Efficient Networks and Enumerations on Forests: Master's thesis in Mathematics

Tamar Lando

April 13, 2009

1 Networks

1.1 Introduction

We study networks connecting n points in an area-n square. By network we mean a collection of straight line segments between points in the square. These points and line segments can respectively be thought of as cities and roads–the network then, is a collection of roads that allows us to travel from any city to any other city. Each such network N will have a total network length, $len(N)$. In building such a system of roads one might measure efficiency in terms of the total length of the network constructed, with longer networks less efficient. But one might also measure efficiency in terms of the shortness of the path between any two points. In particular, if the Euclidean distance between points i and j is $d(i, j)$, and the shortest path in the network between these points has length $l(i, j)$, the ratio $r(i, j) = \frac{l(i,j}{d(i,j)}$ is a measure of how efficiently one can travel between these two points. Generalizing over all pairs of points, we define the following R-statistic for the entire network: $R(N) = \max_{i \neq j} r(i, j)$, where the maximum is taken over all pairs of distinct points i and j. Intuitively, there is a tradeoff between these two features of the network: shorter networks will tend to have a poor R-statistic, and networks with a small R-statistic will tend to be longer.

In this paper we study the worst case tradeoff between network length and route-length efficiency. In particular, we consider only networks whose length grows linearly with n, and look at the worst case R-statistic for a given network, depending on where the points are positioned with respect to one another in the n-area square. More formally, for any $r > 1$ and any set of n points $\{z_1, \ldots, z_n\}$, let

$$L_r(z_1, \ldots, z_n) = \inf\{len(N) : R(N) \leq r\}$$

where the infimum is taken over all networks N that connect the points $z_1, \ldots, z_n$. (Thus $L_r(z_1, \ldots, z_n)$ is a measure for any *fixed* n-tuple of points.) We can now define a function $\theta(r)$ that describes the worst case tradeoff between route-length efficiency and normalized network length:

$$\theta(r) = \limsup_{n} \sup_{z^n} n^{-1} L_r(z_1, \ldots, z_n) \tag{1}$$

where the supremum is taken over all sets of points $\{z_1, \ldots, z_n\}$ in the square (and where we take the lim sup because we are interested in what happens asymptotically as n goes to infinity).

Although we cannot hope to compute $\theta(r)$ explicitly for a given value of r, we can try to bound it from above, by showing how to construct a network that achieves the desired R-statistic around any positioning of n points in the square. In the first section that follows, we analyze a group of networks based on the regular triangular, rectangular and hexagonal lattices, and show that they bound $\theta(r)$ for $r = 2$. In the following section we consider long networks with arbitrarily small R-statistics, and show that for a given r arbitrarily close to 1, we can bound $\theta(r)$ by a constant (this is not the case for $r = 1$, where $\theta(r)$ grows as a polynomial in n).

1.2 Triangular, Rectangular and Hexagonal Lattice Networks

In this section we provide a construction of a network based on the regular triangular lattice that gives upper bounds of x, y, z for $\theta(r_1), \theta(r_2), andtheta(r_3)$ respectively. We then show the construction can be extended to a rectangular and hexagonal lattice, and that these other constructions give bounds of ... By regular polygonal lattice we mean a lattice composed of equilateral polygons of equal size.

Construction: We construct the network as follows. Begin by superimposing a regular triangular lattice on the area-n square containing the n points (cities). Let s be the side length of a single edge of the triangular cells. Each of our n points inhabits exactly one cell of the triangular lattice (unless it lies on the lines of the lattice itself) and we connect these points to the lattice by drawing a line segment from the point perpendicular to each of the three edges of the cell. This produces a connected network, N.

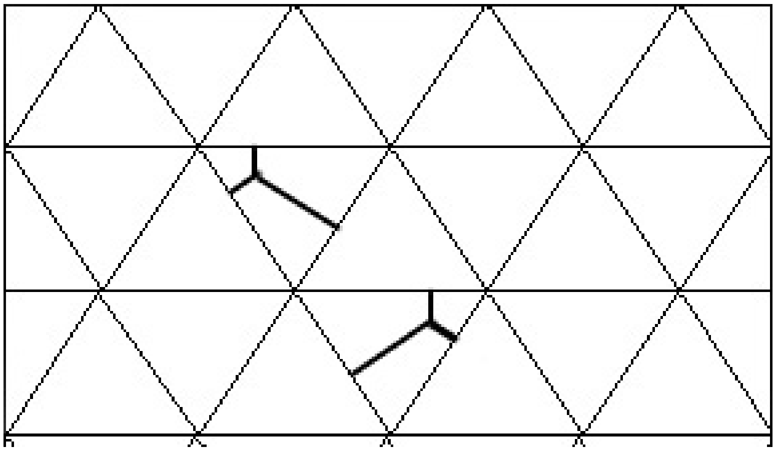


Figure 1. Network constructed around the regular triangular lattice with perpendicular access roads.

In what follows it will be useful to talk about length and ratio for a given path, not just for a pair of points as defined in the introduction. Thus for any path p, let $len(p)$ be the length of p, and let $r(p) = \frac{len(p)}{d(p)}$ where $d(p)$ is the distance between the endpoints of p. We will sometimes refer to r(p) as the "path ratio."

Claim 1. The R-statistic, R(N), for the network constructed above is 2.

To prove the claim, we need the following lemma:

Lemma 1. Let $x_1, \ldots, x_n$ be a sequence of n colinear points in the plane for some $n \geq 2$. Let p_i be a path connecting x_i to x_{i+1}, and let P be the path from x_1 to x_n produced by concatenating paths p_i for $1 \leq i \leq n-1$. Then

$$r(P) \leq \max_{1 \leq i \leq n-1} r(p_i)$$

(where r(P) and $r(p_i)$ are, respectively, the path ratios for paths P and p_i, $1 \leq i \leq n-1$).

Proof of Lemma 1. Let D be the Euclidean distance between i and j, and let d_i be the Euclidean distance between endpoints of path p_i. Then,

$$len(P) = \sum_i len(p_i) = \sum_i r_i \, d_i \leq \max_i r_i \sum_i d_i = \max_i r_i \, D$$

where the last equality holds because points $x_1, \ldots, x_n$ are colinear. Dividing both sides by D we get the desired result.

Proof of Claim 1. We let i, j be two arbitrary points (cities) from among

the n points positioned in the square, and we show that we can always find a path between the two points along the line segments of the network with path ratio at most 2. There are two cases, depending on whether or not i and j inhabit the same cell of the lattice. Let $\triangle i$ and $\triangle j$ be the triangles inhabited by i and j respectively.

Case (i): $\triangle i = \triangle j$.

Note that when we have two points in the same triangle, their access roads intersect at 120-degree angles. This means that in the worst case scenario (from the point of view of path ratio), points i and j are positioned at an equal distance away from the vertex of a 120-degree angle, and the path ratio $r(i,j) = \frac{l(i,j)}{d(i,j)}$ is $\frac{2}{\sqrt{3}}$, which is smaller than 2.

Case (ii): $\triangle i \neq \triangle j$.

Here we construct a path between i and j as follows. Begin by drawing a line segment from i to j (dashed in figure 2). Let A be the point where the line segment intersects $\triangle i$ and let B be the point where the line segment intersects $\triangle j$. In general, the line segment from i to j will traverse a series of triangles of the lattice, and we label the points where it intersects the edges of these triangles successively as $q_2, \ldots, q_{n-1}$, letting $A = q_1$ and $B = q_n$. Note now that q_1 and q_2 are on the first triangle traversed, q_2 and q_3 are on the second triangle traversed, etc. In general, we can travel from q_i to q_{i+1} by navigating around the edges of the ith triangle. In particular, this means navigating around a single 60 degree angle. We call the path from q_i to q_{i+1} constructed in this way p_{2i} $(1 \leq i \leq n-1)$. Concatenating p_{2i} for $1 \leq i \leq n-1$, we have a path p_2 from A to B along a series of colinear points $q_1, \ldots, q_n$.

To complete the path from i to j, we need to append to p_2 a path from i to A, and another path from B to j. But this is easy, because there are perpendicular access roads connecting i to the line segment where A lies, and connecting j to the line segment where B lies. Call the path from i to A defined in this way p_1 and call the path from B to j defined in this way p_3. Putting segments p_1, p_2 and p_3 together, we have a (non-optimal) path, P, from i to j. (In figure 2, segment p_2 is marked in red and segments p_1 and p_3 are marked in blue.)

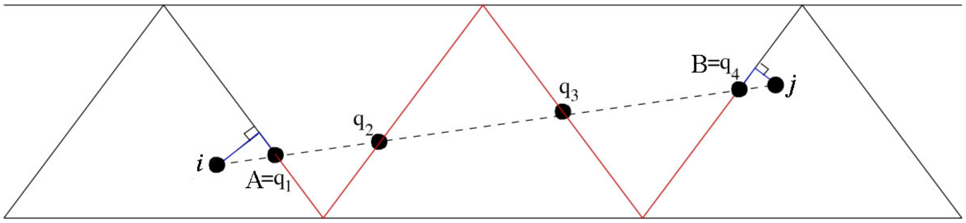


Figure 2. Path P from i to j.

