A Case Study in Approximation

Although it's not the focus of our class, I thought that it might be useful for you to see an example of how computers can be used in conjunction with mathematical techniques to develop approximations for reasonably difficult problems in combinatorics and probability. While I like to think that you have and will continue to see many such examples in your undergraduate career, I suspect that much of the material that I have in mind has been slowly withdrawn from your courses. A couple of our faculty were offering a course in mathematical modeling for some time, but I don't think that it ever made the leap into permanence.

The example that we'll consider is drawn from the experiments in Lab 2: developing an analytical model of the probability that a path exists between the two corners of a random lattice graph. The direct practical relevance is fairly low, although there is some utility for reasoning about simulation methods for critical phenomena (e.g., magnetism near critical temperatures). The main point of this note is to give you an example of the process of using computers to gain insight into problems (see Hamming).

Before we tackle the problem, I want to give you an overview of the process, particularly because the coverage here does not reflect the relative effort on the various elements of the process. As Polya states in the introduction to his book *How To Solve It*, "…mathematics in the making appears as an experimental, inductive science." The bulk of the effort in constructing a proof or approximation is spent identifying and exploring promising possibilities, most of which turn out to be dead ends. For the person trying to solve the problem, these efforts help to reduce the search space both directly and indirectly, and computers are a useful tool in checking and validating approaches and results as well as in computing supporting information such as complex distributions. Some of this effort and value is described as part of the resulting derivation, but most negative results are omitted due to their questionable value: would any capable researcher pursue a given approach in trying to solve the problem, or is the approach clearly flawed and pointless from the outset? Similarly, although computers can help in identifying

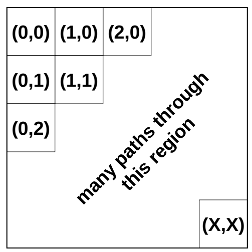
patterns that can be generalized, proving a generalization again requires effort, while only the simplest or most elegant proof is necessarily of interest to a reader. Now on to the problem…

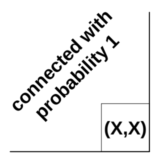
Random Lattice Phase Change

We form a two-dimensional lattice graph of size N×N by placing nodes at all positions (X,Y) in a plane such that X and Y are integers in the range [0,N). Directed edges in the graph are present with probability P from any node (X,Y) to neighbors (X+1,Y) and (X,Y+1) if those neighbors exist in the lattice. All such edge probabilities are independent, and no other edges exist. (Note that the critical phenomena studies use undirected edges, which changes the nature of the problem; undirected edges also produce cyclic structures that make reference-counted garbage collection impossible.)

As part of the Lab 2, I gave you code to estimate the probability that a path from (0,0) to any diagonal element (X,X) exists in such a graph with given edge probability P. The last question, as you probably recall, asks you to identify the value of P at which the shape of the "curve" defined by these probabilities changes from something rapidly approaching zero to something roughly constant. Let's explore the reason behind this phase change and try to approximate the behavior in the two phases.

Let's start with the phase in which the connection probability seems to be independent of X for large values of X. In this type of graph, the probability P is high enough that most of the graph can be reached from the corner (0,0). We can approximate the probability by assuming that if both corners connect to the bulk of the lattice, they also connect to one another. In other words, the likelihood that no path can be found through the lattice is negligible. We'll build up terms by starting with the simplest possible assumptions and then weakening these assumptions step by step to get more accurate estimates of the path probability.

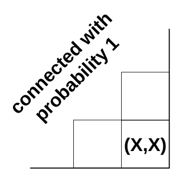




A diagram of the general graph structure appears on the left above. On the right is a diagram for our first approximation. Define probability $Q = 1 - P$ to be the probability that a given edge is not present in the graph. If we assume that the entire graph is connected with the possible exception of the corners (0,0) and (X,X), we can write the probability that a given corner (they're symmetric) is not connected as Q^2. In particular, (X,X) is disconnected only if neither of its neighbors connect to it, and each event occurs independently with probability Q. Note that this approximation is accurate to within factors of Q^3. For $Q \ll 1$ (Q much smaller than 1), the approximation is quite good.

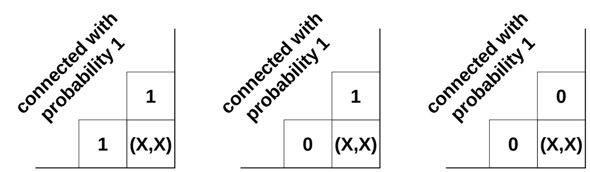
The probability that a path exists from (0,0) to (X,X) is then the probability that neither corner is disconnected, or $(1 - Q^2)^2$.

