A Class of Non-planar Recoverable Electrical Networks

John Hardin

August 13, 2010

Abstract

This paper examines the generic recoverability of various classes of non planar Electrical Networks. These include certain types of networks which have every vertex of degree 4 and arranged in a lattice, or "Permutation Lattice Networks" (Other graph restrictions apply). Also examined are cross networks, which are essentially constructed from the various types of two by two lattice possibilities. Generalizable techniques for building these and other kinds of recoverable non planar Electrical Networks are discussed.

1 Electrical Networks

An electrical network is a weighted graph in which some vertices are assigned to be boundary and some interior [1, p.2]. This is defined in the book by Curtis and Morrow, which will be the major source for this background information. The weights correspond to the conductivities between the vertices.

1.1 Matrix Representation

An electrical network can be represented in a Kirchoff matrix, which is a symmetric square matrix with row/column sum zero. It has as many rows as there are vertices in the network. The off-diagonal entries are equal to the negation of the conductivity between a vertex and the neighbor corresponding to the index of the entry (zero if no edge exists) [1, p.33]. By row/column sum being zero, the diagonal entries are therefore the sum of all the conductivities of the edges connected to the vertex. In our convention, the matrix will be ordered such that the boundary vertices will come first in the indexing. This produces a matrix which can be blocked as follows:

$$K = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

A includes to boundary-boundary edges, B includes boundary-interior edges, and C includes interior-interior edges. The Dirichlet to Neumann map, which takes boundary voltages and returns boundary current can be represented by a matrix, Λ, which is the Schur complement of K with respect to C [1, p.40]. Explicitly:

$$\Lambda = A - BC^{-1}B^T$$

The entries in this matrix and shape of the graph (where the zeros are in the Kirchoff matrix) is the information from which we are trying to recover the conductances of each edge of the graph.

It is important to note that:

$$det(\Lambda) = \frac{det(K)}{det(C)}$$

Further, if I is the set of interior vertices, S and T are sets of boundary vertices of equal size, and $K(S+I;T+I)$ corresponds to the submatrix of K including only the rows in S and the interior and the columns in T and the interior, then:

$$det(\Lambda(S;T)) = \frac{det(K(S+I;T+I))}{det(K(I;I))}$$

This formula is found in section 3.6 of the book [1, p.47]

1.2 Major Recovery Methods

Before discussing recovery methods, it is important to define how connections relate to the existence of a non-zero sub-determinant of K.

Definition 1 *A connection from S to T is defined as a set of "non-intersecting" paths from S to T. "Non-intersecting" in this context means the one of the paths leaves each vertex in S exactly once, enter each vertex in T exactly once, and all of the paths together enter and leave each interior vertex at most once, and no paths enter or leave boundary vertices not contained in S or T.*

[1, p.50]

When visualized, this looks like n connected directed paths through the interiors, where n is the size of the set S. Each monomial in a determinant corresponds to a permutation of the rows over the columns. of the matrix. Since the Kirchoff matrix has quite a few entries of zero, only a few of the $n!$ possible monomials contribute (n is the size of one side of the matrix). By the connection determinant formula, we know that these monomials correspond to possible paths for a given connection to take [1, p.50]. We know that this determinant is generically non zero (as least one monomial contributes) if the connection exists [1, lemma 3.12] Specifically, if an edge must be used by a connection, then it must multiply in every term of the monomial. This is equivalent to stating that it is the only term that contributes from its row and column in the permutation interpretation of determinants. This allows us to quickly derive the boundary spike contraction and boundary edge deletion formulas [1, section 3.8].