By Lemma 1, since points $q_1, \ldots, q_n$ are colinear, we know that $r(p_2) \leq \max_{1 \leq i \leq n-1} r(p_{2i})$ Each of these paths navigates a single 60-degree angle, giving a worst-case path ratio of 2. So we have $r(p_2) \leq 2$. Again by Lemma 1, since i, A, B and j are colinear, we have:

$$r(P) \leq \max_{1 \leq i \leq 3} r(p_i)$$

But clearly $r(p_1), r(p_3) \leq \sqrt{2}$, since access roads are perpendicular to cell edges. Thus we have:

$$r(P) \leq \max\{2, \sqrt{2}\} = 2$$

To see that this bound is sharp, we can place two points on adjacent edges one of the triangular cells, positioned at an equal distance from the vertex where these edges meet. The path between the points along the edges of the triangle has path ratio 2. □

Claim 2. The total network length, $len(N)$ for the triangular lattice network constructed above, after choosing an optimal value for s (triangle side length), is $\sim 2\sqrt{3}n$. Thus, normalized network length is $\sim 2\sqrt{3}$.

Proof. We leave the details of the somewhat tedious calculation to the reader, and simply note that the access road length for each point is independent of where the point is positioned in the triangular cell and is (exactly) $\frac{s\sqrt{3}}{2}$. The lattice road length is (approximately) $\frac{6n}{s\sqrt{3}}$ (small variations occur depending on how one positions the lattice with respect to the area-n square). This gives total network length

$$f(s) = \frac{s\sqrt{3}n}{2} + \frac{6n}{s\sqrt{3}}$$

Optimizing over s we get $s = 2$ and $f(2) = 2\sqrt{3}n$.

Putting Claim 1 and 2 together, we now have an upper bound for $\theta(2)$ (where θ is the function defined in (1):

Claim 3. $\theta(2) \leq 2\sqrt{3}$

We note briefly that we can easily construct networks based on the rectangular and hexagonal lattices in a similar fashion, by superimposing the regular lattice on the square, then drawing access roads from each point perpendicular to all edges of the polygonal cell the point inhabits. Again, analyzing the R-statistic for each of these two networks involves checking Case (i), where two points inhabit the same lattice cell, and Case (ii), where two points inhabit distinct cells. In the rectangular network, for pairs of points subsumed under Case (i) there is always a path with ratio at most $\sqrt{2}$ (access roads intersect in right angles). In Case (ii) we construct a path similar to path P above, where r(P) is bounded by the maximum of $\sqrt{2}$ and the worst ratio for navigating around the edges of a single square, which turns out to be 2. Thus we have $R(N) \leq \max\{\sqrt{2}, 2\}$ and again this upper bound is sharp by positioning two points midway along opposite edges of a rectangular cell. Thus, the R-statistic for the rectangular network constructed in this way is equal to the R-statistic for the triangular network. However, total network length in the rectangular lattice is larger than total length in the triangular lattice (approximately 4n), so this does not improve the upper bound we gave in Claim 3. (Analysis of the hexagonal lattice is similar, but slightly complicated by the fact that access roads need not intersect with all edges of the hexagon, depending on where the point is positioned within the hexagonal cell.)

1.3 Long Networks with small R-statistic

We now turn to long networks with small R-statistics. In particular, we would like to know whether for r arbitrarily close to 1, we can find a network N such that $R(N) \leq r$ and $len(N)$ is linear in n. (Alternatively, if we look at the length of network *per city*, we are interested in networks where the normalized length is constant.)

Note that for $r = 1$ this is not possible since the only network, N, for which R(N)= 1 is the complete graph on n points. Assuming n is even, we can station $\frac{n}{2}$ points arbitrarily close to one corner of the square and $\frac{n}{2}$ arbitrarily close to the opposite corner. Then for any pair of points (i, j) where i and j are in opposite corners, we have $d(i, j) \sim \sqrt{2n}$. There are $(\frac{n}{2})^2$ such pairs (for n odd, there are $(\frac{n+1}{2})(\frac{n-1}{2})$ pairs), so

$$len(N) \sim \frac{\sqrt{2n}\, n^2}{4} \sim O(n^{\frac{5}{2}})$$

Given that we cannot construct a linear network for r=1, it makes sense to ask whether we can do so for r arbitrarily close to 1. In what follows we show, by construction, that this is indeed possible.

1.3.1 Building the network

We construct a network that contains only two angles, major and minor, and show that between any two “good” pair of points, a path can be found that uses only the major angle of the network. Making this angle wide enough guarantees a sufficiently small path ratio for all such pairs. By superimposing a finite number of rotated copies of this network, we get a network that works for *any* pair of points. The details are as follows.

Fix $r > 0$ and let θ_r be called the “major” angle of the network. Intuitively, we would like θ_r to be just wide enough so that a path, p, which navigates only around this angle has path ratio $r(p) \leq r$. The worst path ratio for a path navigating around a single angle is achieved when the starting and ending points–i and j–of the path are at equal distance from the vertex of the angle (see figure 3). Looking at figure 3, $\sin(\frac{\theta(r)}{2}) = \frac{\frac{d(i,j)}{2}}{\frac{l(i,j)}{2}} = \frac{1}{r(i,j)}$ So we set $\theta(r) = 2\sin^{-1}(\frac{1}{r})$. .

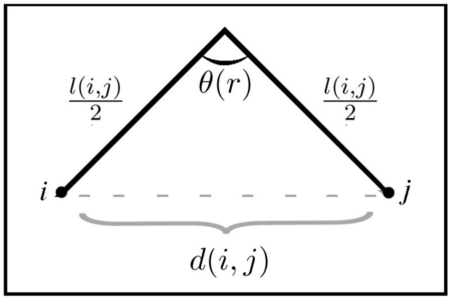


Figure 3. Defining $\theta(r)$.

Construction Our construction involves, again, a certain kind of lattice with access roads connecting points (cities) to roads of the lattice. Moving soutward along one of the vertical edges of our area-n square, we mark off points at a fixed distance y from one another. From each of these points we draw two lines, forming ψ_r and $-\psi_r$ degree angles with the horizontal, respectively. The horizontal lines–dashed in figure 4–are not themselves part of the lattice. (We may need to extend this procedure some way above and below the vertical edge of the square, to make sure we cover the whole square in roads of the lattice, as in the figure).

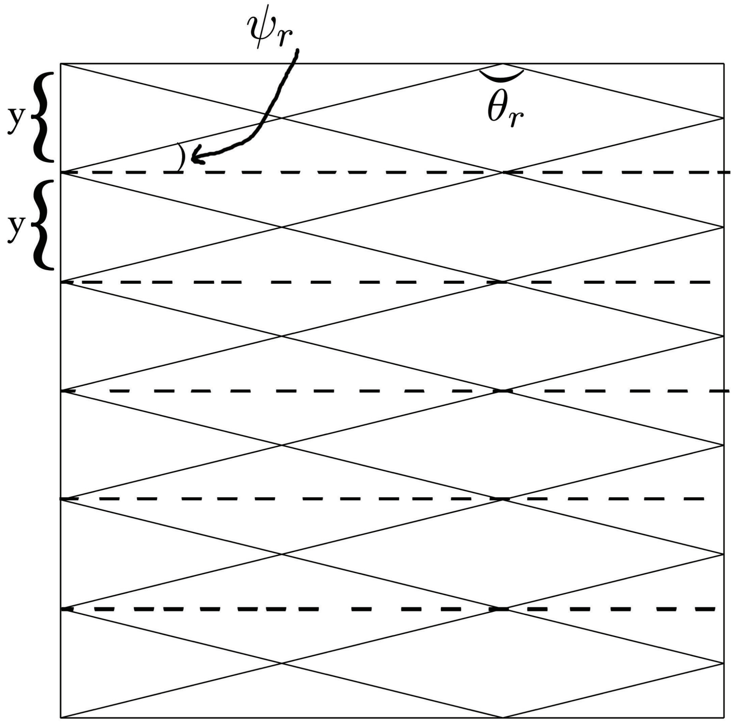


Figure 4. Lattice roads.

Note that, provided that y is small enough, the lines of our lattice intersect to form diamond-shaped cells, where the obtuse angle is θ_r and the acute angle is, say, $2\psi(r)$. We call $\psi(r)$ the minor angle of the network. Each of the n points (or cities) inhabits exactly one of these cells and we now draw two access roads through each point parallel to the edges of the cell (see figure 5).

Analysis of R-statistic. For any points in the square, i and j, draw a line between them and let $\phi_{i,j} \in [0, \pi)$ be the (positive) angle at which this line intersects the horizontal. We would like to show that the network we have constructed provides a sufficiently short path for all points i and j, such that $0 \leq \phi_{i,j} \leq \psi_r$. Consider in particular:

Provisional Claim 4: For any points i and j, such that $0 \leq \phi_{i,j} \leq \psi_r$, the ratio of shortest path distance to Euclidean distance along lines of the network is at most r.