To refine our approximation, we add a second level of exact calculation: as shown to the right, we now assume that elements other than the corner and its neighbors are connected with probability 1. Each of the two neighbors is then disconnected with probability Q^2, and these probabilities are independent (they are based on disjoint sets of edges in the graph).



ECE498SL Spring 2009 Note #3 25 March 2009

We can calculate the overall probability that the corner is disconnected by adding the probabilities of disconnection in the four disjoint parts of the probability space obtained by considering all combinations of neighbor node connectivity. If both neighbor nodes are connected to the bulk of the mesh, for example, the probability that the corner is disconnected reduces to our earlier approximation, Q^2. This case is illustrated by the left figure below. As this particular neighbor connectivity occurs with probability $(1 - Q^2)^2$, the first of our four terms in our sum is simply $(1 - Q^2)^2 Q^2$. Since the neighbors are symmetric, the next two terms are identical, and are represented by the middle figure below. When only one neighbor is connected, corner disconnection occurs with probability Q. Given that the chance of one particular neighbor (but not the other) being connected is $Q^2 (1 - Q^2)$, the two symmetric terms sum to $2 (1 - Q^2) Q^3$. Finally, the diagram on the right represents the case in which neither neighbor is connected, in which case the corner is disconnected with probability 1. This term adds probability Q^4.

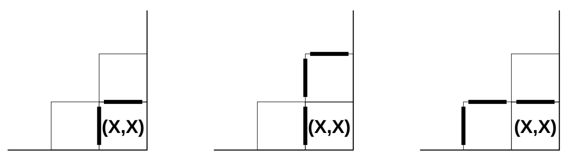


The complete sum is then

$$
\begin{aligned}
& (1 - Q^2)^2 Q^2 + 2 (1 - Q^2) Q^3 + Q^4 \\
= \; & Q^2 - 2Q^4 + Q^6 + 2Q^3 - 2Q^5 + Q^4 \\
= \; & Q^2 + 2Q^3 + \text{higher order terms}
\end{aligned}
$$

Note that the coefficients on higher order terms will change with further refinement, whereas this approximation captures all effects up to Q^3. With this refinement, our path connection probability becomes $(1 - Q^2 - 2Q^3)^2$.

At this point it's worth recalling a property of probabilities that we'll need later. Rather than the method we used to derive a two-term approximation, we could have simply drawn all of the small combinations of edge absences that cut off a corner, as shown below.

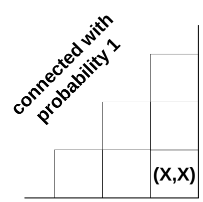


The probability of the figure on the left is Q^2, and those of the others are both Q^3; that is, one power of Q per edge in the cut. By summing these three terms, we get our approximation, and we easily see that no other 2- or 3- edge cuts can isolate the corner. However, the chance of either of two events happening is not their sum unless the two events never happen together. In general, for events A and B,

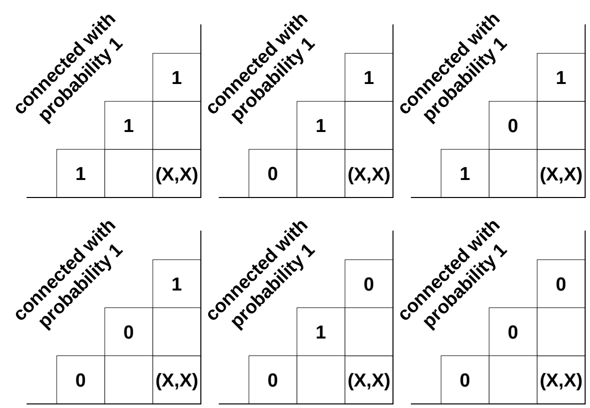
$$\text{prob(A OR B)} = \text{prob(A)} + \text{prob(B)} - \text{prob(A AND B)}$$

In the case of our problem, the last term is easily calculated by simply taking the union of the two edge sets. Either of the 3-edge cuts together with the 2-edge cut forms a set of four edges, which means that we should be subtracting Q^4. These terms did appear in our sum, but we chose to ignore them until we've further refined our approximation.

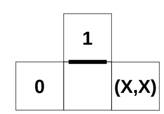
For the next (and final) stage of refinement, we again add an extra level of graph nodes to our calculation. The rest of the graph is still assumed be connected with probability 1, and all three border nodes are again disconnected from the bulk of the graph with probability Q^2 independent of the other border nodes.

