prove that the hitting times to a given node of a connected graph G satisfy the equations

$$H(u,b) = 1 + \frac{1}{\deg(u)} \sum_{v \in N(u)} H(v,u), \quad (u \neq b).$$

(b) Prove that the distances to b in the construction in Section 1.3 satisfy the same equations. (c) Prove that this implies that the hitting times to b are equal to the distances from b.

EXERCISE 1.5. Compute the hitting time (a) between two nodes of a cycle of length n; (b) between two nodes of the skeleton of the dodecahedron; (c) from one vertex of the d-dimensional cube to the opposite one.

EXERCISE 1.6. Let G be a tree, and let $ij \in E$. Let k be the number of nodes of $G \setminus ij$ in the component containing i. Then the hitting time from i to j is the $2k-1$.

Exercise 1.7. Let G be a simple connected graph, and let $ij \in E$. Then $|H(i,j) - H(j,i)| \le n^2 - 3n + 3$, and this can be attained.

Exercise 1.8. Prove that for every graph with at most 5 nodes, with the exception of the pentagon, its Shannon capacity equals to the maximum cardinality of a stable set (mutually nonadjacent nodes) of the graph.

CHAPTER 3

Rubber Bands

3.1. Energy, forces, and centers of gravity

Let us start with an informal description. Let G be a connected graph. Replace the edges by ideal rubber bands (satisfying Hooke's Law). Think of the nodes in a nonempty subset $S \subseteq V$ as nailed to given positions in d-space (thinking of the plane, $d = 2$, will be enough for a while), but let the other nodes settle in equilibrium (Figure 3.1). (We are going to see that this equilibrium position is uniquely determined.) The nodes in S will be called *nailed*, and the other nodes, *free*. We call this equilibrium position of the nodes the *rubber band representation* of G in $\mathbb{R}^d$ extending the representation of the nailed nodes. We represent the edges by straight line segments (the rubber bands don't get tangled).

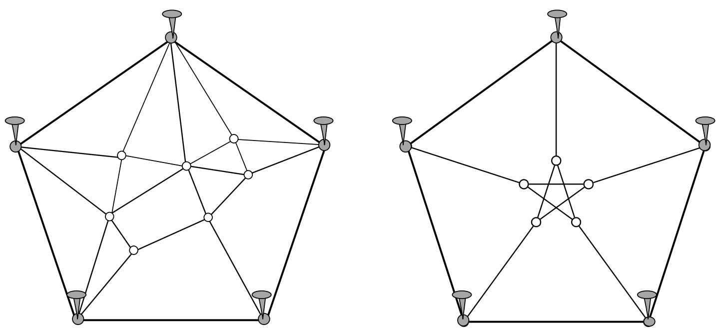

FIGURE 3.1. Rubber band representation of a planar graph and of the Petersen graph.

To make this precise, let $\mathbf{u}_i = (u_{i1}, \dots, u_{id})^\mathsf{T} \in \mathbb{R}^d$ be the position of node $i \in V$. By definition, $\mathbf{u}_i = \overline{\mathbf{u}}_i$ is prescribed for $i \in S$, but arbitrary for the remaining nodes. The *energy* of this representation is defined as

$$\mathcal{E}(\mathbf{u}) = \sum_{ij \in E} |\mathbf{u}_i - \mathbf{u}_j|^2 = \sum_{ij \in E} \sum_{k=1}^{d} (u_{ik} - u_{jk})^2. \tag{3.1}$$

We want to find the representation with minimum energy, subject to the boundary conditions:

$$\begin{aligned} &\text{minimize } \mathcal{E}(\mathbf{u}) \\ &\text{subject to } \mathbf{u}_i = \overline{\mathbf{u}}_i \text{ for all } i \in S. \end{aligned} \tag{3.2}$$

Note that while we use phrases like energy, the representation is defined in exact mathematical terms.

LEMMA 3.1. *The function* $\mathcal{E}: \mathbb{R}^{d\times(V\setminus S)} \to \mathbb{R}$ *is strictly convex.*

PROOF. In (3.1), every function $(u_{ik}-u_{jk})^2$ is convex, so $\mathcal{E}$ is convex. Furthermore, moving along an (affine) line in $\mathbb{R}^{d\times(V\setminus S)}$, this function is strictly convex unless $u_{ik}-u_{jk}$ remains constant along this line. If this applies to each coordinate of each edge, then moving along the line means parallel translation of the vectors $\mathbf{u}_i$, which is impossible if at least one node is nailed. □

It is trivial that if any of the vectors $\mathbf{u}_i$ tends to infinity, then $\mathcal{E}(\mathbf{u})$ tends to infinity (still assuming the boundary conditions 3.2 hold, where S is nonempty). Together with Lemma 3.1, this implies that the representation with minimum energy is uniquely determined. If $i \in V\setminus S$, then for the representation minimizing the energy, the partial derivative of $\mathcal{E}(\mathbf{u})$ with respect to any coordinate of $\mathbf{u}_i$ must be 0:

$$\sum_{j\in N(i)} (\mathbf{u}_i - \mathbf{u}_j) = 0 \qquad (i \in V\setminus S). \tag{3.3}$$

We can rewrite this as

$$\mathbf{u}_i = \frac{1}{\deg(i)} \sum_{j\in N(i)} \mathbf{u}_j. \tag{3.4}$$

This equation means that *every free node is placed in the center of gravity of its neighbors.* Equation (3.3) has a nice physical meaning: the rubber band connecting i and j pulls i with force $\mathbf{u}_j - \mathbf{u}_i$, so (3.3) states that the forces acting on i sum to 0 (as they should at the equilibrium). It is easy to see that (3.3) characterizes the equilibrium position.

We will return to the 1-dimensional case of rubber band representations in Chapter 4, studying harmonic functions on a graph. In those terms, equation (3.4) asserts that *every coordinate function is harmonic at every free node.*

It will be useful to extend the rubber band construction to the case when the edges of G have arbitrary positive weights (or "strengths"). Let $c_{ij} > 0$ denote the strength of the edge ij. We define the energy function of a representation $\mathbf{u}$ by

$$\mathcal{E}_c(\mathbf{u}) = \sum_{ij\in E} c_{ij} |\mathbf{u}_i - \mathbf{u}_j|^2. \tag{3.5}$$

The simple arguments above remain valid: $\mathcal{E}_c$ is strictly convex if at least one node is nailed, there is a unique optimum, and for the optimal representation every $i \in V\setminus S$ satisfies

$$\sum_{j\in N(i)} c_{ij}(\mathbf{u}_i - \mathbf{u}_j) = 0. \tag{3.6}$$

This can be rewritten as

$$\mathbf{u}_i = \frac{1}{\sum_{j\in N(i)} c_{ij}} \sum_{j\in N(i)} c_{ij}\mathbf{u}_j. \tag{3.7}$$

Thus $\mathbf{u}_i$ is no longer at the center of gravity of its neighbors, but it is still a convex combination of them with positive coefficients. In particular, it is in the relative interior of the convex hull of its neighbors.

3.2. Rubber bands, planarity and polytopes

3.2.1. How to draw a graph? The rubber band method was first analyzed in [**Tutte 1963**]. In this classical paper he describes how to use "rubber bands" to draw a 3-connected planar graph with straight edges and convex countries.

Let G be a 3-connected planar graph, and let p_0 be any country of it. Let C_0 be the cycle bounding p_0. Let us nail the nodes of C_0 to the vertices of a convex polygon P_0 in the plane, in the appropriate cyclic order, and let the rest find its equilibrium. We draw the edges of G as straight line segments connecting the appropriate endpoints. Figure 3.2 shows the rubber band representation of the skeletons of the five platonic bodies.