Let NE roads of the lattice (excluding access roads) be called "runs" and NW roads be called "cuts." Also, let the region between two adjacent runs be called a bar (highlighted in pink in figure 5). Looking at figure 5, it turns out the claim is true in the following cases:

(i) the two points belong to the same cell (as e and f)

(ii) the two points are in the same bar and adjacent cells (as c and e)

(iii) the two points are in different bars (as c and d)

But is false in the case,

(iv) the two points belong to the same bar, but are separated by some number $k \geq 1$ of cells (as a and b)

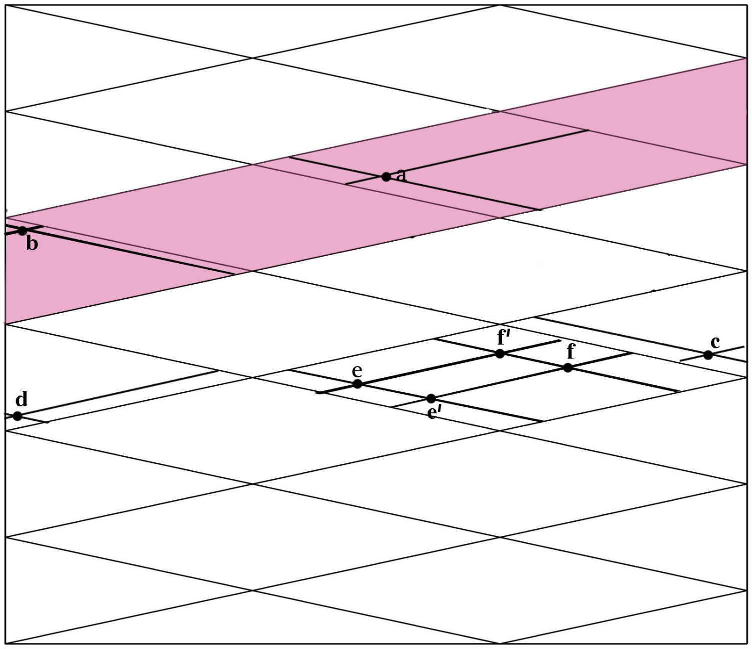


Figure 5. Points in the network.

The worst scenario in case (iv) is when the points are separated by one empty cell, and are each placed arbitrarily close to the midpoint of opposite edges (again, as a and b). To correct this, we need to add interior roads through the center of each cell and parallel to the NE edges (see figure 6). When we do this the worst situation in our new network is when the two points are a quarter of the way along opposite edges. Here the path ratio is $\frac{\frac{z}{4}+z+\frac{z}{4}}{z} = \frac{3}{2}$, where z is the length of the edge of a single cell. Continuing in this fashion, we can instead add *two* lines within each cell parallel to the NE edges and equally spaced apart. Here the worst situation is where the two points are $\frac{1}{6}$ of the way along opposite edges of the cell and the new path ratio is $\frac{\frac{z}{6}+z+\frac{z}{6}}{z} = \frac{4}{3}$ If we do this some finite number of times, we will eventually produce a short enough path between all such points. Indeed, if we partition each cell with n equally spaced lines our new worst situation path ratio is $\frac{\frac{z}{n+1}+z}{z} = \frac{n+2}{n+1}$. In order to ensure that our path ratio is smaller than r we pick n large enough so that $\frac{n+2}{n+1} \leq r$ or simply $n = \lceil \frac{2-r}{r-1} \rceil$.

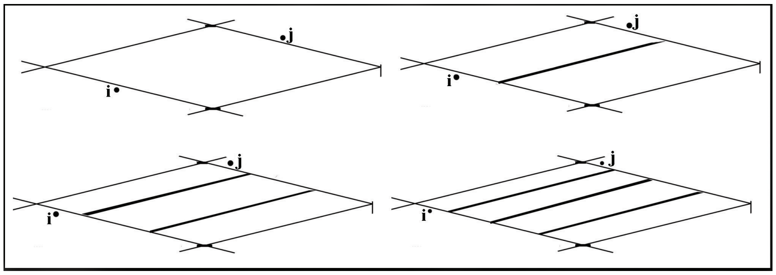


Figure 6. Partitioning each cell 0, 1, 2 and 3 times.

Claim 4: In the network obtained by adding $n = \lceil \frac{2-r}{r-1} \rceil$ interior roads to each cell, the path ratio $r(i,j)$ for all points i, j such that $0 \leq \phi_{i,j} \leq \theta_r$ is at most r.

Proof. We divide the proof into cases (i), (ii) and (iii) listed above (case (iv) is proved in the discussion of interior roads). In case (i) the two points, i, j belong to the same cell. The lattice roads of such points intersect in one of two ways, exhibited in Figure 5 by the pairs $\{e, f\}$ and $\{e\prime, f\prime\}$. The path along access roads from e to f is "short" while the path from $e\prime$ to $f\prime$ is "long". We need to show that all pairs of points i, j with $0 \leq \phi_{i,j} \leq \theta_r$ have access roads that intersect like those of e and f. But this follows from the fact that $0 \leq \phi_{i,j} \leq \theta_r$. Indeed, (WLOG) let i be west of j. The NE access road through i intersects the NW access road through j at point k, say. j must lie below the NE access road thru i, since $0 \leq \phi_{i,j} \leq \theta_r$. This means that the path from i to k to j navigates around a single θ_r-degree angle, so $r(i,j) \leq r$.

In case (ii) the two points i, j belong to different bars. Again, let i be west of j. Draw a line segment starting at i and ending at j, and label points where this line intersects successive bars as $y_1, \ldots, y_n$. We show there is a "short" path between y_i and y_{i+1} for $1 \leq i \leq n-1$. Concatenating such paths, and appending an initial segment from i to y_1 and a final segment from y_n to j, we get a "short" path from i to j.

Fix i, and note that y_i and y_{i+1} are points on the boundary of the same bar, but that y_i lies on the upper boundary (call this line b_i) and that y_{i+1}

lies on the lower boundary (call this line b_{i+1}). In general y_{i+1} is between two "cuts," and each of these cuts intersects both b_i and b_{i+1}. Thus we can travel from y_i along the line b_i to the point where it meets the first (westmost) cut, then follow the cut down to where it intersects line segment between i and j. This is the first segment of our path. The second continues along the cut until it intersects b_{i+1} and then travels along b_{i+1} to the point y_{i+1}. Both segments negotiate a single θ_r-degree angle, so by Lemma 1, the concatenated path from y_1 to y_{i+1} has path ratio at most r.

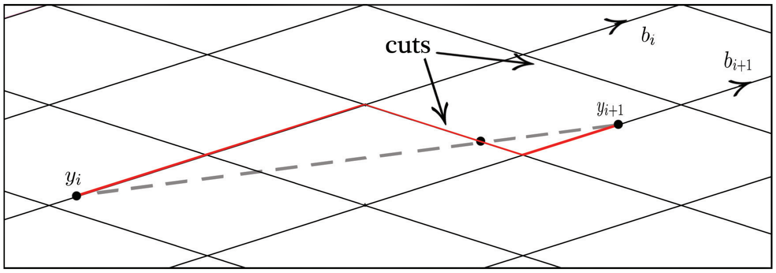


Figure 7. Path segment from y_i to y_{i+1}.

Concatenating segments from y_i to y_{i+1} for $(1 \leq i \leq n-1)$ we now have a path from y_1 to y_n. We need to show that there is a "short" path connecting i to y_1 and another "short" path connecting y_n to j. Indeed, take the access road thru i that intersects b_1 and follow b_1 to y_1. This path negotiates a single major angle of the network. We do the same for the final segment from y_n to j. It follows from Lemma 1 that the path ratio for the path we constructed from i to j is at most r.

In case (iii) i and j are in adjacent cells in the same bar. Again, let i be west of j. Then we can take access roads from i to the cut separating the two cells, travel along this cut, and take an access road to j. The path traverses two major angles of the network and the path ratio is at most r. □

To complete the construction, we need to ensure that pairs of points (i, j) for which $0 \leq \phi_{i,j} \leq \psi_r$ does *not* hold are captured by a "copy" of our original construction. Since ψ_r is constant, we can simply rotate the lattice (not including access roads) some finite number of times, $m = \lceil \frac{\pi}{\psi r} \rceil$, through the angle ψ_r and for each of these lattice copies draw the corresponding access roads. The kth copy captures all pairs of points (i,j) such that $(k-1)\psi_r \leq \phi_{i,j} \leq k\psi_r$. Moreover, since we have m copies of the original construction, the total network length is simply m times the original network length, which (provided the length of the original network is linear in n, as shown below) is clearly still linear in n.