1.2.1 Boundary Spike Recovery

Definition 2 *A boundary spike is the edge which connects a boundary vertex of degree 1 to the interior.*

To recover the conductivity of a boundary spike, call the boundary vertex it connects p, then find sets of boundary vertices S and T such that a connection between them exists and such that they must use the interior vertex connected by the boundary spike. Then compute $det(\Lambda(p+S;p+T))$:

$$det(\Lambda(p+S;p+T)) = \frac{det(K(p+S+I;p+T+I))}{det(K(I;I))}$$

But since every connection from S to T must use p's corresponding interior node, only p's diagonal element can contribute to the determinant. Specifically, the determinant is this element multiplied y the sub-determinant:

$$det(\Lambda(p+S;p+T)) = k_{pp}\frac{det(K(S+I;T+I))}{det(K(I;I))}$$
$$det(\Lambda(p+S;p+T)) = k_{pp}det(\Lambda(S;T))$$
$$k_{pp} = \frac{det(\Lambda(p+S;p+T))}{det(\Lambda(S;T))}$$

This gives us the boundary spike formula [1, p.57]. Note that the book gives the formula in a slightly different form, but the two are equivalent.
To modify the response matrix and contract the spike so that the matrix corresponds to the new network with a boundary node at the interior vertex at the other end of the spike. Explicitly [1, p.106]:

$$let\Lambda = \begin{pmatrix} \lambda_{pp} & a \\ b & C \end{pmatrix}$$

$$\Lambda_{new} = (\lambda_{pp} - \alpha)\begin{pmatrix} \alpha\lambda_{pp} & \alpha a \\ \alpha b & (\lambda - \alpha)C - ba \end{pmatrix}$$

where Λ_{new} is the new response matrix, and α is the recovered conductivity.

1.2.2 Boundary-Boundary Edge Recovery

Boundary edge recovery proceeds in much the same way that boundary spike recovery did. to recover a boundary edge between p and q, first find a connection S to T which has neither p nor q in either set, but disconnects p and q through the interior. This implies that the connection $p+S$ to $q+T$ must use the boundary-boundary edge. We can then proceed to manipulate the determinants as before:

$$det(\Lambda(p+S;q+T)) = \frac{det(K(p+S+I;q+T+I))}{det(K(I;I))}$$

$$det(\Lambda(p+S;q+T)) = k_{pq}\frac{det(K(S+I;T+I))}{det(K(I;I))}$$
$$det(\Lambda(p+S;q+T)) = k_{pq}det(\Lambda(S;T))$$
$$k_{pq} = \frac{det(\Lambda(p+S;q+T))}{det(\Lambda(S;T))}$$

[1, p.56] Modifying Λ to represent the new matrix with the edge deleted is then very simple, as we are modifying only block A, and so we just subtract the recovered value from the diagonals p and q, and add it to λ_{pq} and λ_{qp}. Boundary spike contraction and boundary edge deletion are the two major methods of recovery for graphs.

1.2.3 Open Subgraph Recoverability

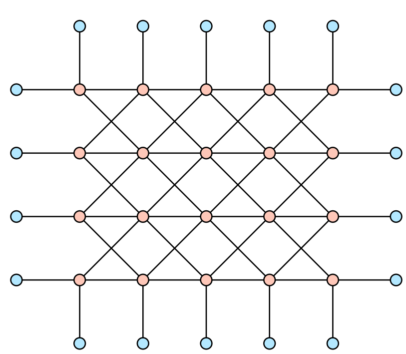
It is important to note that any open subgraph of a recoverable graph is also recoverable [3].

Definition 3 *An open subgraph of an electrical network Γ is a subgraph Γ_{sub} such that if a vertex does not have the same edges that it did in Γ, then it is a boundary vertex in Γ_{sub}.*

Note that the paper referenced demonstrates that a Simon Subgraph of a recoverable graph is recoverable. the only difference between a Simon subgraph and an open subgraph is that the Simon Subgraph must contain boundary boundary edges of the original graph. Since the original graph is recoverable, we can recover these boundary edges directly and modify the response matrix. This gives us the open subgraph result.

2 Recoverable Classes of Lattice Graphs

A lattice graph here is defined by the geometric orientation of its interior vertices. The vertices are place at unit distance from each other and the connected. This orientation is of course purely aesthetic, but it makes visualizing the connection easier. Two types of boundary configurations will be considered: one with "fringe" boundary spikes, and one with these spikes contracted. The former is pictured below and the latter will be discussed later.