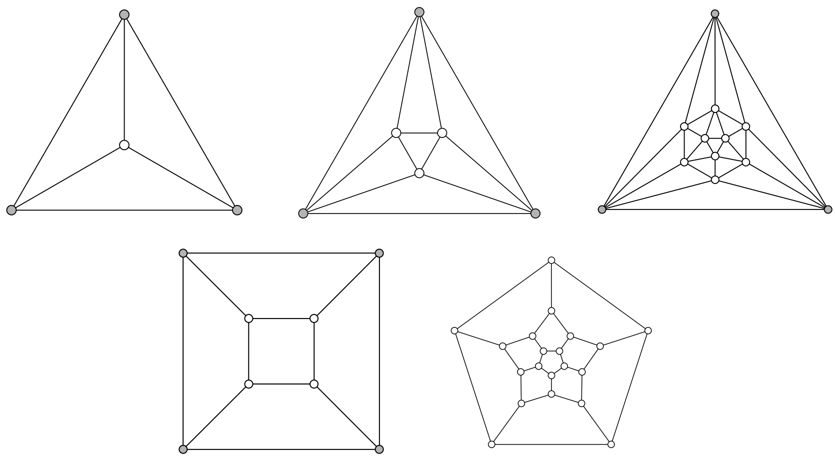

FIGURE 3.2. Rubber band representations of the skeletons of platonic bodies

By the above, we know that each node not on C_0 is positioned at the center of gravity of its neighbors. Tutte's main result about this representation is the following:

THEOREM 3.2. *If G is a simple 3-connected planar graph, then every rubber band representation of G (with the nodes of a particular country p_0 nailed to a convex polygon) gives a straight-line embedding of G in the plane. In addition, each country is a convex polygon.*

PROOF. Let $\mathbf{u}: V \to \mathbb{R}^2$ be this rubber band representation of G. Let ℓ be a line intersecting the interior of the polygon P_0, and let U_0, U_1 and U_2 denote the sets of nodes of G mapped on ℓ and on the two (open) sides of ℓ, respectively. The key to the proof is the following claim.

Claim 1. *The sets U_1 and U_2 induce connected subgraphs of G.*

Let us prove this for U_1. Clearly the nodes of the outer cycle p_0 in U_1 form a (nonempty) path P_1. We may assume that ℓ does not go through any node (by shifting it very little in the direction of U_1) and that it is not parallel to any line connecting two distinct positions (by rotating it with a small angle). Let $a \in U_1 \setminus V(C_0)$, we show that it is connected to P_1 by a path in U_1. Since $\mathbf{u}_a$ is a convex combination of the positions of its neighbors, it must have a neighbor a_1

such that $\mathbf{u}_{a_1}$ is in U_1 and at least as far away from ℓ as $\mathbf{u}_a$. By our assumption that ℓ is not parallel to any edge, either $\mathbf{u}_{a_1}$ is strictly farther from ℓ than $\mathbf{u}_a$, or $\mathbf{u}_{a_1} = \mathbf{u}_a$.

At this point, we have to deal with an annoying degeneracy. There may be several nodes represented by the same vector $\mathbf{u}_a$ (later it will be shown that this does not occur). Consider all nodes represented by $\mathbf{u}_a$, and the connected component H containing a of the subgraph of G induced by these nodes. If H contains a nailed node, then it contains a path from a to P_1, all in U_1. Else, there must be an edge connecting a node $a' \in V(H)$ to a node outside H (since G is connected). Since the system is in equilibrium, a' must have a neighbor a_1 such that $\mathbf{u}_{a_1}$ is farther away from ℓ than $\mathbf{u}_a = \mathbf{u}_{a'}$ (here we use again that no edge is parallel to ℓ). Hence $a_1 \in U_1$, and thus a is connected to a_1 by a path in U_1.

Either a_1 is nailed (and we are done), or we can find a node $a_2 \in U_1$ such that a_1 is connected to a_2 by a path in U_1, and $\mathbf{u}_{a_2}$ is farther from ℓ than $\mathbf{u}_{a_1}$, etc. This way we get a path Q in G that starts at a, stays in U_1, and eventually must hit P_1. This proves the claim (Figure 3.3).

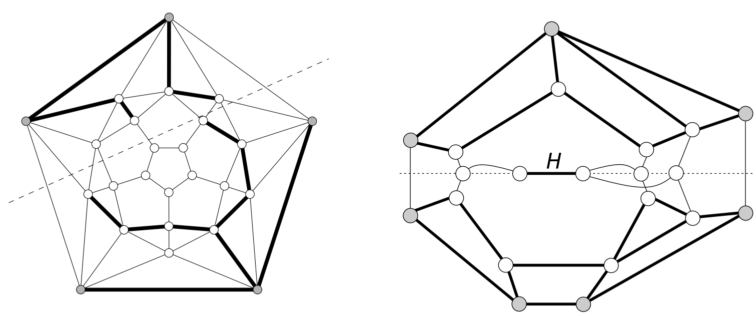

FIGURE 3.3. Left: every line cuts a rubber band representation into connected parts. Right: Each node on a line must have neighbors on both sides of the line.

Next, we exclude some possible degeneracies. (We are not assuming any more that no edge is parallel to ℓ: this assumption could be made for the proof of Claim 1 only.)

Claim 2. *Every node $u \in U_0$ has neighbors both in U_1 and U_2.*

This is trivial if $u \in V(C_0)$, so suppose that u is a free node. If u has a neighbor in U_1, then it must also have a neighbor in U_2; this follows from the fact that $\mathbf{u}_u$ is the center of gravity of the points representing its neighbors. So it suffices to prove that not all neighbors of u are contained in U_0.

Let T be the set of nodes $u \in U_0$ with $N(u) \subseteq U_0$, and suppose that this set is nonempty. Consider a connected component H of $G[T]$ (H may be a single node), and let S be the set of neighbors of H outside H. Since $V(H) \cup S \subseteq U_0$, the set $V(H) \cup S$ cannot contain all nodes, and hence S is a cutset. Thus $|S| \geq 3$ by 3-connectivity.

If $a \in S$, then $a \in U_0$ by the definition of S, but a has a neighbor not in U_0, and so it has neighbors in both U_1 and U_2 by the argument above (see Figure 3.3). The set $V(H)$ induces a connected graph by definition, and U_1 and U_2 induce connected subgraphs by Claim 1. So we can contract these sets to single nodes. These three nodes will be adjacent to all nodes in S. So we have contracted G to $K_{3,3}$, which is a contradiction, since G is planar. This proves Claim 2.

Claim 3. *Every country has at most two nodes in U_0.*

Suppose that $a, b, c \in U_0$ are nodes of a country p. Clearly $p \neq p_0$. Let us create a new node d and connect it to a, b and c; the resulting graph G' is still planar. On the other hand, the same argument as in the proof of Claim 1 (with $V(H) = d$ and $S = \{a, b, c\}$) shows that G' has a $K_{3,3}$ minor, which is a contradiction.

Claim 4. *Let p and q be the two countries sharing an edge ab, where $a, b \in U_0$. Then $V(p_1) \setminus \{a, b\} \subseteq U_1$ and $V(p_2) \setminus \{a, b\} \subseteq U_2$ (or the other way around).*

Suppose not, then p has a node $c \neq a, b$ and q has a node $d \neq a, b$ such that (say) $c, d \in U_1$. (Note that $c, d \notin U_0$ by Claim 3.) By Claim 1, there is a path P in U_1 connecting c and d (Figure 3.4). Claim 2 implies that both a and b have neighbors in U_2, and again Claim 1, these can be connected by a path in U_2. This yields a path P' connecting a and b whose inner nodes are in U_2. By their definition, P and P' are node-disjoint. But look at any planar embedding of G: the edge ab, together with the path P', forms a Jordan curve that does not go through c and d, but separates them, so P cannot exist.