1.4 Network Length

We show that the normalized length of the network we've constructed, when optimizing over y, is:

$$\frac{2\left\lceil \frac{\pi}{\psi_r} \right\rceil \sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}}{sin(\psi_r)}$$

In analyzing the total network length it is useful to think separately about the lattice, access roads and the number of rotations of the entire construction. Letting x be the length of a "full" line in our lattice (one that extends from one edge of the square to the opposite edge), we have $cos(\psi_r) = \frac{\sqrt{n}}{x}$ and therefore, $x = \frac{\sqrt{n}}{cos(\psi_r)}$. The number of lines in our construction is (not exact) $(p_r + 2)\frac{\sqrt{n}}{y}$. Multiplying the two gives **total lattice length** prior to taking copies of the construction (not exact),

$$\frac{(p_r + 2)\, n}{y\, cos(\psi_r)}$$

We saw above that $p_r = \lceil \frac{2-r}{r-1} \rceil$, which gives

$$\frac{(\lceil \frac{2-r}{r-1} \rceil + 2)\, n}{y\, cos(\psi_r)} \tag{2}$$

Letting z be the length of one edge of a single cell we have, $sin(\psi_r) = \frac{\frac{y}{2}}{z}$ or simply, $z = \frac{y}{2sin(\psi_r)}$. The length of access road per point is $2z$ which gives **total access road length**:

$$\frac{yn}{sin(\psi_r)} \tag{3}$$

(We now see why we required that y be constant. If y grows with n or even $\sqrt{n}$ the total access road length is not linear in n.) Thus the total length of a single copy of the network is (not exact)

$$\frac{(\lceil \frac{2-r}{r-1} \rceil + 2)\, n}{y\, cos(\psi_r)} + \frac{yn}{sin(\psi_r)}$$

Optimizing for y we set:

$$\frac{(\lceil \frac{2-r}{r-1} \rceil + 2)\, n}{y\, cos(\psi_r)} + \frac{yn}{sin(\psi_r)} = 0$$

which gives

$$y = \sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}$$

Therefore total optimized network length for a single copy is

$$\frac{(\lceil \frac{2-r}{r-1} \rceil + 2)\, n}{\sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}\, cos(\psi_r)} + \frac{\sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}\, n}{sin(\psi_r)}$$

and normalized network length for a single copy is

$$\frac{(\lceil \frac{2-r}{r-1} \rceil + 2)}{\sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}\, cos(\psi_r)} + \frac{\sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}}{sin(\psi_r)}$$

which is just

$$\frac{2\, \sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}}{sin(\psi_r)} \tag{4}$$

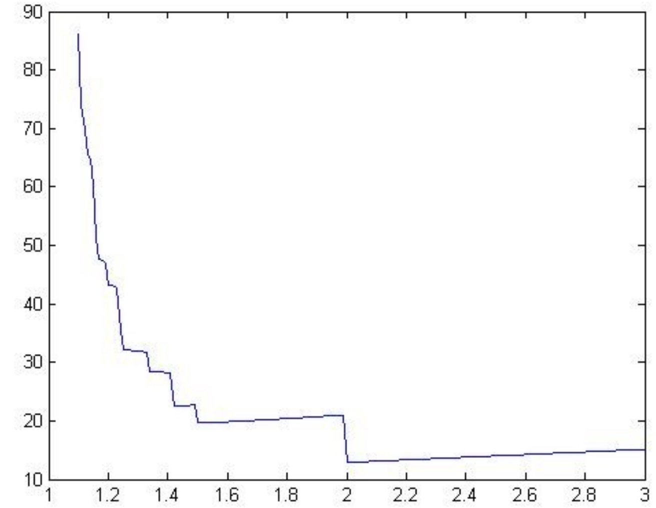
Finally, to "cover" all pairs of points we need $m = \lceil \frac{\pi}{\psi_r} \rceil$ copies of the original network. Thus the final (normalized) network length is:

$$\frac{2\, \lceil \frac{\pi}{\psi_r} \rceil\, \sqrt{tan(\psi_r)(\lceil \frac{2-r}{r-1} \rceil + 2)}}{sin(\psi_r)} \tag{5}$$

Noting that,

$$\psi_r = \frac{\pi - 2\sin^{-1}(\frac{1}{r})}{2}$$

we plot the normalized network length against r:



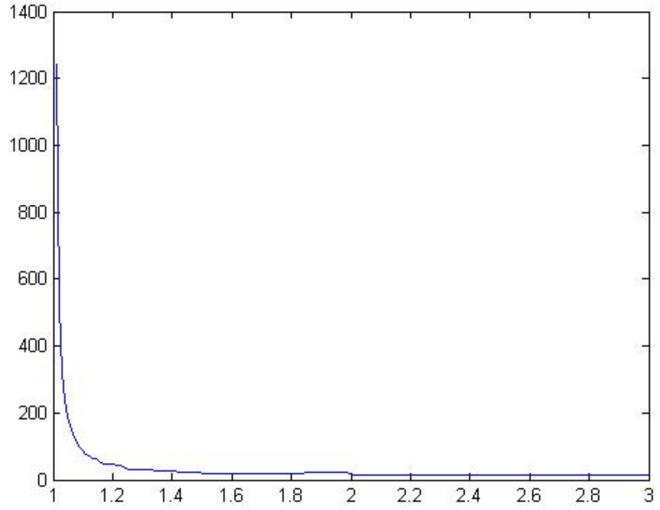


Figure 8. Graph of normalized network length for $1.1 \leq r \leq 3$ (top) and $1 < r \leq 3$ (bottom).

In particular, for values $r = 1.5, 2$ we get roughly 18.36 and 12.89 respectively.

2 Enumerations on forests and their Probabilistic Expressions

2.1 Introduction

In this part of the paper we present some enumerations of labeled trees and forests due to Jim Pitman, Bernard Harris, and Leo Katz, and explore their probabilistic expressions. **All of the results reviewed in this part of the paper have been proved elsewhere.** We begin, in Section 2, with Cayley's formula for the number of trees on n labeled vertices, and the usual derivation by Prufer coding. We then show, in Section 3, how the same formula can be derived as a special case of a more general enumeration of forests involving the notion of refining sequences, due to Pitman. In Section 4, we show how to construct a uniformly distributed random tree and random rooted forest with k components, and study the induced distribution on the set of partitions of [n]. These results are also due to Pitman. Finally, in Section 5, we show how the same distribution on partitions can be constructed from the uniform distribution on the set of mappings or functions from [n] to [n]. These results are due to Harris and Katz. The bulk of this literature review follows quite closely [**?**]

2.2 Cayley's formula and Prufer Coding

Definition 1 A *tree* over a set V is a not minimally connected graph with vertices labelled by the set V.

Definition 2 A *forest* over a set V is a graph whose components (maximally connected subgraphs) form a collection of trees labeled by the sets of some partition of V.

Thus a forest with one component is a tree.

We begin by introducing Cayley's formula, first stated in the late 1800's. In this section we briefly sketch two of the more recent proofs of the formula, the first due to Heinz Prufer and the second to Andre Joyal. Some of the details here are left out in the interest of spending more time on the material that follows.

Proposition 1 (Cayley's Formula) For all positive integers n, the number of trees over vertex set [n] is n^{n-2}.

Proof (Prufer Coding) We construct a bijection between the set of all trees over [n] and the set of sequences $(s_1, \ldots, s_{n-2})$ of length $n-2$, where

$s_i \in$ [n] for $i \leq n-2$. It follows that the cardinality of the set of trees is equal to the cardinality of the set of length $(n-2)$-sequences over [n], which is n^{n-2}.

Construction For a given tree, t, over [n], construct a sequence $S(t) = (s_1, \ldots, s_{n-2})$ as follows. In the first stage, pick the leaf with smallest labelled vertex. Delete the edge connecting it with it's (unique) neighbor and let s_1 be the vertex label of this neighbor. Iterate this procedure until only one edge is left, recording s_i at stage i. Since at each stage we delete one of the n vertices and we stop when only two vertices are left, the length of the sequence constructed is $(n-2)$.

A moment's thought shows that the following algorithm reconstructs a tree from its code. Indeed, the algorithm reconstructs the original tree edge by edge in the order of deletion.

Reconstruction Algorithm Let $S = (s_1, \ldots, s_{n-2})$ be a sequence on [n]. Let

$$L = [n] - \bigcup_{i=1}^{n-2} \{s_i\}$$

i.e. the set of all $x \in [n]$ that do not appear in the sequence, S. In the first stage of reconstruction, join the least element in L with s_1 by an edge, and remove this element from L. If s_1 does not appear again in the sequence, add s_1 to the set L. Repeat this procedure until the sequence S is exhausted. There are two vertices left in L (by the last stage, all vertices have appeared in L, and we've removed a total of $n-2$ vertices): connect these by an edge.