The plausibility of this recovery is no immediately obvious. There appear to be many edges relative to the number of boundary nodes, and any graph with more independent edges than independent entries in the response matrix is trivially non recoverable. However, by examining the number of possibly independent entries in the response matrix, it can be seen that the desired number of recovered edges is much less than the potentially available independent pieces of information. If we consider a square lattice graph where each node is connected to all 8 immediate neighbors, then there are at most 4 unique edges for each vertex. For a square graph this gives us:

$$4n^2$$

edges as a hard max (the real value is a little less, but this is the dominant term in the expression. For comparison, note the if a graph has q boundary vertices, there are up to the $(q-1)th$ triangular number of independent entries in the response matrix. Since a square lattice graph has 4n vertices, it has:

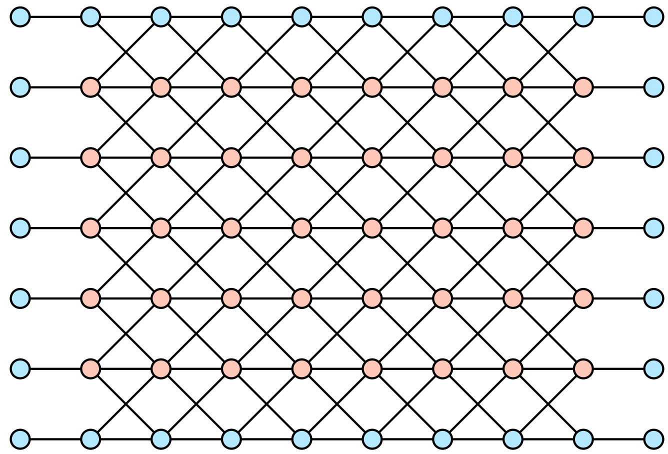
$$\frac{4n(4n-1)}{2} \sim 8n^2$$

possible independent entries. This is well above the lower limit found above. This provides a cursory argument for the plausibility of recovery.

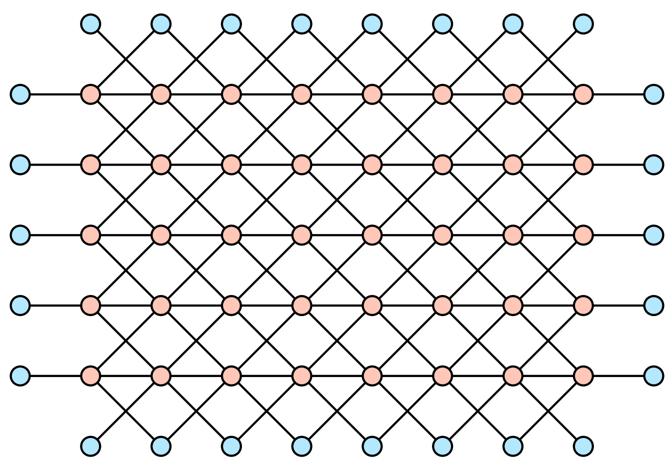
For the purposes of the this paper, orientation of the graph will be referred to using the directions on the compass rose with north corresponding to the top of the page. Recovery will be performed using layer stripping by deleting boundary-boundary edges/contracting boundary spikes based on the fact that these actions will break connections. Let the number of original interior East-West vertices be n and the number of original interior North-South vertices be m.

2.1 Grained Cross Lattice Graphs

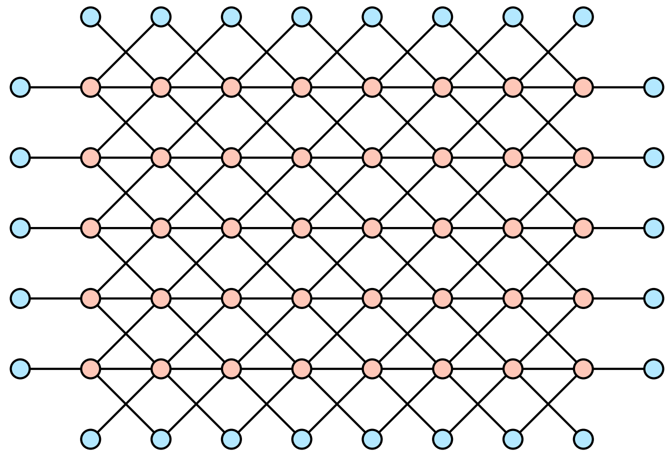
A grained cross lattice graph Recovery of a cross Lattice graph (like the one pictured above) starts by talking all East boundary vertices and connecting them to all West boundary vertices. This connection exists, as no crossing must be made, so all can be taken straight across, this allows contraction of the North and South boundary spikes. Each of the m western boundary vertices must traverse n vertices in some way to get to the eastern ones. Since these vertices must be disjoint by the definition of connection, nm interior vertices must be used by any connection. Since there are only nm interior vertices total, all must be used, and contracting any of the North or South spikes will break the connection. A sample graph with $n = 8$ and $m = 7$ is shown below.