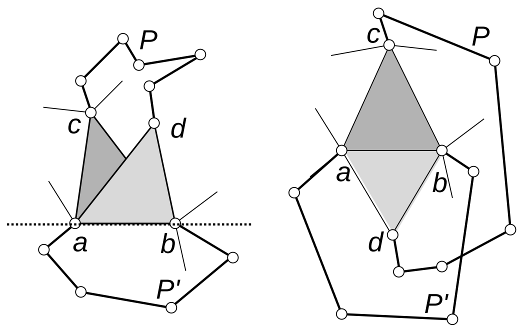

FIGURE 3.4. Two adjacent countries having nodes on the same side of ℓ in the rubber band representation (left), and the supposedly disjoint paths in the planar embedding (right).

Claim 5. *The boundary of every country q is mapped onto a convex polygon P_q.*

This is immediate from Claim 4, since no edge of a country, extended to a line, can intersect its interior.

Claim 6. *The interiors of the polygons P_q ($q \neq p_0$) are disjoint.*

Let $\mathbf{x}$ be a point inside P_{p_0}, we want to show that it is covered by one P_q only. Clearly we may assume that $\mathbf{x}$ is not on the image of any edge. Draw a line through $\mathbf{x}$ that does not go through the position of any node, and see how many times its points are covered by interiors of such polygons. As we enter P_{p_0}, this number is clearly 1. Claim 4 says that as the line crosses an edge, this number does not change. So $\mathbf{x}$ is covered exactly once.

Now the proof is essentially finished. Suppose that the images of two edges have a common point (other than their common endpoints). Then two of the countries incident with them would have a common interior point, which is a contradiction except if these countries are the same, and the two edges are consecutive edges of this country. □

Before going on, let's make a couple of remarks about this drawing method. The key step, namely Claim 1, is very similar to a basic fact concerning convex polytopes, mentioned in Appendix C.1, that every hyperplane intersecting the interior of the polytope cuts the skeleton into connected parts. Note that the proof of Claim 1 did not make use of the planarity of G (see Exercise 3.4, and also Section 16.5).

Tutte's method, as described above, is a very efficient procedure to find straight-line embeddings of 3-connected planar graphs in the plane. These embeddings look nice for many graphs (as our figures show), but they may have bad parts, like points getting too close. Figure 3.5 shows a simple situation in which positions of nodes get exponentially close to each other and to the midpoint of an edge. You may play with the edge weights, but finding a good weighting adds substantially to the algorithmic cost.

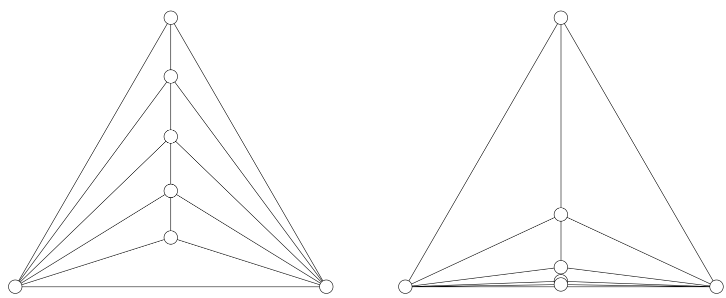

FIGURE 3.5. The rubber band representation can lead to crowding of the nodes. Each node on the middle line will be placed at or below the center of gravity of the triangle formed by the lower edge and the node immediately above it. So the distance between the k-th node from the top and the lower edge decreases faster than 3^{-k}.

A reasonable way to exclude nodes being positioned too close is to require that their coordinates are integers. Then the question is, of course, how to minimize these coordinates. In which rectangles $[0,a]\times[0,b]$ can every planar graph on n nodes be squeezed in so that we still get a straight line embedding? It turns out that this can be achieved with $a,b=O(n)$ [**Fraysseix–Pach–Pollack 1990**, **Schnyder 1990**], with several good bounds on the appropriate pairs (a,b) [**Chrobak–Nakano 1998**].

Representing edges by straight lines is not always the most important format of useful drawings. In some very important applications of planar embedding, most notably the design of integrated circuits (chips), the goal is to embed the graph in a grid, so that the nodes are drawn on gridpoints, and the edges are drawn as zig-zagging grid paths. (Of course, we must assume that no degree is larger than 4.) This way all nonzero distances between nodes and/or edges are automatically at least 1 (the edge-length of the grid). Besides trying to minimize the size or area of the grid in which the embedding lies, it is very natural (and practically important) to minimize the number of bends. The good news is that this can be achieved within a very reasonable area ($O(n^2)$) and with $O(n)$ bends [**Tamassia 1987**, **Tamassia–Tollis 2014**]. See the Handbook of Graph Drawing and Visualization [**Tamassia 2014**] for details.

3.2.2. How to lift a graph? An old construction of Cremona and Maxwell can be used to "lift" Tutte's rubber band representation to a Steinitz representation. We begin with analyzing the reverse procedure: projecting a convex polytope onto a face. This procedure is similar to the construction of the Schlegel diagram from Section 2.2, but we consider orthogonal projection instead of projection from a point.

Let P be a convex 3-polytope, let F be one of its faces, and let Σ be the plane containing H. Suppose that for every vertex $\mathbf{v}$ of P, its orthogonal projection onto H is an interior point of F; we say that the polytope P is *straight over the face* F.

THEOREM 3.3. (a) *Let P be a 3-polytope that is straight over its face F, and let G be the orthogonal projection of the skeleton of P onto the plane of F. Then we can assign positive strengths to the edges of G so that G will be the rubber band representation of the skeleton with the vertices of F nailed.*

(b) *Let G be a 3-connected planar graph, and let T be a triangular country of G, and let Δ be a triangle in a plane Σ. Then there is a convex polytope P in 3-space such that T is a face of P, and the orthogonal projection of P onto the plane Σ gives the rubber band representation of G obtained by nailing T to Δ.*

In other words (b) says that we can assign a number $\eta_i \in \mathbb{R}$ to each node $i \in V$ such that $\eta_i = 0$ for $i \in V(T)$, $\eta_i > 0$ for $i \in V \setminus V(T)$, and the mapping

$$i \mapsto \mathbf{v}_i = \begin{pmatrix} \mathbf{u}_i \\ \eta_i \end{pmatrix} = \begin{pmatrix} u_{i1} \\ u_{i2} \\ \eta_i \end{pmatrix} \in \mathbb{R}^3$$

is a Steinitz representation of G.

EXAMPLE 3.4 (Triangular Prism). Consider the rubber band representation of a triangular prism (or of a triangular frustum, a truncated tetrahedron) in Figure 3.6, left. If this is an orthogonal projection of a convex polyhedron, then the lines of the three edges pass through one point: the point of intersection of the planes of the three quadrangular faces. It is easy to see that this condition is necessary and sufficient for the picture to be a projection of a triangular frustum. To see that it is satisfied by a rubber band representation, it suffices to note that the inner triangle is in equilibrium, and this implies that the lines of action of the forces acting on it must pass through one point. ♦

Now we are ready to prove theorem 3.3.

PROOF. (a) Let's call the plane of the face F "horizontal", spanned by the first two coordinate axes, and the third coordinate direction "vertical", so that the polytope is "above" the plane of F. For each face p, let $\mathbf{g}_p$ be a normal vector. Since no face is vertical, no normal vector $\mathbf{g}_p$ is horizontal, and hence we can normalize $\mathbf{g}_p$ so that its third coordinate is 1. Clearly for each face p, $\mathbf{g}_p$ will be an outer normal, except for $p = F$, when $\mathbf{g}_p$ is an inner normal (Figure 3.6).