We need to show that the mapping we've defined from the set of trees over [n] to the set of all sequences of length $(n-2)$ over [n] is one-to-one and onto. To see that it is one-to-one, simply note that the reconstruction algorithm is well-defined. To see that the mapping is onto, we need to show that the reconstruction algorithm yields a tree for every sequence over [n] of length $n-2$. Note that at each stage the reconstruction algorithm yields a forest and that after the final stage, we have constructed $n-1$ edges. A forest with $n-1$ edges is connected, so the reconstruction algorithm yields a tree. $\square$

Proof (Joyal) We construct a bijection between the set of all doubly rooted trees and the set of mappings from [n] to [n]. Letting $F_{1,n}$ be the set of all trees over [n] (the reason for this notation will be apparent in subsequent sections), this gives the identity

$$n^n = n^2 \, \# F_{1,n}$$

from which Cayley's formula follows immediately.

Let f be a mapping from [n] to [n] and draw the "short diagram" of f–i.e. the graph on vertices [n] with an arrow from x to y iff $f(x) = y$ for all $x, y \in$ [n]. Let C be the set of vertices that are in directed cycles of the short diagram of f, and let N be the set of vertices that are not. (Note that C is non-empty, since we have a graph on n vertices with n edges.) Now let $(c_1, \ldots, c_p)$ be the ordering of elements in C such that $f(c_1) < f(c_2) < \cdots < f(c_p)$, and construct a doubly rooted graph over [n] as follows. For each $(1 \leq i \leq p - 1)$, draw an edge from c_i to c_{i+1}, and let c_1 and c_p be the start and end roots respectively (these may be identical). For each non-cyclic vertex $x \in N$, draw an edge from x to $f(x)$.

We need to make sure that the graph we constructed is a tree. Clearly the graph has $(n - 1)$ edges ($p - 1$ edges between vertices in C, and one edge for each of the $n - p$ vertices in N). Moreover, it's connected, since in the short diagram, each directed path starting at a vertex $x \in N$ must eventually reach a vertex in C. Finally, recall that any connected graph with $(n - 1)$ vertices is a tree.

We now show that the mapping we defined between mappings from [n] to [n] and doubly-rooted trees is both one to one and onto. Indeed, our mapping has an inverse defined on all such trees. Starting from a doubly rooted tree, we can recover the set C by taking all vertices in the (unique) path from the start root to the end root and choose f so that it sends the ith smallest vertex in C to the ith vertex on the path. The remaining vertices are elements of N and for all $x \in$ N we let $f(x)$ be the unique neighbor of x that is closer to the path (from start root to end root) than x is. □

2.3 Refining sequences

Definition 3 A *rooted forest* over a set V is a forest over V, where each component (or tree) has a distinguished vertex.

Thus a rooted forest with k component trees has k distinguished vertices, each belonging to different trees. We let $R_{k,n}$ be the set of all rooted forests over [n] with k components and let $F_{k,n}$ be the set of all unrooted forests over [n] with k components.

Definition 4 A *digraph* or directed graph, is a graph where each edge has a direction (i.e. is directed toward one of the two end-vertices).

Definition 5 If D, D* are two digraphs over a vertex set V, say D *contains* D* if each directed edge in D* is also in D.

In what follows, all trees and forests will have vertices labeled by the set [n], for some integer n.

We begin by introducing the notion of refining sequences of forests on [n], due to Pitman, which allows us to derive a series of enumerations on forests. A rooted forest can be identified with its digraph, where all edges in the digraph point away from the root. Let a length-k refining sequence of rooted forests over [n] be a sequence$(r_1, \ldots, r_k)$ of rooted forests where for $(1 \leq i \leq k-1)$, the digraph of r_i contains the digraph of r_{i+1} and where each forest is over the vertex set [n]. Thus e.g. a length-n refining sequence of rooted forests on [n] begins with a rooted tree and ends with the trivial forest over [n] (n vertices, no edges).

Proposition 2. For each rooted forest $r_k \in R_{k,n}$, the number $N(r_k)$ of rooted trees over [n] that contain r_k is n^{k-1}.

Proof. Fix $r_k \in R_{k,n}$. Let $N^*(r_k)$ be the number of length-k refining sequences $(r_1, ..., r_k)$ ending in the forest r_k, and note that for each tree $r_1 \in R_{k,n}$ containing r_k, the number of length-k refining sequences where the first term is r_1 and the last term is r_k is $(k-1)!$ (There are $k-1$ edges that must be deleted from r_1 and they can be removed in any order). Thus we have:

$$N^*(r_k) = N(r_k)(k-1)!$$

To count $N^*(r_k)$, consider choosing a refining sequence in reverse order–that is, building a tree from the forest $r_k \in R_{k,n}$. At each stage we add a single edge between unconnected vertices in such a way that the resulting graph is still a forest. We do this by choosing one of the n vertices in the forest and connecting it with the root of a different tree. (We can only join the vertex to a *root* since otherwise the edges of our resulting digraph will not all point away from the roots.) Thus, at the first stage we have $n(k-1)$ edges to choose from, at the second stage we have $n(k-2)$ choices and so on. We must add $k-1$ edges to get a tree, so there are $N^*(r_k) = n^{k-1}(k-1)!$ ways of doing this. From (1) we have:

$$N(r_k) = n^{k-1}$$

□

Corollary 1. Cayley's formula. Note that there is only one forest over [n] with n components (the trivial forest) and that every tree over [n] contains this forest. Letting $k = n$ in the previous proposition, we see that the number of rooted trees over [n] is simply $N(r_n) = n^{n-1}$. Dividing by n we get Cayley's formula for the number of (unrooted) trees over [n].

The method of refining sequences also allows us to count the number $\#R_{k,n}$ of rooted forests over [n] with k components. Indeed, we count the total number of length-n refining sequences and divide by the number of refining sequences containing a particular $r_k \in R_{k,n}$ (which depends only on k). The total number of length-n refining sequences $(r_1, ..., r_n)$ is just the number of ways to build a tree, edge by edge, from the trivial forest over n vertices (since each such refining sequence must end in the trivial forest). As above, in the first stage of the construction, we can add any of the n(n-1) edges. In the second stage, we can add any of the $n(n-2)$ edges, etc. There are $n-1$ stages in the construction (one for each edge added) so the total number of sequences is $n^{n-1}(n-1)!$. To count the number of such sequences that contain some particular rooted forest $r_k \in R_{k,n}$, we count the number of "ways up" (from the rooted forest to a tree) and multiply by the number of "ways down" (from the rooted forest to the trivial forest). As we saw above, the first of these numbers is just $n^{k-1}(k-1)!$. The number of ways down is $(n-k)!$, since r_k has $(n-1)-(k-1)$ edges, which we can delete in any order. Finally, dividing these two expressions, we see that

$$\#R_{k,n} = \frac{n^{n-1}(n-1)!}{n^{k-1}(k-1)!(n-k)!} \tag{6}$$

We turn now to unrooted forests, and derive the unrooted analog of Proposition 1. For a given unrooted forest $f_k \in F_{k,n}$ with k components, let $n_1, ..., n_k$ be the sizes of component trees. Unlike Proposition 1, the number of trees containing a particular (unrooted) forest depends in general on the sizes $n_1, ..., n_k$.

Proposition 3. For each (unrooted) forest $f_k \in F_{k,n}$, with component sizes $n_1, ..., n_k$, the number $N(f_k)$ of (unrooted) trees over [n] that contain f_k is $(\prod_{i=1}^{k} n_i) n^{k-2}$.

Proof. We count the number of *rooted* trees that contain f_k with directions of edges ignored (i.e. a rooted tree contains an unrooted forest if there is some way of assigning directions to edges such that containment follows). Each such tree is obtained uniquely by first choosing roots for all k component trees of the unrooted forest, then picking from one of the rooted trees over [n] that contain this rooted forest. There are $(\prod_{i=1}^{k} n_i)$ ways of choosing roots, and (by Proposition 2) n^{k-1} trees for each rooted forest. Multiplying the two we get

$$nN(f_k) = (\prod_{i=1}^{k} n_i) n^{k-1} \tag{7}$$

and dividing by n gives the desired enumeration.

Note that we cannot use the same method as before (dividing the total number of refining sequences by the number of sequences containing a particular forest) to derive an enumeration of $F_{k,n}$, the set of unrooted forests with k components, because the number of refining sequences containing a particular k-component forest depends in general on the sizes of component trees.