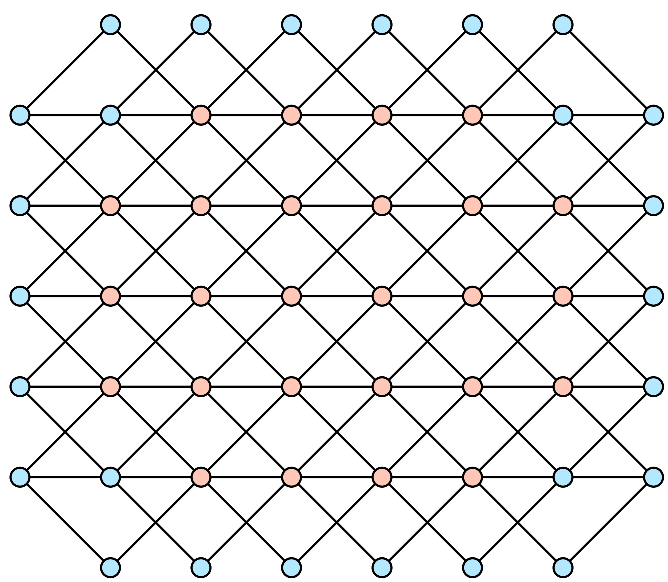
Now there are boundary nodes with no path to each other through the interior. They can be trivially recovered by being read off of the response matrix. Then another walk across the graph from east to west using the middle $m-2$ vertices must used all the remaining interior vertices by analogous reasoning. Then to remove the boundary-boundary edges, one boundary vertices is added to one side of the connection and its neighbor through that edge added to the other. Then deleting the edge would break the connection, and so it can be recovered. Once this is performed the graph is further reduced and, in the current example, looks like the below.



Once this is performed, the corner boundary spikes can then be contracted by another east-west walk across. This result is show with the current example below:



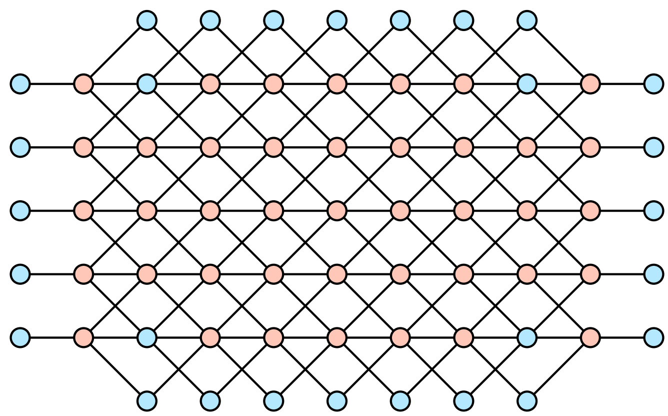
Now the recovery has to be broken down into two cases: n is even and n is odd. If n is even the connection from North to the South must be paired together for a march down. These include the recently contracted spikes, which will be paired with there immediate vertical neighbors, others will be paired with their neighbors going across. Since diagonals must always be taken, as the interior boundary vertices will be saturated by the above pairing by the same reasoning as before. The path exists, and so the East-West boundary spikes can be contracted. In the example this looks like:



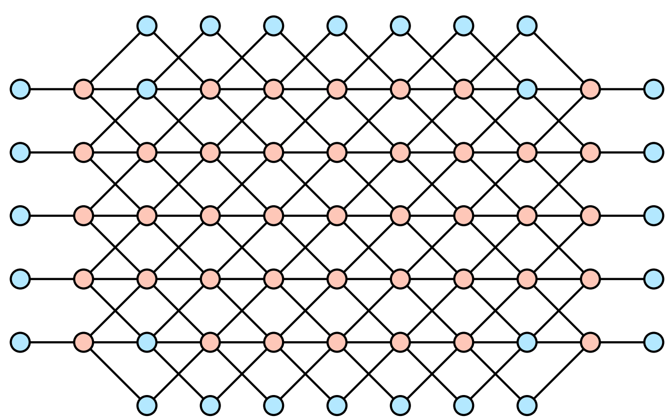
Once the graph looks like this, parity no longer matters, as the $m-2$ East West node closest to the interior in each row can be connected to its opposite. This allows isolation of boundary boundary edges and the contraction of the resultant spikes. Thus the rows can be pulled in from top to bottom by induction.

Now all that is left is to consider the odd case and reduce it to the above picture. In the odd case, there is not an obvious connection between all North and all South vertices, as they will collide in the first row. The diagonals force a parity switch for the paths, and there are not equal numbers of "even" and "odd" columns in each row.

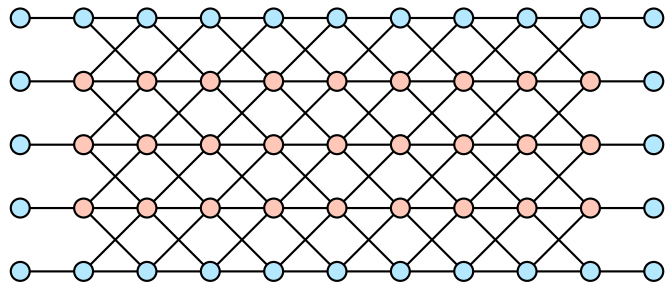
In order to isolate the nodes, it is necessary to force a connection to use a boundary-boundary edge. This edge can then be recovered and ignored. This allows that boundary vertex to then be treated as a spike and recovery can continue. In order to pick the connection properly, the South nodes shall be assigned parity odd and even, starting from the left with even. All of these will be on one side of the connection. The remaining east vertices, along with as many North vertices as necessary to fill out the connection (the North vertices being as far to the east as possible), will be placed on the other side of the connection. The the most easterly odd node has only 1 path to the inner corner boundary node, and it is over the boundary boundary edge:



Once this is recovered, there are spikes on the boundary, which can be recovered with horizontal walks:



Once the process is repeated to make more spikes, and contract more edges until the entire South (or North) boundary has been recovered.

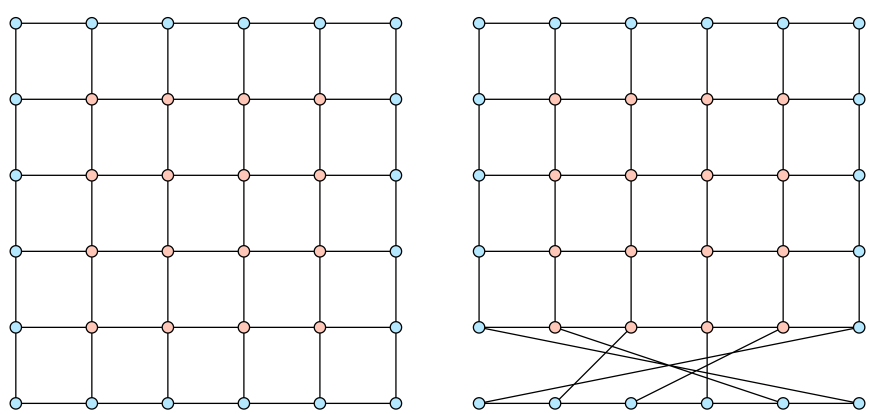


After this, it is possible to repeat the process and bring the vertical direction in until no more interior nodes are left.