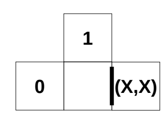


We start this time with eight terms, with two symmetric pairs. The combinations are shown in the figure below. As before, when all border nodes are connected, the problem reduces to our previous calculation.



As an aid to evaluating the other combinations, we first calculate a couple of simple cases. The path to the corner has to be blocked through both of its neighbors. If the neighbor is adjacent to exactly one connected border node, the path can be blocked by one of two edges (as shown below), and the total chance of blocking is $2Q - Q^2$. The Q^2 represents the chance that both edges are absent.





Similarly, if a neighbor node is adjacent to two connected border nodes, we can calculate the probability that no path exists through that neighbor (using the cuts shown below) to be $Q + Q^2 – Q^3$.

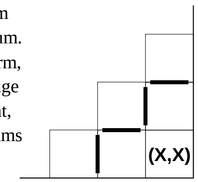
Note that $[Q + Q^2 – Q^3]^2$ is identical to our previous result (before dropping higher order terms). Using these two equations, we can now simply write down the answer term by term (right to left, top to bottom in the diagram of combinations).

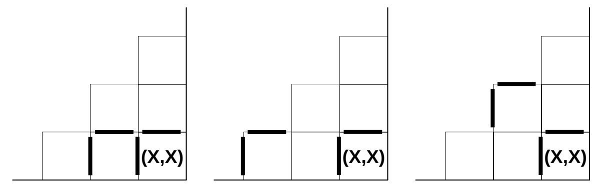
$$
\begin{aligned}
& (1 – Q^2)^3\,[Q + Q^2 – Q^3]^2 \\
+\ & 2Q^2\,(1 – Q^2)^2\,(2Q – Q^2)\,(Q + Q^2 – Q^3) \\
+\ & Q^2\,(1 – Q^2)^2\,(2Q – Q^2)^2 \\
+\ & 2Q^4\,(1 – Q^2)\,(2Q – Q^2) \\
+\ & Q^4\,(1 – Q^2)\,(2Q – Q^2)^2 \\
+\ & Q^6
\end{aligned}
$$

Only the first term above contributes terms of order below Q^4, and only the first three terms contribute terms of order below Q^5. It's worth exploring the meaning of these terms graphically so as to get some intuition of exactly how hard it can be to get right. Try to count all 4-edge cuts (positive) as well as all 4-edge unions of two cuts (negative) before going on.

ECE498SL Spring 2009 Note #3 25 March 2009

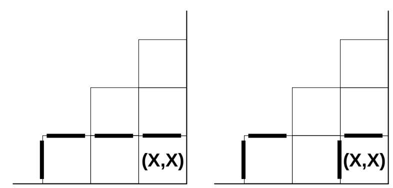
Dropping higher-order terms, the first term from our sum contributes a total of $Q^2 + 2Q^3 – 4Q^4$ to the final sum. If you look carefully at the products that make up the term, however, you'll note that it actually consists of one 4-edge cut and five 4-edge unions. The cut is shown to the right, and the unions are shown below (two of the three diagrams represent two symmetric unions, for a total of five).



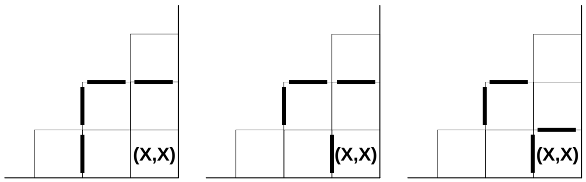


Note that the two diagrams on the right include edge pairs that disconnect border nodes; the same sets of edges appear in later terms in our sum as 4-edge cuts, but are cancelled in this term since the 4-edge cut is a proper superset of the 2-edge cut.

The second term from our sum contributes $4Q^4$ to the final sum in the form of four 4-edge cuts shown in the diagrams below (the contribution also includes the symmetric counterpart of each diagram).



The third term from our sum also contributes 4Q^4 to the final sum in the form of four 4-edge cuts shown in the diagrams below (the contribution also includes the symmetric counterpart of the middle diagram).



As mentioned before, none of the remaining terms produce results of order less than Q^5, so our final approximation is given as Q^2 + 2Q^3 + 4Q^4, and our path connection probability becomes (1 – Q^2 – 2Q^3 – 4Q^4)^2. A graph of our three increasingly accurate estimates compared with simulation data for a 20×20 graph with P=[0.8,1] appears on the next page.