We can, however, count the number $N^*_{n_1,...,n_k}(f_k)$ of refining sequences (on unrooted forests) that contain a given forest $f_k \in F_k$ with component sizes $n_1, ..., n_k$:

$$N^*_{n_1,...,n_k}(f_k) = (\prod_{i=1}^{k} n_i) n^{k-2}(k-1)!(n-k)! \tag{8}$$

(Count the number of trees containing f_k and multiply by the number of sequences in which edges can be deleted, which is just $(k-1)!(n-k)!$)

We can also count the total number of refining sequences on unrooted forests over [n], by counting the number of (unrooted) trees and multiplying by the number of sequences in which all $n-1$ edges can be deleted. Using Cayley's formula, this is just:

$$n^{n-2}(n-1)! \tag{9}$$

2.4 A probability distribution on labelled forests and partitions of [n]

2.4.1 Distributions on the set of rooted and unrooted forests

Using the enumerations from the previous section, we now study the probability distribution on $R_{k,n}$ constructed by choosing uniformly from among all refining sequences of rooted forests over [n]. In particular, choosing uniformly from all such sequences and selecting the kth component, we get a random k-component forest R_k, which has uniform distribution over the set

$R_{k,n}$. The following theorem states that we can achieve the same distribution by choosing uniformly at random from among all trees over [n] and deleting (uniformly) $k-1$ edges, or by starting from the trivial forest over [n] and building a k-component forest by adding edges according to the given coalescent condition. More formally:

Proposition 4. The following three descriptions of the distribution of a random refining sequence $(R_1, ..., R_n)$ of rooted forests over [n] are equivalent and yield the uniform distribution on $R_{k,n}$ for $(1 \leq k \leq n)$:

(i)Choose R_1 uniformly from the set of all rooted trees over [n] and let $(E_1, \ldots, E_{n-1})$ be a uniformly chosen permutation of the $n-1$ edges in R_1. For $(1 \leq k \leq n)$, R_k is the rooted forest with k components obtained by deleting the first $n-1$ edges in the permutation $(E_1, \ldots, E_{n-1})$ from R_1.

(ii)R_n is the trivial rooted forest over [n], and for $(2 \leq k \leq n)$, R_{k-1} is obtained from R_k by adding an edge chosen uniformly at random from among the $n(k-1)$ edges that, when added, yields a forest with $k-1$ components.

(iii)$(R_1, ..., R_n)$ is a refining sequence of forests over [n] chosen uniformly from the set of all $(n-1)!(n)^{n-1}$ such sequences.

Proof.

Fix $r_k \in R_{k,n}$. We show that

$$Prob(R_k = r_k) = \frac{1}{\#R_{k,n}} = \frac{1}{\binom{n-1}{k-1}n^{n-k}}$$

for each of the constructions given in (i), (ii) and (iii).

(i) For each $r_k \in R_{k,n}$, $Prob(R_k = r_k)$ is just the probability of picking a tree that contains r_k times the probability of picking a "good" permutation of edges. But note that the number of such good permutations is $(k-1)!(n-k)!$ since we require only that the first $(k-1)$ edges in the sequence be those that are absent in the k-forest, r_k. We have:

$$\begin{aligned} Prob(R_k = r_k) =& \frac{N(r_k)}{\#\text{trees over [n]}} \frac{(k-1)!(n-k)!}{(n-1)!} \\ =& \frac{n^{k-1}}{n^{n-1}} \frac{(k-1)!(n-k)!}{(n-1)!} \\ =& \frac{1}{\binom{n-1}{k-1}n^{n-k}} \end{aligned}$$

(ii) We can think of the coalescent process described in (ii) as the process of constructing a refining sequence $(R_1, \ldots, R_n)$ in reverse order (i.e. starting from the trivial forest), where according to (ii) the probability $P(R_k = r_k \mid R_{k+1} = r_{k+1})$ for any r_k containing r_{k+1} is just $\frac{1}{n(k-1)}$ for $(1 \leq k \leq n-1)$. This means that the probability of constructing a given sequence $(r_1, \ldots, r_n)$ from the original (trivial) forest is $\frac{1}{n(n-1)\, n(n-2) \ldots n(1)} = \frac{1}{n^{n-1}\,(n-1)!}$–i.e. the probability is the same for each sequence. We count the number of such sequences containing r_k and multiply by this probability to get

$$P(R_k = r_k) = \frac{n^{k-1}(k-1)!(n-k)!}{n^{n-1}(n-1)!} = \frac{1}{\#R_{k,n}}$$

(iii) As in (ii), for any $r_k \in R_{k,n}$, the number of refining sequences $(r_1, \ldots, r_n)$ containing r_k depends only on k (and not on any other features of the particular forest r_k), so we get a uniform distribution over the set $R_{k,n}$.□

We now state the analog of Proposition 4 for unrooted forests. Here, however, the uniform distribution on refining sequences of unrooted forests does not give the uniform distribution on the set $F_{k,n}$ for $2 \leq k \leq n-1$. This is because the number of such refining sequences that a particular unrooted forest belongs to depends on the sizes of its tree components, and in general, forests where component sizes are more evenly distributed occur in more refining sequences than forests where tree sizes are unevenly distributed. (Since there is only one component size for unrooted trees, this distribution *is* uniform over the set of unrooted tres, $F_{1,n}$.)

Proposition 5. The following three statements describe the same distribution on the set of refining sequences $(F_1, ..., F_n)$ and in particular imply that for each $f_k \in F_k$ with tree component sizes $n_1, ..., n_k$,

$$P(F_k = f_k) = \frac{(\prod_{i=1}^{k} n_i)}{n^{n-k}\binom{n-1}{k-1}} \qquad (10)$$

(i*/*) Choose F_1 uniformly from the set of all n^{n-1} trees in $F_{1,n}$ and choose uniformly a permutation $(E_1, ..., E_{n-1})$ of the edges in F_1. F_k is the forest produced by deleting the first $k-1$ edges of the permutation from F_1, for $1 \leq k \leq n$.

(ii*/*) F_n is the trivial forest over [n] and for $2 \leq k \leq n$, given $F_n, F_{n-1}, ...F_k$, the forest F_{k-1} is derived as follows. If tree T_i has size n_i and tree T_j has size

n_j in F_k and $i < j$, then choose the pair (i, j) with probability $\frac{n_i+n_j}{n(k-1)}$. Now, for any vertex $a \in T_i$ and $b \in T_j$ choose a with probability $\frac{1}{n_i}$ and choose b (independently) with probabiity $frac1n_j$. F_{k-1} is the forest derived from F_k by adding an edge between a and b.

(iii$\prime$) $(F_1, ..., F_n)$ is a refining sequence of unrooted forests over [n] chosen uniformly from the set of all such sequences.

Proof. In (iii$\prime$), $P(F_k = f_k)$ is just the number of refining sequences containing f_k divided by the total number of refining sequences, which, from (2) and (3) is just the fraction

$$\frac{(\prod_{i=1}^{k} n_i)n^{k-2}(k-1)!(n-k)!}{n^{n-2}(n-1)!} \tag{11}$$

which is clearly equivalent to (5).

In (i$\prime$) $P(F_k = f_k)$ is the probability of selecting a tree that contains f_k multiplied by the conditional probability of selecting a "good" edge sequence given we have chosen such a tree. There are $(\prod_{i=1}^{k} n_i)n^{k-2}$ trees that contain f_k and n^{n-2} trees over [n], so the first of these probabilities is

$$\frac{(\prod_{i=1}^{k} n_i)n^{k-2}}{n^{n-2}}$$

Given that we have picked a tree containing f_k there are $(k-1)!(n-k)!$ good edge sequences and $(n-1)!$ total edge sequences. So the second probability is

$$\frac{(k-1)!(n-k)!}{(n-1)!}$$

Multiplying the two, we get the same probability as (5).

Note now that the construction of f_k given in (i$\prime$) and (iii$\prime$) is just the "unrooting" of the construction given in (i) and (iii) of Proposition 4 respectively. So to show that (ii$\prime$) is equivalent to (i$\prime$) and (iii$\prime$), we just need to show that (ii$\prime$) is the unrooting of the construction given in (ii). To do this, we show that the the conditional probability, $P(F_{k-1} = f_{k-1} \mid F_k = f_k)$ that we get from (ii$\prime$) is just the probability

$$\text{Prob (unrooting of } R_{k-1} = f_{k-1} \mid \text{unrooting of } R_k = f_k)$$

Let f_k be a forest with k component trees where vertex a is in tree T_i and vertex b is in tree T_j for $i < j$, and let f_{k-1} be the forest derived from f_k by joining a to b.There are two ways this could happen in the sequence of rooted forests.

Case 1: a is the root of T_i in R_k, and R_{k-1} adds the edge from b to a

Case 2: b is the root of T_j in R_k , and R_{k-1} adds the edge from a to b

Given $F_k = f_k$ the probability that a is a root of T_i in R_k is just $\frac{1}{n_i}$ and the probability of joining a and b given that a is a root of T_i is $\frac{1}{n(k-1)}$. So Case 1 happens with probability $\frac{1}{n_i}\frac{1}{n(k-1)}$. Likewise, Case 2 happens with probability $\frac{1}{n_j}\frac{1}{n(k-1)}$. So Prob (unrooting of $R_{k-1} = f_{k-1}$ | unrooting of $R_k = f_k$) is

$$\frac{1}{n_i}\frac{1}{n(k-1)} + \frac{1}{n_j}\frac{1}{n(k-1)} = \frac{1}{n_i n_j}\frac{n_i + n_j}{n(k-1)}$$

which is just the probability of joining a and b given in (ii$\prime$). □

2.4.2 Partitions of [n]

There is a very natural mapping from the set of unrooted forests over [n] to the set of partitions of [n], which we get by letting the vertices of each component of the forest define a single equivalence class. In this section, we study the distribution on the set of partitions on [n] induced by the distribution on forests described in Proposition 5. First, some definitions.