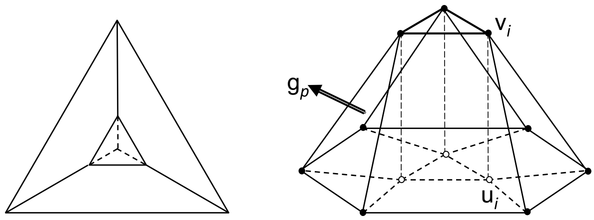

FIGURE 3.6. Left: The rubber band representation of a triangular prism is the projection of a polytope. Right: Vertical projection of a polytope into one of its faces.

Write $\mathbf{g}_p = \binom{\mathbf{h}_p}{1}$. Let ij be any edge of G, and let p and q be the two countries on the left and right of ij. Then

$$(\mathbf{h}_p - \mathbf{h}_q)^{\mathsf{T}}(\mathbf{u}_i - \mathbf{u}_j) = 0. \tag{3.8}$$

Indeed, both $\mathbf{g}_p$ and $\mathbf{g}_q$ are orthogonal to the edge $\mathbf{v}_i\mathbf{v}_j$ of the polytope, and therefore so is their difference, and

$$(\mathbf{h}_p - \mathbf{h}_q)^{\mathsf{T}}(\mathbf{u}_i - \mathbf{u}_j) = \begin{pmatrix} \mathbf{h}_p - \mathbf{h}_q \\ 0 \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \mathbf{u}_i - \mathbf{u}_j \\ \eta_i - \eta_j \end{pmatrix} = (\mathbf{g}_p - \mathbf{g}_q)^{\mathsf{T}}(\mathbf{v}_i - \mathbf{v}_j) = 0.$$

We have $\mathbf{h}_T = 0$, since the face T is horizontal.

Let R denote the counterclockwise rotation in the plane by 90°, then it follows that $\mathbf{h}_p - \mathbf{h}_q$ is parallel to $R(\mathbf{u}_j - \mathbf{u}_i)$, and so there are real numbers c_{ij} such that

$$\mathbf{h}_p - \mathbf{h}_q = c_{ij} R(\mathbf{u}_j - \mathbf{u}_i). \tag{3.9}$$

We claim that $c_{ij} > 0$. Let k be any node on the boundary of p different from i and j. Then $\mathbf{u}_k$ is to the left from the edge ij, and hence

$$(\mathbf{u}_k - \mathbf{u}_i)^{\mathsf{T}} R(\mathbf{u}_j - \mathbf{u}_i) > 0. \tag{3.10}$$

Going up to the space, convexity implies that the point $\mathbf{v}_k$ is below the plane of the face q, and hence $\mathbf{g}_q^{\mathsf{T}}\mathbf{v}_k < \mathbf{g}_q^{\mathsf{T}}\mathbf{v}_i$. Since $\mathbf{g}_p^{\mathsf{T}}\mathbf{v}_k = \mathbf{g}_p^{\mathsf{T}}\mathbf{v}_i$, this implies that

$$c_{ij}\big(R(\mathbf{u}_j - \mathbf{u}_i)\big)^{\mathsf{T}}(\mathbf{u}_k - \mathbf{u}_i) = (\mathbf{h}_p - \mathbf{h}_q)^{\mathsf{T}}(\mathbf{u}_k - \mathbf{u}_i) = (\mathbf{g}_p - \mathbf{g}_q)^{\mathsf{T}}(\mathbf{v}_k - \mathbf{v}_i) > 0. \tag{3.11}$$

Comparing with (3.10), we see that $c_{ij} > 0$.

To complete the proof of (a), we argue that the projection of the skeleton is indeed a rubber band embedding with strengths c_{ij}, with F nailed. We want to prove that every free node i is in equilibrium, i.e.,

$$\sum_{j \in N(i)} c_{ij}(\mathbf{u}_j - \mathbf{u}_i) = 0. \tag{3.12}$$

Using the definition of c_{ij}, it suffices to prove that

$$\sum_{j \in N(i)} c_{ij} R(\mathbf{u}_j - \mathbf{u}_i) = \sum_{j \in N(i)} (\mathbf{h}_{p_j} - \mathbf{h}_{q_j}) = 0,$$

where p_j is the face to the left and q_j is the face to the right of the edge ij. But this is clear, since every term occurs once with positive and once with negative sign.

Let us make a remark here that will be needed later. Using that not only $\mathbf{g}_p - \mathbf{g}_q$, but also $\mathbf{g}_p$ is orthogonal to $\mathbf{v}_j - \mathbf{v}_i$, we get that

$$0 = \mathbf{g}_p^{\mathsf{T}}(\mathbf{v}_j - \mathbf{v}_i) = \mathbf{h}_p^{\mathsf{T}}(\mathbf{u}_j - \mathbf{u}_i) + \eta_j - \eta_i,$$

and hence

$$\eta_j - \eta_i = -\mathbf{h}_p^\mathsf{T}(\mathbf{u}_j - \mathbf{u}_i). \tag{3.13}$$

(b) The proof consists of going through the steps of the proof of part (a) in reverse order: given the Tutte representation, we first reconstruct the vectors $\mathbf{h}_p$ so that all equations (3.8) are satisfied, then using these, we reconstruct the numbers η_i so that equations (3.13) are satisfied. It will not be hard to verify then that we get a Steinitz representation.

We need a little preparation to deal with edges on the boundary triangle. Recall that we can think of $\mathbf{F}_{ij} = c_{ij}(\mathbf{u}_j - \mathbf{u}_i)$ as the force with which the edge ij pulls its endpoint i. Equilibrium means that for every free node i,

$$\sum_{j\in N(i)} \mathbf{F}_{ij} = 0. \tag{3.14}$$

This does not hold for the nailed nodes, but we can modify the definition of $\mathbf{F}_{ij}$ along the three boundary edges so that $\mathbf{F}_{ij}$ remains parallel to the edge $\mathbf{u}_j - \mathbf{u}_i$ and (3.14) will hold for *all* nodes (this is the only point where we use that the outer country is a triangle). The only complication is that if we write $\mathbf{F}_{ij} = c_{ij}(\mathbf{u}_j - \mathbf{u}_i)$ for the boundary edges, then we have to allow negative strengths c_{ij}; this will cause not trouble in the computations. The existence of such forces $\mathbf{F}_{ij}$ is natural by a physical argument: let us replace the outer edges by rigid bars, and remove the nails. The whole framework will remain in equilibrium, so appropriate forces must act in the edges ab, bc and ac to keep balance. To translate this to mathematics, one has to work a little; this is left to the reader as Exercise 3.5.

We claim that we can choose vectors $\mathbf{h}_p$ for all countries p so that

$$\mathbf{h}_p - \mathbf{h}_q = R\mathbf{F}_{ij} \tag{3.15}$$

if ij is any edge and p and q are the two countries on its left and right. This follows by a "potential argument", which will be used several times in our book. Starting with $\mathbf{h}_T = 0$, and moving from country to adjacent country, equation 3.15 will determine the value of $\mathbf{h}_p$ for every country p. What we have to show is that we do not run into contradiction, i.e., if we get to the same country p in two different ways, then we get the same vector $\mathbf{h}_p$. This is equivalent to saying that if we walk around a closed cycle of countries, then the total change in the vector $\mathbf{h}_p$ is zero. It suffices to verify this when we move around countries incident with a single node. In this case, the condition boils down to

$$\sum_{i\in N(j)} R\mathbf{F}_{ij} = 0,$$

which follows by (3.14). This proves that the vectors $\mathbf{h}_p$ are well defined.

Second, we construct numbers η_i satisfying (3.13) by a similar argument (just working on the dual graph). We set $\eta_i = 0$ if i is a node of the unbounded country. Equation (3.13) tells us what the value at one endpoint of an edge must be, if we have it for the other endpoint.