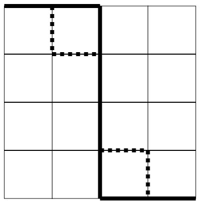
Evaluating for Low Values of P

We now turn to the other phase of the random lattices, in which P is low and path probabilities are vanishingly small. Gathering good data here is difficult, since many attempts are necessary to generate any successes with simulation. Given an N×N lattice, a path of length 2N – 2 is necessary to travel from (0,0) to (N – 1, N – 1). For any given path, the probability of that path being present (independent of all other graph edges) is P^(2N – 2).

More than one path is possible, of course. How many paths must we check? Think of the problem in the context of our earlier drawings as follows: to move from (0,0) to (N – 1, N – 1), we must move right N – 1 times and down N – 1 times. In other words, out of 2N – 2 moves, we must choose N – 1 of them to be to the right (the others are down). Any such combination represents a distinct path, so we have a total of C(2N – 2, N – 1) = (2N – 2)! / [(N – 1)!]^2 possible paths.

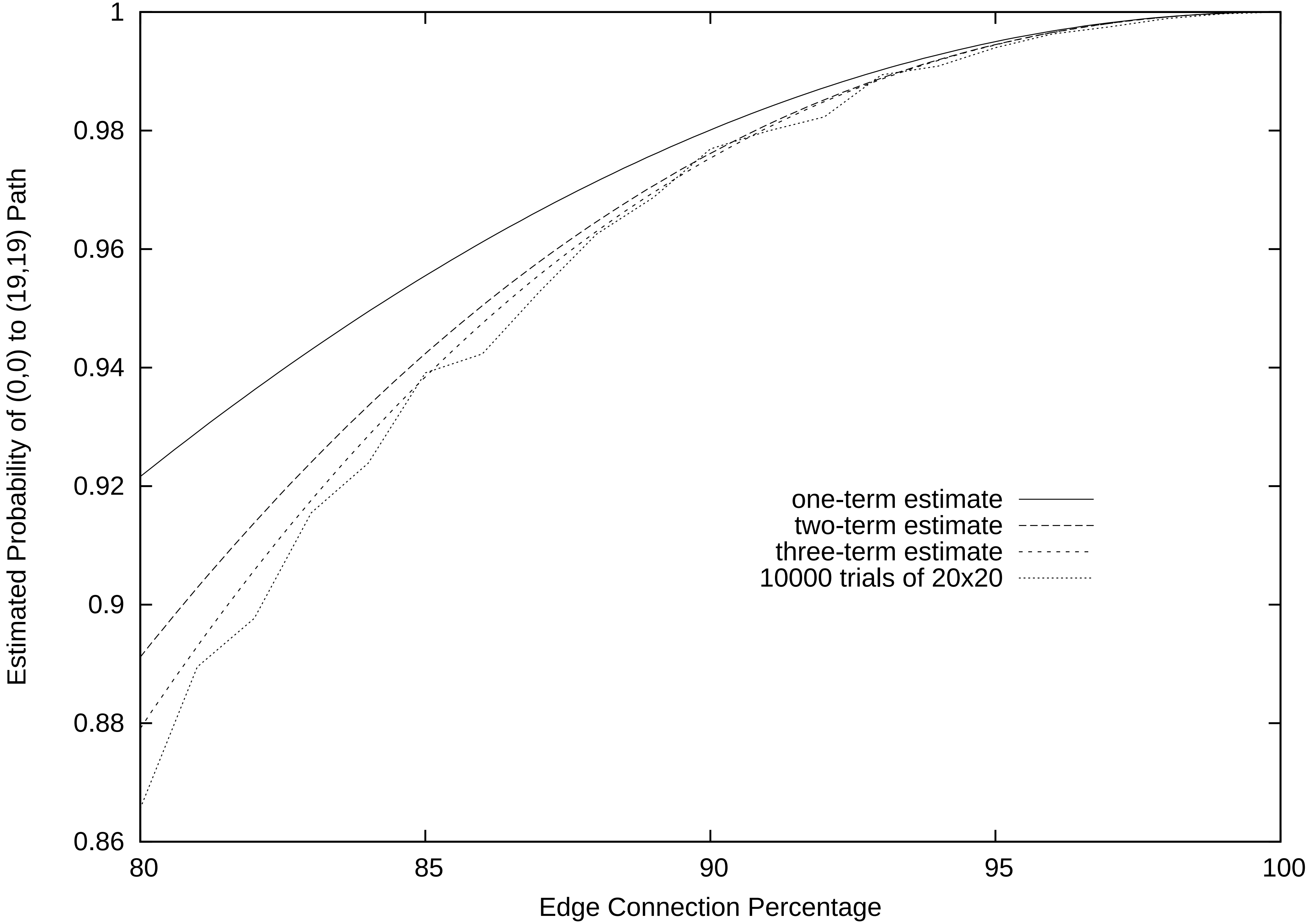
Our first estimate for small P assumes that we can treat these probabilities independently and simply add them up: C(2N – 2, N – 1) P^(2N – 2).