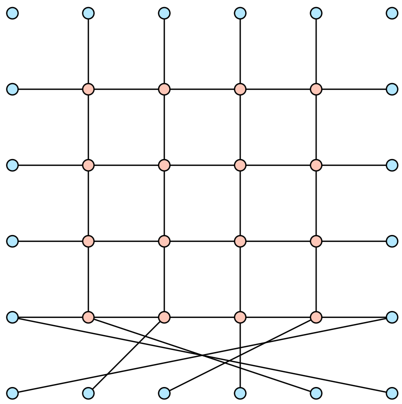
Since both the even and the odd case work, all graphs which have the fringed boundary spike and crosses in over the whole interior work. We can show that this also implies the non fringed case, as the non fringed case is always an open subgraph of the fringed case, and therefore recoverable if the fringed case is recoverable.

2.2 Permutation Graphs

In this section we will demonstrate the recoverability of permutation lattice Graph. A permutation lattice graph is a lattice graph constructed such that each interior vertex is still of degree four, but its edges, which would normally go to its rectangular neighbors, can go to any vertex in the neighboring row or column, but not both. That is, for this method of recovery, in either rows, or columns,



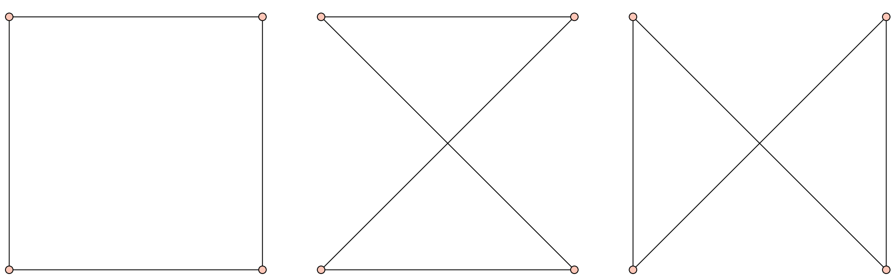
This process can be repeated on each row (rows were chosen in this case). Note that the corners can always be isolated from the interior, and therefore are always recoverable. Without loss of generality, we will assume this direction for the permutations. The major observation is that the North-South connection still exists, and it still saturations the boundary neighbors in a way which allows recovery of the eastern and western boundary boundary edges. The connection still exists because the path which takes one of the southern nodes to the northern nodes is merely permuted (at each level) from the straight rectangular path. Also, it is true that East-West connections still exist and saturate the nodes, so northern and southern boundary boundary edges can be recovered.



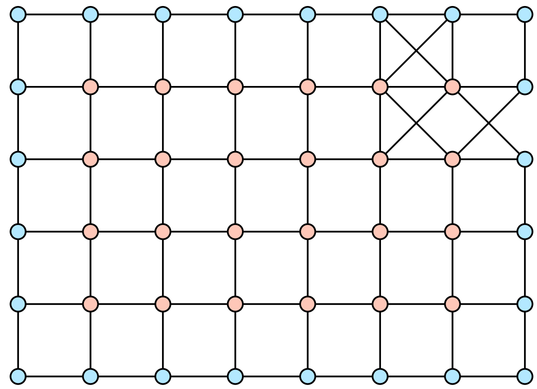
Now all boundary nodes are either disconnected from the interior or members or part of a boundary spike. Using the connections described above, the boundary spikes can be contracted, and the permutation graph can be reduced to a smaller permutation graph. This completes the recovery, as this process can just be repeated until the entire graph is recovered.

2.3 Building Block Network

A building block network is a non fringed lattice graph, which uses 1 or more of the following three "building blocks" arranged in a rectangular formation with the boundary nodes at the standard places for a non fringed lattice graph. When the blocks are put together, any parallel edges will be ignored and made into one edge. The building blocks are pictured below, and will be referred to as the square, horizontal grained cross, and vertical grained cross respectively.



Below is an example such graph using only the square and horizontal grained blocks. Note that it is impossible to have a vertical line in between two crosses using only these two blocks.