One complication is that (3.13) gives two conditions for each difference $\eta_i - \eta_j$, depending on which country incident with it we choose. But if p and q are the two countries incident with the edge ij, then

$$\mathbf{h}_p^\mathsf{T}(\mathbf{u}_j - \mathbf{u}_i) - \mathbf{h}_q^\mathsf{T}(\mathbf{u}_j - \mathbf{u}_i) = (\mathbf{h}_p - \mathbf{h}_q)^\mathsf{T}(\mathbf{u}_j - \mathbf{u}_i) = (R\mathbf{F}_{ij})^\mathsf{T}(\mathbf{u}_j - \mathbf{u}_i) = 0,$$

since $\mathbf{F}_{ij}$ is parallel to $\mathbf{u}_i - \mathbf{u}_j$ and so $R\mathbf{F}_{ij}$ is orthogonal to it. Thus the two conditions on the difference $\eta_i - \eta_j$ are the same.

As before, equation (3.13) determines the values η_i, starting with $\eta_a = 0$. To prove that it does not lead to a contradiction, it suffices to prove that the sum of changes is 0 if we walk around a country p. In other words, if C is the cycle bounding a country p (oriented, say, clockwise), then

$$\sum_{\overrightarrow{ij} \in E(C)} \mathbf{h}_p^{\mathsf{T}}(\mathbf{u}_j - \mathbf{u}_i) = 0,$$

which is clear. It is also clear that $\eta_b = \eta_c = 0$.

Now define $\mathbf{v}_i = \binom{\mathbf{u}_i}{\eta_i}$ for every node i, and $\mathbf{g}_p = \binom{\mathbf{h}_p}{1}$ for every country p. It remains to prove that $i \mapsto \mathbf{v}_i$ maps the nodes of G onto the vertices of a convex polytope, so that edges go to edges and countries go to facets. We start with observing that if p is a country and ij is an edge of p, then

$$\mathbf{g}_p^{\mathsf{T}}\mathbf{v}_i - \mathbf{g}_p^{\mathsf{T}}\mathbf{v}_j = \mathbf{h}_p^{\mathsf{T}}(\mathbf{u}_i - \mathbf{u}_j) + (\eta_i - \eta_j) = 0,$$

and hence there is a scalar α_p so that all nodes of p are mapped onto the hyperplane $\mathbf{g}_p^{\mathsf{T}}\mathbf{x} = \alpha_p$. We know that the image of p under $i \mapsto \mathbf{u}_i$ is a convex polygon, and so the same follows for the map $i \mapsto \mathbf{v}_i$.

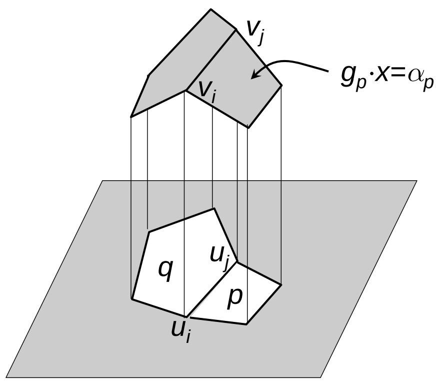

FIGURE 3.7. Lifting a rubber band representation to a polytope.

To conclude, it suffices to prove that if ij is any edge, then the two convex polygons obtained as images of countries incident with ij “bend” in the right way; more exactly, let p and q be the two countries incident with ij, and let Q_p and Q_q be two corresponding convex polygons (see Figure 3.7). We claim that Q_p lies on the same side of the plane $\mathbf{g}_p^{\mathsf{T}}\mathbf{x} = \alpha_p$ as the bottom face. Let $\mathbf{v}_k$ be any vertex of the polygon Q_q different from $\mathbf{v}_i$ and $\mathbf{v}_j$. We want to show that $\mathbf{g}_p^{\mathsf{T}}\mathbf{v}_k < \alpha_p$. Indeed,

$$\mathbf{g}_p^{\mathsf{T}}\mathbf{v}_k - \alpha_p = \mathbf{g}_p^{\mathsf{T}}\mathbf{v}_k - \mathbf{g}_p^{\mathsf{T}}\mathbf{v}_i = \mathbf{g}_p^{\mathsf{T}}(\mathbf{v}_k - \mathbf{v}_i) = (\mathbf{g}_p - \mathbf{g}_q)^{\mathsf{T}}(\mathbf{v}_k - \mathbf{v}_i)$$

(since both $\mathbf{v}_k$ and $\mathbf{v}_i$ lie on the plane $\mathbf{g}_q^{\mathsf{T}}\mathbf{x} = \alpha_q$),

$$= \begin{pmatrix} \mathbf{h}_p - \mathbf{h}_q \\ 0 \end{pmatrix}^{\mathsf{T}} (\mathbf{v}_k - \mathbf{v}_i) = (\mathbf{h}_p - \mathbf{h}_q)^{\mathsf{T}}(\mathbf{u}_k - \mathbf{u}_i) = (R\mathbf{F}_{ij})^{\mathsf{T}}(\mathbf{u}_k - \mathbf{u}_i) < 0$$

(since $\mathbf{u}_k$ lies to the right from the edge $\mathbf{u}_i\mathbf{u}_j$). □

Theorem 3.3 proves Steinitz's theorem in the case when the graph has a triangular country. If this is not the case, then Proposition 2.1 implies that the graph has a node of degree 3, which means that the dual graph has a triangular country. So we can represent the dual graph as the skeleton of a 3-polytope. Choosing the origin in the interior of this polytope, and considering its polar, we obtain a representation of the original graph.

3.3. Rubber bands and connectivity

Rubber band representations can be related to graph connectivity, and this relation can be used to give a test for k-connectivity of a graph [**Linial–Lovász–Wigderson 1988**]. It turns out that connectivity of the underlying graph is related to the degeneracy of the representation—a notion various versions of which will be crucial in many parts of this book.

3.3.1. Degeneracy: essential and nonessential. We start with a discussion of what causes degeneracy in rubber band embeddings. Consider the two graphs in Figure 3.8. It is clear that if we nail the nodes on the convex hull, and then let the rest find its equilibrium, then there will be a degeneracy: the grey nodes will all move to the same position. However, the reasons for this degeneracy are different: In the first case, it is due to symmetry; in the second, it is due to the node that separates the grey nodes from the rest, and thereby pulls them onto itself.

One can distinguish the two kinds of degeneracy in another way: In the first graph, the strengths of the rubber bands must be strictly equal; varying these strengths it is easy to break the symmetry and thereby get rid of the degeneracy. However, in the second graph, no matter how we change the strengths of the rubber bands (as long as they remain positive), the grey nodes will always be pulled together into one point.

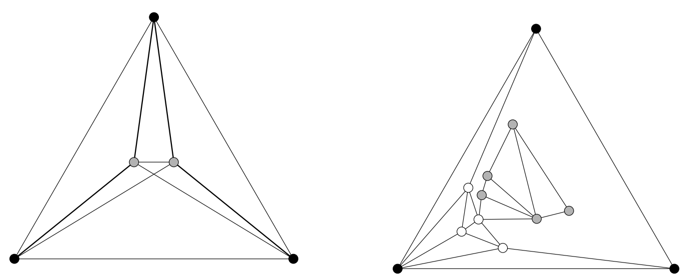

FIGURE 3.8. Two reasons why two or more nodes end up on top of each other: symmetry, or a separating node

Figure 3.9 illustrates a bit more delicate degeneracy. In all three pictures, the grey points end up collinear in the rubber band embedding. In the first graph, the reason is symmetry again. In the second, there is a lot of symmetry, but it does not explain why the three grey nodes are collinear in the equilibrium. (It is not hard to argue though that they are collinear: a good exercise!) In the third graph (which is not drawn in its equilibrium position, but before it) there are two nodes separating the grey nodes from the nailed nodes, and the grey nodes will end up on the segment connecting these two nodes. In the first two cases, changing the strength of the rubber bands will pull the grey nodes off the line; in the third, this does not happen.