Removing the assumption of independence is not easy. Dynamic programming—for example, to calculate probabilities based on neighboring nodes—does not yield useful refinements. What does help, at least to obtain the next term, is to realize that only interaction between the “closest” paths matters much. Consider the diagram below representing a path on a 5×5 graph. Note that the lines now represent edges, and nodes occur at intersections rather than appearing as boxes (as in our previous diagrams).



As you may recall from our previous efforts, the amount to be subtracted from our first estimate depends on the size of the edge set generated by the union of two paths. Paths that have little or no overlap create unions of nearly double the size, and thus have roughly the square of the probability. With P << 1, we need as small a union as possible for the difference to matter. Adding another edge to a path does not create another path, but consider the dotted extensions in the figure above: each represents an alternate path such that the union with the bold path forms a set with only two extra edges. Each pair of paths with such an intersection implies subtraction of P^(2N) from our total probability.

How many such path pairs exist? Notice that such a pair is implied by each corner in the path shown, and in fact is implied by every corner of every path. If we count the total number of corners in all paths and divide by two, we have our answer. Start counting!



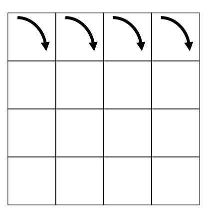
Estimated Probability of (0,0) to (19,19) Path
1
0.98
0.96
0.94
0.92
0.9
0.88
0.86
80
85
90
95
100
Edge Connection Percentage
one-term estimate
two-term estimate
three-term estimate
10000 trials of 20x20

Not so easy? I had to use other methods, including computer calculation of distributions for small lattices, to figure out what form the answer took. Then I generalized it a little on paper. Then I realized what it was supposed to be and figured out how to prove it. I'll give you the proof and spare you the rest.

Here's another way to think about paths. Let a 1 represent a move to the right, and a 0 represent a move to the left. A path in an N×N lattice consists of N – 1 ones and N – 1 zeroes in an arbitrary sequence in a string. If you start with the N – 1 zeroes, you can imagine putting the 1s before, between, or after the 0s to form a path.

Now: how many of the paths have 1s before 0s? That's an easy answer: half of them, since you can flip all of the bits and have another path, and all paths have exactly one such symmetric path. But the first place (before all 0s) is no different from the second place (between the 1st and 2nd 0s) or the third place. So if I ask: how many paths have 1s between any pair of 0s, the answer is the same: half of the paths.

Why is that important? Let's look back at our diagram (shown again below). Paths with at 1s before 0s correspond to paths that turn an upper corner (the arcs) in the first row. Paths with 1s between the 1st and 2nd 0s correspond to paths that turn an upper corner in the second row, and so forth. But that gives us a count of corners! We have half of the paths appearing in path pairs for each row of the graph. With N – 1 rows, we have a total of (N – 1) C (2N – 2, N – 1) / 2 path pairs.



Thus we find our refined approximation:

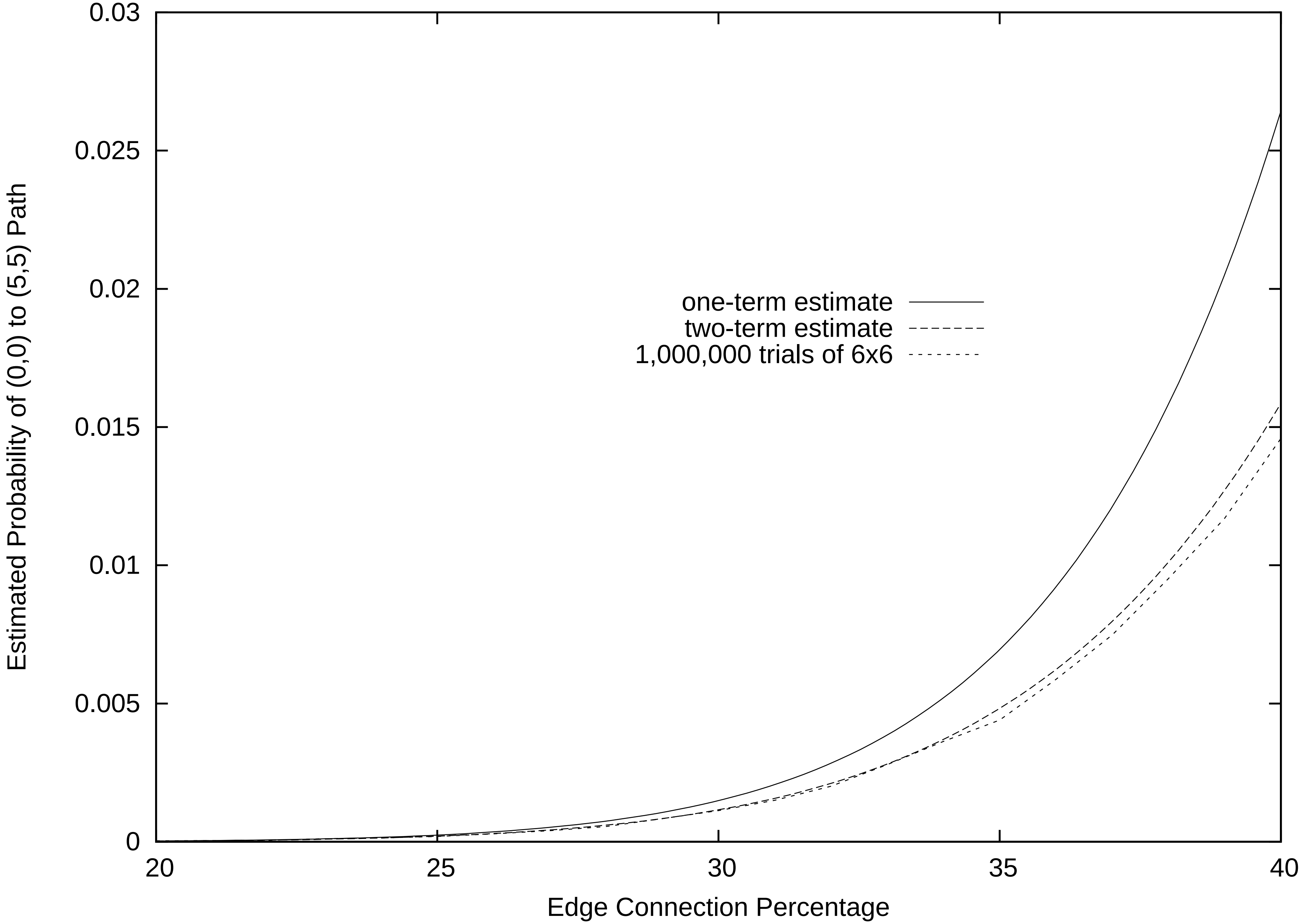
C(2N – 2, N – 1) P^(2N – 2) [1 – (N – 1) P^2 / 2]