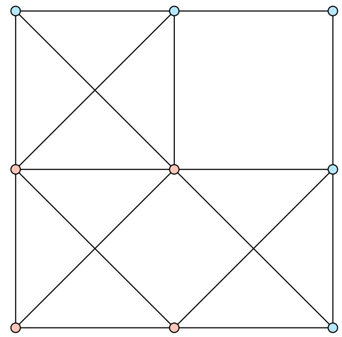


2.3.1 Two Block Cross Graphs

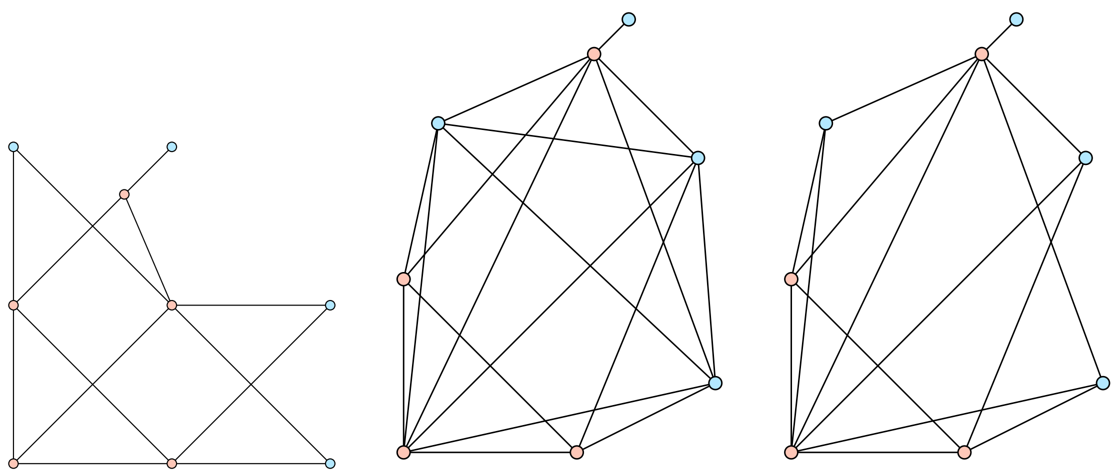
The quickest example is a building block graph built out of the two cross blocks. First we assume that the building block graph has even by even parity. It will be sufficient to show that this is recoverable, as all building block graphs made of these blocks are open subgraphs of an even by even graph. To show that such a graph is recoverable, we need only note that all boundary boundary edges can be recovered by taking paired (and therefore existent) connections across or down, and then recovering the parallel boundary edge. This is guaranteed to produce 4 spikes, one in each corner. From this we have essentially reduced the problem to the case without spikes for the even grained cross graph. The East-West connections are not as obviously existent, but pairing the nodes closest to the middle for connections across or down will recover boundary-boundary edges, and recover more spikes, and so on. This allows induction in on the entire graph. This completes the recovery for all even-even cases, and therefore, using recoverability of open subgraphs, for all building block graphs made from two crosses.

2.3.2 Including the Square Block

When analyzing the recoverability of building block graphs which include the square block, many things happen which are not so easy to analyze. Since most recovery in the past has started at the corners, this seems to be the most logical place to start for building block graphs including the square graph. The corners, however, have a configuration which it is not immediately clear how to recover past and begin the induction. The seemingly worst case for a cross and square graph is in the upper right corner of the example. The relevant subgraph is shown below:



Note that this is not an open subgraph, and that connections exist going both down and across, as this is assumed to be part of an even by even graph, and so pairing guarantees the existence of connections. The first thing to do is to take a Y-delta transform of the upper triangle. Then take a Star-K transformation of the upper right interior node. The Y-delta transformation is a perfectly electrically equivalent transformation, and changing a star into a complete graph leaves a recoverable graph [1, p.16]. The edges of the new complete graph are also guaranteed to satisfy a quadrilateral rule. That is, any two opposite sides of a quadrilateral subgraph of the complete graph multiplied together (or the cross in between) will equal any other two opposite sides multiplied together [2, p.3]. This also keeps the number of independent variables in the graph the same. Though we added 5 edges, we also added 5 relations.



Note that the initial boundary boundary edges were recovered directly, as was the corner boundary node. In the final picture, three boundary boundary edges were recovered by forcing the two boundary vertices not connected by the edge to connect down or across. It is the fact that this recovery removes the corner node from consideration that makes this a considerably harder problem than the two cross building blocks. At this point several recovery techniques were tried, most of which were unsuccessful.