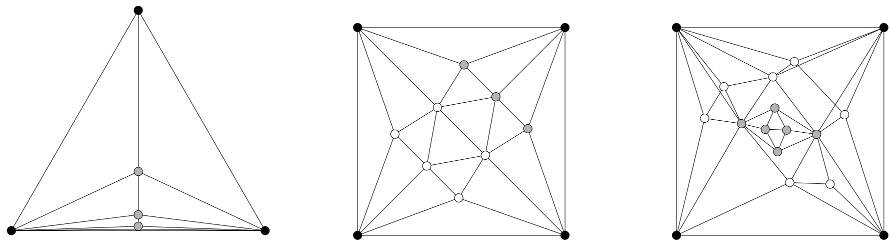

FIGURE 3.9. Three reasons why three nodes can end up collinear: symmetry, just accident, or a separating pair of nodes

3.3.2. Connectivity and degeneracy. Our goal is to prove that essential degeneracy in a rubber band embedding is always due to low connectivity. We start with the easy direction of the connection, relating connectivity and affine rank, formalizing the examples from the previous section.

LEMMA 3.5. *Let G be a graph and $S,T \subseteq V$. Then for every rubber band representation $\mathbf{u}$ of G with S nailed,* $\mathrm{rk}_{\mathrm{aff}}\big(\mathbf{u}(T)\big) \le \kappa(S,T)$.

PROOF. There is a subset $U \subseteq V$ with $|U| = \kappa(S,T)$ such that $V \setminus U$ contains no (S,T)-paths. Let W be the union of connected components of $G \setminus U$ containing a vertex from T. Then $\mathbf{u}$, restricted to W, gives a rubber band representation of $G[W]$ with boundary U. Clearly $\mathbf{u}(W) \subseteq \mathrm{conv}\big(\mathbf{u}(U)\big)$, and so

$$\mathrm{rk}_{\mathrm{aff}}\big(\mathbf{u}(T)\big) \le \mathrm{rk}_{\mathrm{aff}}\big(\mathbf{u}(W)\big) = \mathrm{rk}_{\mathrm{aff}}\big(\mathbf{u}(U)\big) \le |U| = \kappa(S,T). \qquad \square$$

The Lemma gives a lower bound on the connectivity between two sets S and T. The following theorem asserts that if we take the best convex representation, this lower bound is tight:

THEOREM 3.6. *Let G be a graph and $S,T \subseteq V$ with $\kappa(S,T) = d+1$. Then G has a rubber band representation in $\mathbb{R}^d$, with suitable rubber band strengths, with S nailed such that* $\mathrm{rk}_{\mathrm{aff}}\big(\mathbf{u}(T)\big) = d+1$.

This theorem has a couple of consequences about connectivity not between sets, but between a set and any node, and between any two nodes.

COROLLARY 3.7. *Let G be a graph, $d \ge 1$ and $S \subseteq V$. Then G has a rubber band representation in general position in $\mathbb{R}^d$ with S nailed if and only if no node of G can be separated from S by fewer than $d+1$ nodes.*

COROLLARY 3.8. *A graph G is k-connected if and only if for every $S \subseteq V$ with $|S| = k$, G has a rubber band representation in $\mathbb{R}^{k-1}$ in general position with S nailed.*

To prove Theorem 3.6, we choose generic edge weights.

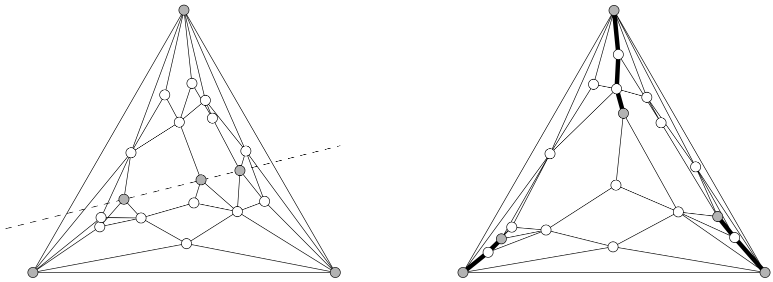

FIGURE 3.10. Three nodes accidentally on a line. Strengthening the edges along the three paths pulls them apart.

THEOREM 3.9. *Let G be a graph and $S,T \subseteq V$ with $\kappa(S,T) \geq d+1$. Choose algebraically independent edgeweights $c_{ij} > 0$. Map the nodes of S into $\mathbb{R}^d$ in general position. Then the rubber band extension of this map satisfies* $\mathrm{rk}_{\mathrm{aff}}\big(\mathbf{u}(T)\big) = d+1$.

If the edgeweights are chosen randomly, independently and uniformly from $[0,1]$, then the algebraic independence condition is satisfied with probability 1.

PROOF. The proof will consist of two steps: first, we show that there is *some* choice of the edgeweights for which the conclusion holds; then we use this to prove that the conclusion holds for *every* generic choice of edge-weights.

For the first step, we use that by Menger's Theorem B.2, there are $d+1$ disjoint paths P_i connecting a node $i \in S$ with a node $i' \in T$. The idea is to make the rubber bands on these paths very strong (while keeping the strengths of the other edges fixed). Then these paths will pull each node i' very close to i. Since the positions of the nodes $i \in S$ are affine independent, so will be the positions of the nodes i' (Figure 3.10).

To make this precise, let D be the diameter of the set $\{\mathbf{u}(S)\}$. Fix any $R > 0$, and define strengths c_{ij} of the rubber bands by

$$c_{ij} = \begin{cases} R, & \text{if } ij \text{ is an edge of one of the paths } P_r, \\ 1, & \text{otherwise.} \end{cases}$$

Let $\mathbf{u}$ be the rubber band extension of the given mapping of the nodes of S with these strengths.

Recall that f minimizes the potential $\mathcal{E}_c$ (defined by (3.5)) over all representations of G with the given nodes nailed. Let $\mathbf{v}$ be the representation with $\mathbf{v}_j = \mathbf{u}_i$ if $j \in P_i$ (in particular, $\mathbf{v}_i = \mathbf{u}_i$ if $i \in S$); for any node j not on any P_i, let $\mathbf{v}_j$ be any point in $\mathrm{conv}\{\mathbf{u}(S)\}$. In the representation $\mathbf{v}$ the edges with strength R have 0 length, and so

$$\mathcal{E}_S(\mathbf{u}) \leq \mathcal{E}_S(\mathbf{v}) \leq D^2 m.$$

On the other hand, for every edge uv on one of the paths P_i,

$$\mathcal{E}_S(\mathbf{u}) \geq R|\mathbf{u}_u - \mathbf{u}_v|^2,$$

and hence

$$|\mathbf{u}_u - \mathbf{u}_v| \leq \frac{D\sqrt{m}}{\sqrt{R}}.$$

So these edges can be made arbitrarily short if we choose R large enough, and hence every node i' will be arbitrarily close to the corresponding nailed node i. Since $\{\mathbf{u}_i : i \in S\}$ are affine independent, this implies that the nodes in $\{\mathbf{u}_{i'} : i \in T\} = \{\mathbf{u}_i : i \in S\}$ are affine independent for a large enough value of R.

This completes the first step. Now we argue that this holds for all possible choices of generic edge-weights. To prove this, we only need some general considerations. The embedding minimizing the energy is unique, and so it can be computed from the equations (3.6) (say, by Cramer's Rule). What is important from this is that the vectors $\mathbf{u}_i$ can be expressed as rational functions of the edgeweights. Furthermore, the value $\det\big((1+\mathbf{u}_{t_i}^\top \mathbf{u}_{t_j})_{i,j=0}^d\big)$ is a polynomial in the coordinates of the $\mathbf{u}_i$, and so it is also a rational function of the edgeweights. We know that this rational function is not identically 0; hence it follows that it is nonzero for every algebraically independent substitution. □

Remark 3.10. Instead of rubber band embeddings, it would suffice to assume that each free node is placed in the convex hull of its neighbors [**Cheriyan–Reif 1992**].