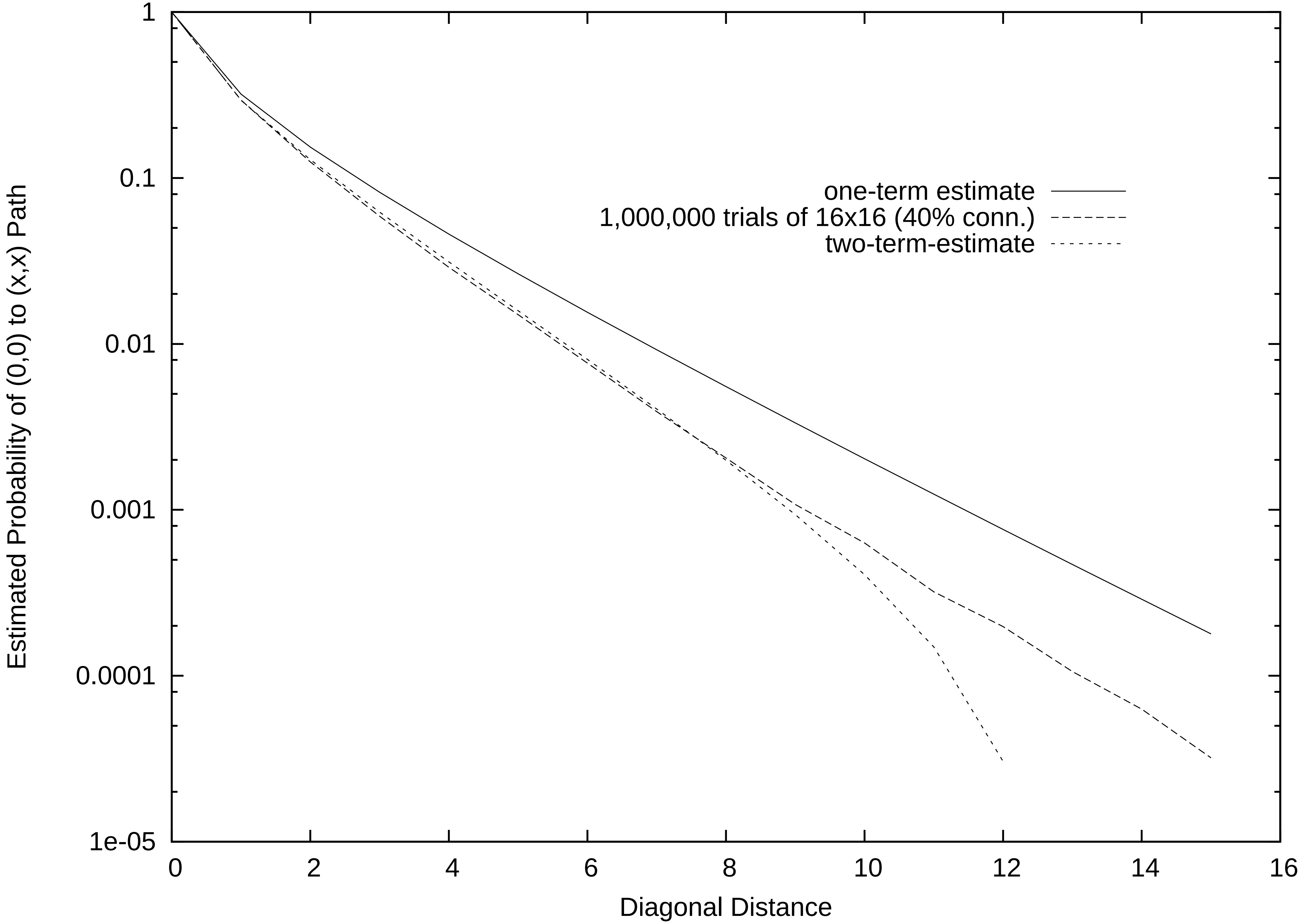
Two graphs using these estimates appear on the following pages. First, we plot simulation data for a 6×6 graph with P=[0.2,0.4]; note the use of 1,000,000 trials per point for the simulation data. The second graph shows the estimates against simulation results for a fixed value of P, in this case 0.4. The graph uses a logarithmic scale to show the relationship clearly over the large dynamic range. Note that the two-term estimate becomes unreliable when (N – 1) P^2 / 2 approaches 1, which is fairly quick at P=0.4. The one-term estimate is fairly inaccurate from the start and gets worse as we move further down the diagonal.

That's the end for now.

(4 May 2009)

I explored the problem further after that day, but never got around to writing up everything that I found. Basically, you can model the problem as an infinite line of 0s and 1s (1s are connected, 0s are not) and then find fixed points (equilibrium distributions) given certain assumptions about independence/dependence. You can also write a program that explores such a model (with finite bit width, of course) but runs much more quickly than the graph program, allowing exploration of 10s or 100s of thousands of steps, at which point the transition is sharp. Except for the possibility of 1s flowing back in from the boundaries, which can't happen in the bit model, this system should be identical to the graph program. However, while the phase change point matches pretty well, and the high-probability density matches pretty well, the shape of the curve between the threshold and high probability is different (second derivatives had opposite sign for part of it). The reason is actually that the graph program never reaches steady state; modeling a short run with the new program produces an almost identical curve. The next few pages are analytical results for steady-state (fixed-point) solutions.

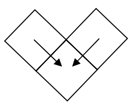

0.03
0.025
0.02
0.015
0.01
0.005
0
20
25
30
35
40
Estimated Probability of (0,0) to (5,5) Path
Edge Connection Percentage
one-term estimate
two-term estimate
1,000,000 trials of 6x6


Estimated Probability of (0,0) to (x,x) Path
Diagonal Distance
one-term estimate
1,000,000 trials of 16x16 (40% conn.)
two-term-estimate
1
0.1
0.01
0.001
0.0001
1e-05
0
2
4
6
8
10
12
14
16

Fixed-Point Approximations

We had assumed that the probability of any given square's being connected in the high probability region to be 1 if the upper left corner connected. In practice, however, even if most of the squares are connected, many of them will not be. Here we attempt to calculate a more accurate probability by treating the problem as an infinite line of squares and calculating the fixed-point in probability between successive lines. We'll first derive a simple model. Plugging this probability into the high-probability region models (which takes a little work), by the way, improves the quality of those curves. However, after the simple model, we'll move on to a more complex model. I've been working on an even more complicated one, but have yet to finish it.