3 Attempted Recovery Techniques

One such recovery technique is to recover one edge in terms of another. If we contract a boundary spike, then we get the new matrix as shown in the first part of this matrix, call it Λ'. It is then possible to recover the newly created boundary-boundary edges (if the proper connections can be found to for them) in terms of the conductivity of the boundary spike. recall from above:

$$\Lambda' = \begin{pmatrix} \lambda_{pp} & a \\ b & C \end{pmatrix}$$

$$\Lambda_{new} = \frac{1}{(\lambda_{pp} - \alpha)} \begin{pmatrix} \alpha\lambda_{pp} & \alpha a \\ \alpha b & (\lambda - \alpha)C - ba \end{pmatrix}$$

If we then proceed to recover k'_{pq} in terms of alpha, where K' is K with the boundary spike associated with boundary node p contracted, we first let

$$\delta = \lambda_{pq} - k_{pp}$$
$$g = (\lambda_{pp}\Lambda(S;q) - \lambda_{pq}\Lambda(S;p) - k_{pp}\Lambda(S;q))C(S;T)^{-1}\Lambda(p;T)$$

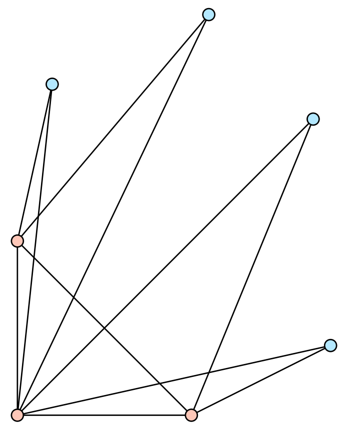
Then we solve for k_{pq} and get:

$$k_{pq} = \frac{k_{pp} * \lambda_{pq}(1 - \frac{1}{\lambda_{pq}} g)}{(\lambda_{pp} - k_{pp})(1 - \frac{1}{\delta} g)}$$

If we have 2 distinct connections with this property (S_1, T_1) and (S_2, T_2), then we also have g_1 and g_2. Setting the recovered values in terms of k_{pp} equal to each other, we discover that $g_1 = g_2$. We can set these two things equal to each other because they are the same edge and must have the same conductivity. This produces some possibly non trivial condition on k_{pp} and its recoverability which seems to be related to the geometry of the graph.

Another technique using this recovery formula applies in cases where you have 2 edges which are recoverable in terms of k_{pp}, and you have some other condition on them such as a ratio resulting from the quadrilateral rule. Setting the values equal to each other multiplied by their known ratio (derived from the other edges in a quadrilateral). This produces a generically non trivial condition and allows recovery of k_{pp}. The condition is generically non trivial because the value of the ratio is essentially independent of the matrix elements currently being used in the recovery of each edge.

Even if one of these two techniques work (as it seems they might), and the corner gets to contract its boundary spike and remove all existing boundary-boundary edges (by sending the two not on either end of the edge to the side or down with the other connections), The corner is still only reduced to the following:



Though 4 of the edges are still related by the quadrilateral rule, this network has no obvious way to start recovery.

4 Conclusions

When attempting to build recoverable non planar graphs, it is useful to build them in such a way that certain connections must exist and use certain nodes. It is sometimes advantageous to look at larger graphs which contain the smaller graphs as open subgraphs, solely for the connection forcing properties of the larger graph. The recovery of building block graphs and, more general types of non planar network (such as conditions for recoverability) is still an open problem. Finally, starting from planar graphs and adding crossings or edges is how many of the graphs in this paper were imagined.

References

[1] Edward B. Curtis, James A. Morrow, *Inverse Problems for Electrical Networks*. World Scientific Publishing Co 2000.

[2] Jeffery Russell, *$\star$ and κ Solve the Inverse Problem*. University of Washington Math REU, 2003. http://www.math.washington.edu/ reu/jim_announcement/current/reu.html, paper 17 as of August 13, 2010.

[3] Jeffery Russell, *Recoverability of Subgraphs*. University of Washington Math REU, 2003. www.math.washington.edu/ reu/papers/2003/russell/subgraph.ps of August 13, 2010.