3.3.3. Rubber bands and connectivity testing. Rubber band representations yield a (randomized) graph connectivity algorithm with good running time [**Linial–Lovász–Wigderson 1988**]; however, a number of algorithmic ideas are needed to make it work, which we describe in their simplest form. We refer to the paper for a more thorough analysis.

Connectivity between given sets. We start with describing a test checking whether or not two given k-tuples S and T of nodes are connected by k node-disjoint paths. For this, we assign random weights c_{ij} to the edges, and map the nodes in S into $\mathbb{R}^{k-1}$ in general position. Then we compute the rubber band representation extending this map, and check whether the points representing T are in general position.

This procedure requires solving a system of linear equations, which is easy, but the system is quite large, and it depends on the choice of S and T. With a little work, we can save quite a lot.

Let G be a connected graph on $V = [n]$ and $S \subseteq V$, say $S = [k]$. Given a map $\mathbf{u} : S \to \mathbb{R}^{k-1}$, we can compute its rubber band extension by solving the system of linear equations

$$\sum_{j\in N(i)} c_{ij}(\mathbf{u}_i - \mathbf{u}_j) = 0 \qquad (i \in V \setminus S). \tag{3.16}$$

This system has $(n-k)(k-1)$ unknowns (the coordinates of nodes $i \in S$ are considered as constants) and the same number of equations, and we know that it has a unique solution, since this is where the gradient of a strictly convex function (which tends to ∞ at ∞) vanishes.

At the first sight, solving (3.16) takes inverting an $(n-k)(k-1) \times (n-k)(k-1)$ matrix, since there are $(n-k)(k-1)$ unknowns and $(n-k)(k-1)$ equations. However, we can immediately see that the coordinates can be computed independently, and since they satisfy the same equations except for the right hand side, it suffices to invert the matrix of the system once.

Below, we will face the task of computing several rubber band representations of the *same* graph, changing only the nailed set S. Can we make use of some of the computation done for one of these representations when computing the others?

The answer is yes. First, we do something which seems to make things worse: We create new "equilibrium" equations for the nailed nodes, introducing new variables for the forces that act on the nails. Let $\mathbf{f}_i = \sum_{j\in N(i)} c_{ij}(\mathbf{u}_j - \mathbf{u}_i)$ be the force acting on node i in the equilibrium position. Note that $Y\mathbf{e}_i = \mathbf{f}_i = 0$ for $i \notin S$. Let X and Y be the matrices of the vector labelings $\mathbf{u}$ and $\mathbf{f}$. Let L_c be the symmetric $V \times V$ matrix

$$(L_c)_{ij} = \begin{cases} -c_{ij} & \text{if } ij \in E, \\ \sum_{k\in N(i)} c_{ik}, & \text{if } i = j, \\ 0, & \text{otherwise} \end{cases}$$

(the weighted Laplacian of the graph). Then we can write (3.16) as $XL_c = Y$. To simplify matters a little, let us assume that the center of gravity of the node positions is 0, so that $XJ = 0$. We also have $YJ = XL_cJ = 0$. It is easy to check that

$$X = Y(L_c + J)^{-1} \tag{3.17}$$

satisfies these equations (for more about this, see the "inverting the Laplacian" trick in Section 4.3.2).

It is not clear that we are nearer our goal, since how do we know the matrix Y (i.e., the forces acting on the nails)? But the trick is this: we can prescribe these forces arbitrarily, as long as the conditions $\sum_{i\in S} \mathbf{f}_i = 0$ and $\mathbf{f}_j = 0$ for $j \notin S$ are satisfied. Then the rows of X will give some position for each node, for which $XL_c = Y$. So in particular the nodes in $V \setminus S$ will be in equilibrium, so if we nail the nodes of S, the rest will be in equilibrium. If Y has rank d, then so does X, which implies that the vectors representing S will span the space $\mathbb{R}^d$. Since their center of gravity is 0, it follows that they are not on one hyperplane. We can apply an affine transformation to move the points $\mathbf{u}_i$ ($i \in S$) to any other affine independent position if we wish, but this is irrelevant: what we want is to check whether the nodes of T are in general position, and this is not altered by any affine transformation.

To sum up, to check whether the graph is k-connected between S and T (where $S, T \subseteq V$, $|S| = |T| = k$), we select random weights c_{ij} for the edges, select a convenient matrix Y, compute the matrix $X = Y(L_c + J)^{-1}$, and check whether the rows with indices from T are affine independent. The matrix $L_c + J$ is positive definite, and it has to be inverted only once, independently of which sets are we testing for connectivity. The matrix Y has dimensions $(k-1) \times n$, but most of its elements are zero; a convenient choice for Y is

$$Y = \begin{pmatrix} 1 & 0 & \dots & 0 & -1 & 0\dots \\ 0 & 1 & \dots & 0 & -1 & 0\dots \\ \vdots & & & \ddots & & \\ 0 & 0 & \dots & 1 & -1 & 0\dots \end{pmatrix}.$$

and then the matrix multiplication $Y(L_c + J)^{-1}$ is trivial.

Connectivity between all pairs. If we want to apply the previous algorithm for connectivity testing, it seems that we have to apply it for all pairs of k-sets. Even though we can use the same edgeweights and we only need to invert $L_c + J$ once, we have to compute $X = Y(L_c + J)^{-1}$ for potentially exponentially many different sets S, and then we have to test whether the nodes of T are represented in general position for exponentially many sets T. The following lemma shows how to get around this.

LEMMA 3.11. *For every vertex $v \in V$ we select an arbitrary k-subset $S(v) \subseteq N(v)$. Then G is k-connected if and only if $S(u)$ and $S(v)$ are connected by k node-disjoint paths for every u and v.*

PROOF. The "only if" part follows from the well-known property of k-connected graphs that any two k-subsets are connected by k node-disjoint paths. The "if" part follows from the observation that if $S(u)$ and $S(v)$ are connected by k node-disjoint paths, then u and v are connected by k openly disjoint paths. □

Thus the subroutine in the first part needs be called only $O(n^2)$ times. In fact, we do not even have to check this for every pair (u, v), and further savings can be achieved through randomization, using Exercises 3.11 and 3.12.

Numerical issues. The computation of the rubber band representation requires solving a system of linear equations. We have seen (Figure 3.5) that for a graph with n nodes, the positions of the nodes can get exponentially close in a rubber band representation, which means that we might have to compute with exponentially small numbers (in n), which means that we have to compute with linearly many digits, which gives an extra factor of n in the running time. One way out is to solve the system in a finite field rather than in $\mathbb{R}$. Of course, this "modular" embedding has no physical or geometrical meaning any more, but the algebraic structure remains!

For the analysis, we need the Schwartz–Zippel Lemma [**Schwartz 1980**, **Zippel 1979**]:

LEMMA 3.12. *Given a polynomial over any field of degree k in m variables, the probability that it vanishes for a random substitution, where each variable is chosen uniformly from N possible values, is bounded by k/N.* □

Let G be a graph and let $S \subseteq V$, $|S| = k$, and $d = k - 1$. Let p be a prime and $c_e \in \mathbb{F}_p$ for $e \in E$. A *modular rubber band representation* of G (with respect to S, p and c) is defined as an assignment $i \mapsto \mathbf{u}_i \in \mathbb{F}_p^d$ satisfying

$$\sum_{j \in N(i)} c_{ij}(\mathbf{u}_i - \mathbf{u}_j) = 0 \qquad (\forall i \in V \setminus S). \tag{3.18}$$

This is formally the same equation as for real rubber bands, but we work over $\mathbb{F}_p$, so no notion of convexity can be used. In particular, we cannot be sure that the system has a solution. But things work if the prime is chosen at random.