Let F denote the probability that a given square in an infinite line is connected; for the purposes of this discussion, let's say contains a 1. The value 0 then represents squares that are not connected. The question, then, is as follows: given a value of P, what value of F produces the same value of F in the next line?



As shown at the right, a given square can become a 1 if either of its incoming neighbors has a 1 and the square connects to that neighbor (with probability P). We can then write the overall probability that a square contains a 1 as the probability that either neighbor contains a 1 and connects to the square, or 2FP – (FP)^2. But this probability must equal F in the steady state, so F = 2FP – (FP)^2, which leads to two solutions: F = 0 or F = (2P – 1) / P^2. We have our first threshold: below P=0.5, F=0.

That's it for the first model. The model is not particularly accurate because it fails to account for correlations in adjacent elements. If a particular square contains a 1, the chance that its neighbor also contains a 1 is not F, as the two are related by a common ancestor. In order to capture these dependencies, we need a more complex model. In particular, let G be the probability that a square's neighbor containss a 1 given that the square itself contains a 1 (note that our process is symmetric, so it can't matter which of the square's two neighbors we consider).

We'll write S for the square, and N for the neighbor. Then

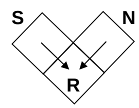
prob (N = 1 | S = 1) = G
prob (N = 0 | S = 1) = 1 – G

Due to symmetry, prob (N = 1 and S = 0) = prob (N = 0 and S = 1), so

prob (N = 1 | S = 0) = prob (N = 0 | S = 1) prob (S = 1) / prob (S = 0)
= (1 – G) F / (1 – F)
prob (N = 0 | S = 0) = 1 – (1 – G) F / (1 – F)

Now let's start to use these probabilities to calculate a fixed-point solution. Let R be a square that depends on some square S and its neighbor N. We have

prob (R = 1) = prob (R = 1 and N = 0 and S = 0) +
prob (R = 1 and N = 1 and S = 0) +
prob (R = 1 and N = 0 and S = 1) +
prob (R = 1 and N = 1 and S = 1)



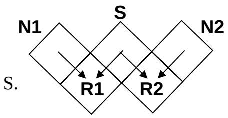
We expand each using Bayes' theorem. For example:

prob (R = 1 and N = 1 and S = 1)
= prob (R = 1 | N = 1 and S = 1) prob (N = 1 | S = 1) prob (S = 1)
= (2P – P^2) G F

Replacing term for term, we obtain:

prob (R = 1) = 0 +
P [(1 – G) F / (1 – F)] (1 – F) +
P (1 – G) F +
(2P – P^2) G F

But prob (R = 1) is equal to F for a fixed-point, so we find F = PF (2 – PG), and solutions are F = 0 or G = (2P – 1) / P^2. But what is F in the second case? We need to constrain G to be equal in the next row to find the answer.



Consider the diagram shown to the right. For a fixed-point solution, our extra constraint can be written as FG = prob (R2 = 1 and R1 = 1).

Here it's useful to have the probability of R given S. From before, we have

prob (R = 1 and N = 0 | S = 0) = 0
prob (R = 1 and N = 1 | S = 0) = P [(1 – G) F / (1 – F)]
prob (R = 1 and N = 0 | S = 1) = P (1 – G)
prob (R = 1 and N = 1 | S = 1) = (2P – P^2) G

thus

prob (R = 1 | S = 0) = P [(1 – G) F / (1 – F)]
prob (R = 1 | S = 1) = P (1 + G – GP)

and

FG = (1 – F) { P [(1 – G) F / (1 – F)] }^2 + F { P (1 + G – GP) }^2
= F^2 (1 – G)^2 P^2 / (1 – F) + F (1 + G – GP)^2 P^2

Recall that GP^2 = 2P – 1 and solve for F to obtain (eventually)

F = (-P^2 + 4P – 2) / (2P – 1)

Note that the threshold is now at P = 2 – sqrt(2) = 0.586. I think that if one carries the complexity further (with more dependency information), one can get close to the actual threshold, although again the shape of the density curve in the graph problem will be different, as the process does not reach steady state in a square matrix.