Definition 6 A *k-partition* of the set [n] is a set of sets $\{A_1, ..., A_k\}$ where $\cup_{i=1}^{k} A_i = [n]$ and the A_i's are disjoint and non-empty.

Definition 7 For each component tree T_i in $f_k \in F_{k,n}$ $(1 \leq i \leq k)$, let A_i be the set of vertices in T_i. Then the partition $\Pi_k = \{A_1, ..., A_k\}$ is the partition of [n] *induced by* f_k.

Proposition 6 Let Π_k be the random k-partition of [n] induced by the distribution on the random forest F_k described in Proposition 5. For any partition $\{A_1, ..., A_n\}$ with $\#A_i = n_i$ for $(1 \leq i \leq k)$

$$P(\Pi_k = \{A_1, ..., A_k\}) = \frac{\prod_{i=1}^{k} n_i^{n_i - 1}}{n^{n-k}\binom{n-1}{k-1}} \tag{12}$$

Proof. Note that each forest that yields the partition $\{A_1, ..., A_k\}$ has component sizes $\{n_1, ..., n_k\}$. Thus by Proposition 5, $P(F_k = f_k)$ for each such forest f_k is the same. So to prove Proposition 6, we simply multiply the number of forests that yield the partition $\{A_1, ..., A_k\}$ by $P(F_k = f_k)$ for each such forest. The number of forests that induce the partition $\{A_1, ..., A_k\}$ is just the number of forests over $[n_1]$ vertices times the number of forests over $[n_2]$, etc, which is, from Cayley's formula, $\displaystyle\prod_{i=1}^{k} n_i^{n_i-2}$. Since each such forest, f_k, has component sizes $(n_1, ..., n_k)$ we have from Proposition 5 that $P(F_k = f_k) = \frac{(\prod_{i=1}^{k} n_i)}{n^{n-k}\binom{n-1}{k-1}}$. Multiplying the two together we get (7). □

If we now let the number of groupings, k, in our partition be random, we can study the corresponding distribution on the set $\bigcup_{k=1}^{n} P_{n,k}$ of all partitions of the set [n]. In particular, let Π_K be the random partition given by first fixing K according to some probability distribution and then letting the conditional probability $P(\Pi_K = \{A_1, ..., A_k\} \mid K = k)$ be the distribution on $F_{k,n}$ described in Proposition 5. We have:

$$P(\Pi_K = \{A_1, ..., A_k\}) = P(K = k)\frac{\displaystyle\prod_{i=1}^{k} n_i^{n_i-1}}{n^{n-k}\binom{n-1}{k-1}} \tag{13}$$

In the next section we study a very different model for constructing a random forest and determine for what probability $P(K = k)$ in (8) this new distribution on random forests gives the same induced distribution on partitions of [n].

2.5 Probability distribution revisited: mappings from n to n

In what follows we let J be the set of all mappings $T : [n] \to [n]$.

Definition 8 A point $x \in [n]$ is *cyclical* if $T^k(x) = x$ for some positive integer k.

We can identify a mapping $T \in J$ with its short diagram (see Section 2) by letting each element in [n] be a vertex and drawing directed edge $i \to j$ iff $T(i) = j$. Note that in general, each component of the short diagram of a mapping has a single cycle and trees that connect to (some) vertices of the cycle. The edges of a tree in the digraph always point toward the cycle.

Given a mapping $T : [n] \to [n]$, we construct subsets of the set [n] as follows:

Let $M_0(T)$ be the set of all cyclical points in T.

Let $M_1(T)$ be the set of points $x \in [n]$ such that $T(x)$ is cyclical, but x is not.

Let $M_2(T)$ be the set of points $x \in [n]$ such that $T^2(x)$ is cyclical, but $T(x)$ is not.

etc.

We go on in this way until the points of [n] are exhausted. Let p be the number of such non-empty sets. We have in general:

$$M_i(T) = \{x \in [n] \mid T^i(x) \text{ is cyclical but } T^{i-1}(x) \text{ is not}\}$$

Clearly $M_i(T)(1 \leq i \leq p)$ is a partition of the set [n]. We let $m_0, ..., m_p$ be the cardinalities of the sets $M_0(T), ..., M_p$ respectively.

Proposition 7. Let $S_{(m_0,\ldots,m_p)}$ be the set of all mappings for which $\#M_i = m_i$ for $(0 \leq i \leq p)$. Then we have:

$$\#S_{(m_0,\ldots,m_p)} = n!\frac{{m_0}^{m_1}{m_1}^{m_2}\ldots{m_{p-1}}^{m_p}}{m_1!m_2!\ldots m_p!}$$

Proof. We count the number of ways of paritioning [n] into sets $M_0 \ldots M_p$, which is just

$$\binom{n}{m_0\, m_1 \ldots m_p}$$

and multiply by the number of ways of connecting elements in $M_0(T)$ to each other, times the number of ways of connecting elements in $M_1(T)$ to elements in $M_0(T)$, times the number of ways of connecting elements in $M_2(T)$ to elements in $M_1(T)$, etc.

The number of ways of connecting elements in $M_0(T)$ to each other is just the number of ways of dividing m_0 elements into an arbitrary number of groups and forming a cycle from the elements of each group. Note that this is equivalent to the problem of counting the number of ways to divide m_0 people into an arbitrary number of "cliques" and seat each clique around a circular table. It can be shown using exponential generating functions that the solution is $m_0!$. Clearly the number of ways of connecting vertices in M_i to vertices in $M_{i-1}(1 \leq i \leq p)$ is just $m_{i-1}^{m_i}$, since we can connect any number of vertices in $M_i(T)$ to the same vertex in $M_{i-1}(T)$. Altogether we have

$$\#S_{(m_0,\ldots,m_p)} = \binom{n}{m_0\, m_1 \ldots m_p} m_0!\, {m_0}^{m_1}{m_1}^{m_2}\ldots{m_{p-1}}^{m_p}$$

which is equivalent to the expression above. □

Corollary. The number of mappings $T : [n] \to [n]$ with j cyclical points is

$$\sum_{p=1}^{n} \sum n! \frac{j^{m_1} {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}}{m_1! m_2! \ldots m_p!} \tag{14}$$

where the inner sum is taken over all sequences $(m_1, \ldots, m_p)$ such that $\sum_{i=1}^{p} m_i = n - j$.

Proposition 8. Let T be a random mapping from [n] to [n] with uniform distribution over the set of all n^n such mappings. Let K be the (random) number of cyclical points in T. Then,

$$Prob(K = j) = \frac{(n-1)!\, j}{(n-j)!\, n^j} \quad j = 1, \ldots, n \tag{15}$$

Lemma.

$$\sum_{p=1}^{n} \sum \frac{1}{j} \frac{j^{m_1} {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}}{m_1! m_2! \ldots m_p!} = \frac{n^{n-j-1}}{(n-j)!} \tag{16}$$

where the inner sum is again taken over all sequences $(m_1, \ldots, m_p)$ such that $\sum_{i=1}^{p} m_i = n - j$.

Proof of Lemma (Algebra). To simplify, we let $M = n - j$. We need to show that

$$\sum_{m=1}^{n} \sum \frac{1}{j} \frac{j^{m_1} {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}}{m_1! m_2! \ldots m_p!} = \frac{(M+j)^{M-1}}{M!}$$

Expanding the binomial on the RHS, we get

$$\begin{aligned} \frac{(M+j)^{M-1}}{M!} &= \sum_{m_1=1}^{M} \binom{M-1}{m_1-1} \frac{1}{M!} j^{m_1-1} M^{M-m_1} \\ &= \sum_{m_1=1}^{M} \frac{j^{m_1-1}}{(m_1-1)!} \frac{(M-1)!}{M!} \frac{M^{M-m_1}}{(M-m_1)!} \\ &= \sum_{m_1=1}^{M} \frac{j^{m_1-1}}{(m_1-1)!} \frac{M^{M-m_1-1}}{(M-m_1)!} \end{aligned} \tag{17}$$

Now letting $M - m_1 = M_1$ we have:

$$= \sum_{m_1=1}^{M} \frac{j^{m_1-1}}{(m_1-1)!} \frac{M^{M_1-1}}{(M_1)!}$$

Note that the second factor above is of the same form as (11) and can be expanded similarly with respect to m_2. Iterating this (an arbitrary number m times) we get:

$$\sum_{m_1=1}^{M} \frac{j^{m_1-1}}{(m_1-1)!} \sum_{m_2=1}^{M_1} \frac{{m_1}^{m_2-1}}{(m_2-1)!} \cdots \sum_{m_p=1}^{M_{p-1}} \frac{m_{p-1}^{m_p-1}}{(m_p-1)!} \frac{1}{m_p}$$

for m arbitrary, which is equivalent to the expression in the Lemma. □

Proof of Proposition. Prob (K=j) is just the number of mappings with j cyclical points divided by the total number of mappings, n^n, which is, from (9),

$$\begin{aligned}
&\frac{\sum_{m=1}^{n} \sum n! \frac{j^{m_1} {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}}{m_1! m_2! \ldots m_p!}}{n^n} \\
&= \frac{n!\, j}{n^n} \sum_{m=1}^{n} \sum \frac{1}{j} \frac{j^{m_1} {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}}{m_1! m_2! \ldots m_p!} \\
&= \frac{n!\, j\, n^{n-j-1}}{n^n (n-j)!} \quad \text{(by Lemma)} \\
&= \frac{(n-1)!\, j\, n^{n-j}}{n^n (n-j)!} \\
&= \frac{(n-1)!\, j}{(n-j)!\, n^j}
\end{aligned}$$

where the inner sum is again taken over all sequences $(m_1, \ldots, m_p)$ such that the m_i's sum to $(n-j)$ □

Aside. We can use the above enumeration to study the probability that the short diagram of a uniform mapping from [n] to [n] is connected, and to give an alternative proof of formula (1) for $\#R_{k,n}$. The first of these results is due to Katz.