LEMMA 3.13. *Let $N > 0$ be an integer. Choose uniformly a random prime $p < N$ and random weights $c_e \in \mathbb{F}_p$ ($e \in E$).*

(a) *With probability at least $1 - n^2/N$, there is a modular rubber band representation of G (with respect to S, p and c), such that the vectors $\mathbf{u}_i$ ($i \in S$) are affine independent. This representation is unique up to affine transformations of $\mathbb{F}_p^d$.*

(b) *Let $T \subseteq V$, $|T| = k$. Then with probability at least $1 - n^2/N$, the representation $\mathbf{u}_c$ in* (a) *satisfies $\mathrm{rk}_{\mathrm{aff}}(\{\mathbf{u}_i : i \in T\}) = \kappa(S, T)$.*

PROOF. The determinant of the system (3.18) is a polynomial of degree at most n^2 of the weights c_{ij}. The Schwartz–Zippel Lemma gives (a). The proof of (b) is similar. □

We sum up the complexity of this algorithm without going into fine details. In particular, we restrict this short analysis to the case when $k < n/2$. We have to fix an appropriate N; $N = n^5$ will do. Then we pick a random prime $p < N$. Since N has $O(\log n)$ digits, the selection of p and every arithmetic operation in $\mathbb{F}_p$ can be done in polylogarithmic time, and we are going to ignore these factors of $\log n$. We have to generate $O(n^2)$ random weights for the edges. We have to invert (over $\mathbb{F}_p$) the matrix $L_c + J$, this takes $O\big(M(n)\big)$ operations, where $M(n)$ is the cost of multiplying two $n \times n$ matrices (currently known to be $O(n^{2.3727})$ [**Vassilevska Williams 2012**]). Then we have to compute $Y(L_c + J)^{-1}$ for a polylogarithmic number of different matrices Y (using Exercise 3.12(b) below). For each Y, we have to check affine independence for one k-set in the neighborhood of every node, in $O\big(nM(k)\big)$ time. Up to polylogarithmic factors, this takes $O\big(M(n) + nM(k)\big)$ time.

Using much more involved numerical methods, solving linear equations involving the Laplacian and similar matrices can be done more efficiently. For details, we refer to [**Spielman 2011**] and [**Spielman–Teng 2014**].

EXERCISE 3.1. Prove that $\min_{\mathbf{u}} \mathcal{E}_c(\mathbf{u})$ (where the minimum is taken over all positions $\mathbf{u}$ with some nodes nailed) is a concave function of the rubber band strengths $c \in \mathbb{R}^E$.

EXERCISE 3.2. Let $\mathbf{x}$ be the equilibrium position of a rubber band framework with the nodes in S nailed, and let $\mathbf{y}$ be any other position of the nodes (but with the same nodes nailed at the same positions). Then

$$\mathcal{E}_c(\mathbf{y}) - \mathcal{E}_c(\mathbf{x}) = \mathcal{E}_c(\mathbf{y} - \mathbf{x}).$$

EXERCISE 3.3. Let G be a connected graph, $\emptyset \neq S \subseteq V$, and $\overline{\mathbf{u}} : S \to \mathbb{R}^d$. Extend $\overline{\mathbf{u}}$ to $\mathbf{u} : V \setminus S \to \mathbb{R}^d$ as follows: starting a random walk at j, let i be the (random) node where S is first hit, and let $\mathbf{u}_j$ denote the expectation of the vector $\overline{\mathbf{u}}_i$. Prove that $\mathbf{u}$ is the same as the rubber band extension of $\overline{\mathbf{u}}$.

EXERCISE 3.4. Let G be a connected graph, and let $\mathbf{u}$ be a vector-labeling of an induced subgraph H of G (in any dimension). If $(H, \mathbf{u})$ is section-connected, then its rubber-band extension to G is section-connected as well.

EXERCISE 3.5. Let $\mathbf{u}$ be a rubber band representation of a planar map G in the plane with the nodes of a country T nailed to a convex polygon. Define $\mathbf{F}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ for all edges in $E \setminus E(T)$. (a) If T is a triangle, then we can define $\mathbf{F}_{ij} \in \mathbb{R}^2$ for $ij \in E(T)$ so that $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$, $\mathbf{F}_{ij}$ is parallel to $\mathbf{u}_j - \mathbf{u}_i$, and $\sum_{i \in N(j)} \mathbf{F}_{ij} = 0$ for every node i. (b) Show by an example that (a) does not remain true if we drop the condition that T is a triangle.

EXERCISE 3.6. Prove that every Schlegel diagram with respect to a face F can be obtained as a rubber band representation of the skeleton with the vertices of F nailed (the strengths of the rubber bands must be chosen appropriately).

EXERCISE 3.7. Let G be a 3-connected simple planar graph with a triangular country $p = \overline{abc}$. Let q, r, s be the countries adjacent to p. Let G^* be the dual graph. Consider a rubber band representation $\mathbf{u} : V \to \mathbb{R}^2$ of G with a, b, c nailed down (both with unit rubber band strengths). Prove that the segments representing the edges can be translated so that they form a rubber band representation of $G^* - p$ with q, r, s nailed down (Figure 3.11).

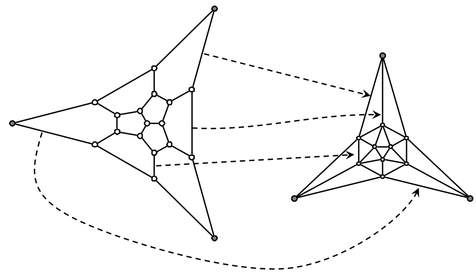

FIGURE 3.11. Rubber band representation of a dodecahedron with one node deleted, and of an icosahedron with the edges of a triangle deleted. Corresponding edges are parallel and have the same length.

EXERCISE 3.8. A *convex representation* of a graph G (in dimension d, with boundary $S \subseteq V$) is a mapping of $V \to \mathbb{R}^d$ such that every node in $V \setminus S$ is in the relative interior of the convex hull of its neighbors. (a) The rubber band representation extending any map from $S \subseteq V$ to $\mathbb{R}^d$ is convex with boundary S. (b) Not every convex representation is constructible this way.

EXERCISE 3.9. In a rubber band representation, increase the strength of an edge between two unnailed nodes (while leaving the other edges invariant). Prove that the length of this edge decreases.

EXERCISE 3.10. Prove that a 1-dimensional rubber band representation of a 2-connected graph, with boundary nodes s and t, nondegenerate in the sense that the nodes are all different, is an s-t-numbering (as defined in the Introduction). Show that instead of the 2-connectivity of G, it suffices to assume that deleting any node, the rest is either connected or has two components, one containing s and one containing t.

EXERCISE 3.11. Let G be any graph, and let H be a k-connected graph with $V(H) = V$. Then G is k-connected if and only if u and v are connected by k openly disjoint paths in G for every edge $uv \in E(H)$.

EXERCISE 3.12. Let G be any graph and $k < n$. (a) For t pairs of nodes chosen randomly and independently, we test whether they are connected by k openly disjoint paths. Prove that if G is not k-connected, then this test discovers this with probability at least $1 - \exp(-2(n-k)t/n^2)$. (b) For r nodes chosen randomly and independently, we test whether they are connected by k openly disjoint paths to every other node of the graph. Prove that if G is not k-connected, then this test discovers this with probability at least $1 - \big((k-1)/n\big)^r$.

EXERCISE 3.13. Let G be a graph, and let Z be a matrix obtained from its Laplacian L_G by replacing its nonzero entries by algebraically independent transcendentals. For $S, T \subseteq V$, $|S| = |T| = k$, let $Z_{S,T}$ denote the matrix obtained from Z by deleting the rows corresponding to S and the columns corresponding to T. Then $\det(Z_{S,T}) \neq 0$ if and only if $\kappa(S,T) = k$.