Proposition 9. The probability that a uniform mapping from [n] to [n] is connected is

$$\frac{(n-1)!}{n^n} \sum_{j=1}^{n-1} \frac{n^{n-j}}{(n-j)!}$$

Proof. Note that a connected mapping from [n] to [n] has only one cycle (since such mappings always have one cycle per component). We find the (compound) probability that a random uniform mapping is connected and has cycle length j, and then sum over j.

Fix $1 \leq j \leq n$. Let $M_0, M_1, \ldots$ be defined as above. Then for any sequence $(m_1, \ldots m_p)$ such that $\sum_{i=1}^{p} m_i = n - j$, the number of connected mappings from [n] to [n] where $M_0 = j$ and $M_i = m_i$ for $(1 \leq i \leq p)$ is clearly

$$\binom{n}{j\, m_1 \ldots m_p} (j-1)!\, j_1^m \, {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}$$

(We count the number of ways of choosing j cyclical points, m_1 vertices for M_1, m_2 vertices for M_2, etc., and then multiply by the number of ways to arrange the j cyclical points in a cycle, times the number of ways to attach the vertices in M_i to the vertices in M_{i-1} for $(1 \leq i \leq p)$. Note that attaching points in this way ensures that the mapping is connected.) Summing over all sequences $(m_1, \ldots m_p)$ where $\sum_{i=1}^{p} m_i = n - j$ and again over p, we see that the number of all connected mappings from [n] to [n] with j cyclical points is

$$\begin{aligned} &\sum_p \sum \binom{n}{j\, m_1 \ldots m_p} (j-1)! j_1^m \, {m_1}^{m_2} \ldots {m_{p-1}}^{m_p} \\ =&\sum_p \sum n! \frac{1}{j} \frac{j_1^m \, {m_1}^{m_2} \ldots {m_{p-1}}^{m_p}}{(m_1)! \ldots (m_p)!} \end{aligned}$$

which, by the Lemma, is just

$$n! \frac{n^{n-j-1}}{(n-j)!}$$

Dividing by the total number of mappings, and letting T be a a uniform random mapping from [n] to [n] we see that

$$P(\text{T is connected} \bigcap \text{T has j cyclical points}) = \frac{n!}{n^n} \frac{n^{n-j-1}}{(n-j)!}$$

Summing over j we now have:

$$\begin{aligned} P(\text{T is connected}) =& \sum_{j=1}^{n-1} P(\text{T is connected} \bigcap \text{T has j cyclical points}) \\ =& \sum_{j=1}^{n-1} \frac{n!}{n^n} \frac{n^{n-j-1}}{(n-j)!} \\ =& \frac{(n-1)!}{n^n} \sum_{j=1}^{n-1} \frac{n^{n-j}}{(n-j)!} \end{aligned}$$

where the upper index in the sum is $n-1$ (and not n) since the short diagram of a connected mapping on [n] with n cyclical points would have only $n-1$ edges. □

We would like to associate with each mapping $T : [n] \to [n]$ a rooted forest, and then, as in the previous section, study the induced distribution on the set of partitions of [n] when we take the uniform distribution on all mappings. For each such mapping, T, let D(T) be the short diagram of T. In general, as noted above, the components of D(T) consist of a single cycle and trees connected to the cyclical points, with all edges pointing toward the cycle. To transform this into a forest, we simply delete all edges between cyclical points, and reverse the direction of the remaining edges. In our new graph, R(T), all cyclical points are transformed into roots of component trees.

We can now define the function $\phi : J \to \bigcup_{k=1}^{n} R_{k,n}$ such that $\phi(T) = R(T)$ for all $T \in J$ where J is the set of mappings from [n] to [n]. Note that ϕ is not one-to-one (but is onto), since in general the roots of any given forest might have been attached to one another in the short diagram in any number of ways. Indeed, for any particular forest $r \in \cup_{k=1}^{n} R_{k,n}$ the cardinality of the set $\phi^{-1}(r) = \{T \in J \mid \phi(T) = r\}$ depends only on the number of tree components in r, and is equal to the number of ways in which we can group together the roots of each tree, and connect each group to form a cycle. We've seen above that for a forest with k roots, this is just the problem of dividing k people into cliques and seating each clique around a circular table. So for any forest r_k with k components, there are $k!$ ways of connecting the roots, hence $k!$ mappings that get sent to r_k by the function ϕ.

This means that if we take the uniform distribution on the set of mappings, J, the induced distribution on the set of rooted forests is uniform on $R_{k,n}$ for $(1 \leq k \leq n)$. Indeed, this yields another proof of the formula $\#R_{k,n}$, as we can write:

$$\begin{aligned}
\frac{1}{\#R_{k,n}} =& P(R(T) = r_k \mid K = k) \\
=& \frac{\text{\# of mappings that yield } r_k}{\text{\# of mappings with } k \text{ cyclical points}} \\
=& \frac{k!(n-k)!}{(n-1)!k\, n^{n-k}} \quad \text{(from proof of Proposition 8)} \\
=& \frac{1}{\binom{n-1}{k-1} n^{n-k}}
\end{aligned}$$

So $\#R_{k,n} = \binom{n-1}{k-1} n^{n-k}$.

Returning to our main line of argument, for fixed k we get the uniform distribution on $R_{k,n}$ for $(1 \leq k \leq n)$. Looking now at the induced distribution Γ_K on the set of all k-partitions of [n] for $(1 \leq k \leq n)$ (i.e. the partition of [n] induced by taking a uniformly distributed mapping from [n] to [n] and transforming it by ϕ into a forest), we have

$$P(\Gamma_K = \{A_1, ..., A_k\}) = P(K = k)\, P(\Gamma_K = \{A_1, ..., A_k\} \mid K = k)$$

Where the conditional probability is just the probability induced by the uniform distribution on $R_{k,n}$, and $P(K = k)$ is (from Proposition 8) $\frac{(n-1)!\,k}{(n-k)!\,n^k}$. Comparing this to (8) we see that , as shown by Pitman,

Proposition 10 The following constructions of a random forest, $F \in \bigcup_{k=1}^{n} F_{k,n}$ give the same induced distribution on partitions of [n]:

(A) Choose a refining sequence $(F_1, ..., F_n)$ uniformly from the set of all refining sequences of unrooted forests over [n], and let F be the kth component of this random sequence with probability $\frac{(n-1)!k}{(n-k)!n^k}$.

(B) Let T be a uniformly distributed mapping from [n] to [n], and let R(T) be the rootedd forest over [n] derived by deleting edges between cyclical points in the short diagram of T, and reversing edge directions. Finally, let F be the unrooted forest derived from R(T) by ignoring all edge directions.

References

[1] Leo Katz, "Probability of Indecomposability of a Random Mapping Function, *The Annals of Mathematical Statistics* Vol 26, No.3. (Sep., 1955)

[2] Bernard Harris, "Probability Distributions Related to Random Mappings," *The Annals of Mathematical Statistics*, Vol 31, No. 4. (Dec., 1960)

[3] Jim Pitman, "Coalescent Random Forests," Journal of Combinatorial Theory, Series A **85**, 1999

[4] Jim Pitman, "Enumerations of Trees and Forests Related to Branching Processes and Random Walks," Technical Report No. 482, Department of Statistics, Berkeley

[5] Miklos Bona, *Introduction to Enumerative Combinatorics*, 2007, McGraw-Hill, New York, NY.

[6] David P. Dobkin, Steven J. Friedman, and Kenneth J. Supowit, "Delaunay Graphs are Almost as Good as Complete Graphs," *Discrete and Computational Geometry*, 5, 1990

[7] Giri Rarasimhan and Michiel Smid, *Geometric Spanner Networks*, Cambridge University Press, 2007, New York, NY.

[8] Jim Pitman, "Random Forests and the Additive Coalescent," Chapter 9 of Book Notes